

# ROOT Q4 2024 Report

*Danilo Piparo (CERN, EP-SFT) for the ROOT Project*

11-12-2024



## ▶ **ROOT 6.34.00 Development Release was released on Nov 29<sup>th</sup>**

- Development release: will be superseded by 6.36, expected in May 2025
- **RNTuple**: the first version of the RNTuple on-disk binary format was finalized ([CHEP talk](#)).
- **Batch generator**: Read the data with RDataFrame, apply transformations if needed and then feed your favourite ML tool ([CHEP 2024](#))
- **LLVM18 and upgraded Python interface** for an even more rewarding experience
- An even **more powerful RooFit**: automatic Differentiation through Clad

## ▶ 2025 proposed Release schedule: **6.36 in May for production**, 6.38 in November 6.38 for development.

## ▶ Metrics as of December 1<sup>st</sup> :

- **915 open issues (1539 in January) – 40% reduction**
- 72 items PoW 61% complete - 2023 55% (55 items), Q1 16%, Q2 28%, , Q3 44%
- **6h on average to provide feedback on posts on the Forum**

*All statistics in this talk  
were gathered on  
December 1st*



- ▶ **ROOT reporting every quarter to stakeholders**, 4th meeting of [a series](#) in 2024
- ▶ Usual questions:
  - Is the current Quarterly Report format useful to you?
  - What can we improve?
  - What should be preserved?
- ▶ Questions, comments, feedback are welcome at the end or during the talk!

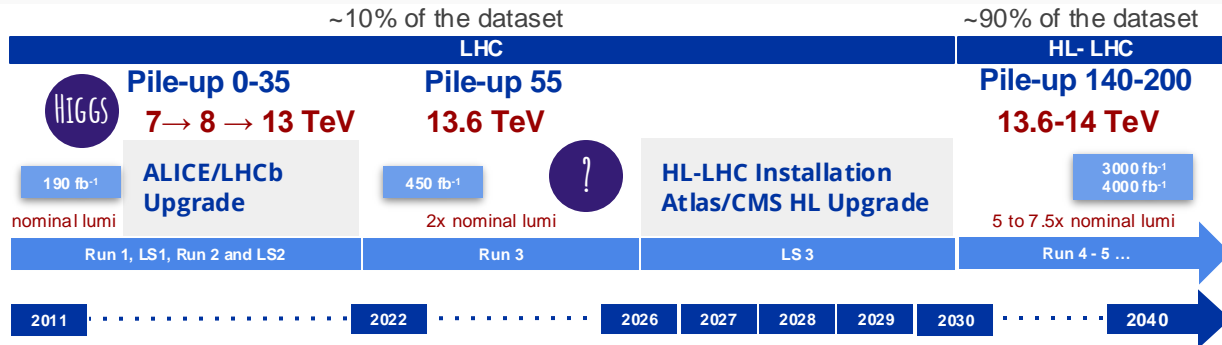


**Provide a unified software package for the storage, processing, visualisation and analysis of scientific data that is reliable, performant, supported and sustainable, that is easy to use and obtain, and that minimises computing resources and scientists' time needed to achieve results.**

**The success of experiments and all ROOT users at large is our priority**



# The Need of Strategic Thinking



HL-LHC timescale: the **EIC** will also start producing data!

QoS	ALICE	ATLAS	CMS	LHCb	Total
Disk [PB]	199	406	304	93	1002
Tape [PB]	283	666	673	250	1875

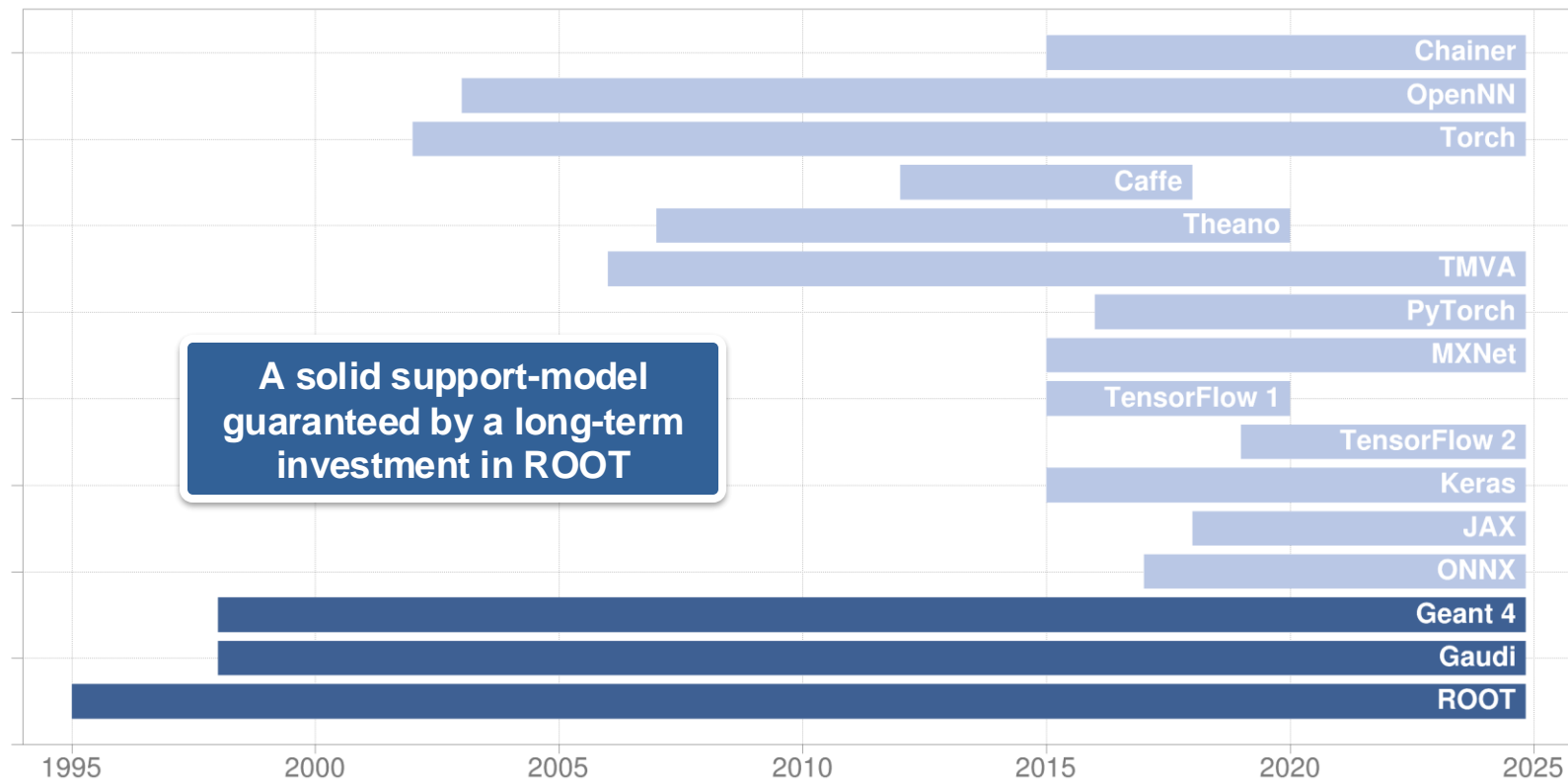
<sup>24</sup> Pledges: source WLCG [CRIC](#)

The full exploitation of the physics potential of present and future accelerators also passes through ROOT

LHC Runs 1,2,3 → Today, >2 EB in ROOT format  
**HL-LHC: ~30 EB in ROOT format?**



# HEP Common Software Support Timeline





# ROOT Core Developers

- ▶ **ROOT is its user community, contributors, and developers**
- ▶ **ROOT is an international collaboration**
  - Institutional responsibilities, but also precious contributions coming from the user community!

Core Devs from <https://root.cern> , PhD students onwards



From left to right, starting from the first row (the affiliation is CERN if not specified):

1. Devajit Valaparambil, Bertrand Bellenot, Danilo Piparo, Florine de Geus, Jakob Blomer
2. Jonas Hanfeld, Jonas Rembser, Marta Czurylo, Olivier Couet, Philippe Canal (**FNAL**),
3. Vasil Vassilev (**Princeton**), Lorenzo Moneta, Monica Dessole, Vincenzo Padulano, Serguei Linev (**GSI**)
4. Giacomo Parolini, Stephan Hageböck, Martin Føll (**Uni Oslo**), Lukas Breitwieser

Plus students, working with us a few months, up to ~1 year  
*Not everybody in this slide is 100% dedicated to one project, but most are.*





# Highlights and Releases





- ▶ Released **on time on November 29<sup>th</sup>**, 6 months after 6.32
- ▶ **>200 items in our trackers solved** for this release
- ▶ **Development release: short term support (until mid ~2025)**
  - Will be superseded by the 6.36 data taking release (May 2025)
  - The first integrated also on GPU-equipped and ARM nodes (besides Linux, Win and macOS – see [CHEP talk](#))
- ▶ **LLVM 18** & upgraded Python interface
- ▶ **ML training directly from ROOT datasets** ([CHEP 2024](#))
  - Read the data with RDataFrame (and therefore leverage the power of ROOT Tree/RNTuple IO), no intermediate conversion
  - Apply transformations if needed
  - Feed your favourite ML tool: PyTorch tensors, TensorFlow Dataset, numpy arrays
- ▶ **RNTuple on-disk format 1.0** ([CHEP 2024](#))

# New ML Training Facility: Batches Generation

```
# Returns two generators that return training
# and validation batches as PyTorch tensors.
gen_train, gen_validation =
ROOT.TMVA.Experimental.CreatePyTorchGenerators(
    rdataframe, batch_size, chunk_size,
    columns=features+labels, target=labels,
    max_vec_sizes=100, validation_split=0.3,
)
# [...] Create PyTorch model
for x_train, y_train in gen_train:
    # Make prediction and calculate loss
    pred = model(x_train)
    loss = loss_fn(pred, y_train)
```

Provide a native data loading abstraction to pipe ROOT data (TTree, RNTuple) into ML training workflows (e.g. PyTorch, TF)

- ▶ **Asynchronous** loading (C++ thread)
- ▶ Supports **scalar** inputs as well as **collections**
- ▶ **Native** ROOT I/O: can read **any** HEP **EDM**, **local** or **remote** files
- ▶ **No** need for **pre-conversion** step to other data formats thus **no duplication**
- ▶ Integrated with **RDataFrame** for batch **preprocessing**

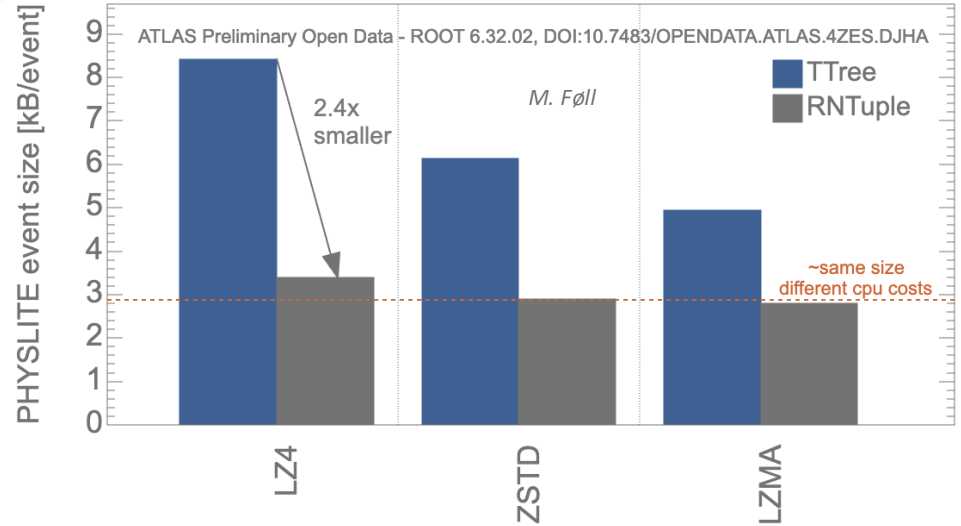
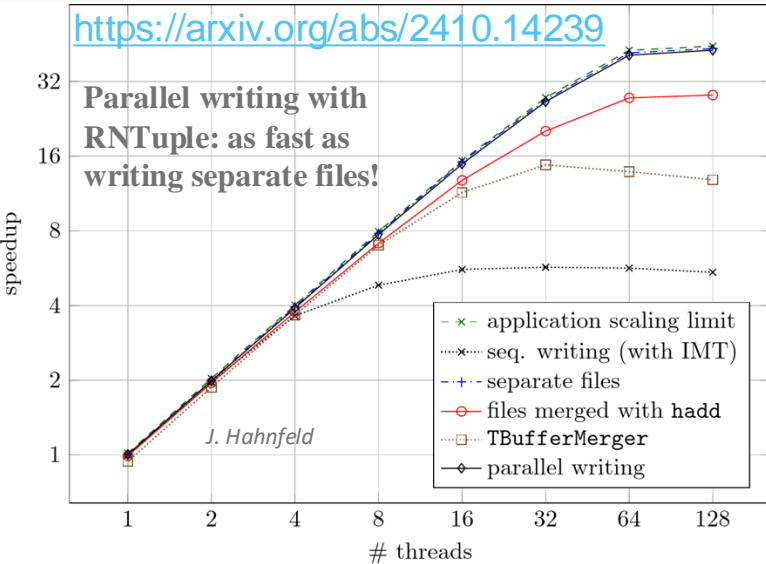




# From CHEP 2024: RNTuple, Some Results

<https://arxiv.org/abs/2410.14239>

**Parallel writing with RNTuple: as fast as writing separate files!**



**Faster, smaller files, more scalable and sustainable**

[Direct I/O for RNTuple Columnar Data](#)

[On-the-fly concatenations and joins with ROOT RNTuple](#)

[RNTuple in the ATLAS production framework Athena](#)

[RNTuple & EOS: Comparative Analysis of Physics Data Formats](#)

[Thread-safe N-tuple Writing in Gaudi with TTree and Migration to RNTuple](#)



[RNTuple implementation in Julia](#)

[ATLAS software tools to handle ROOT RNTuple](#)

[RNTuple and EOS: The Next Generation of Event Data I/O](#)

[RNTuple: A CMS Perspective](#)



## Closing Remarks

- 2024 was a critical year for RNTuple
  - Came from not being able to store any CMS data products except nano to being able to store all of them
  - Many more important challenges solved, in important areas for multiple experiments
  - → Released the first stable on-disk format in time!
- I believe we had many open questions at the end of last year...
  - ... and we have a full plan of work (even beyond 2025) by the end of this year
  - Solved enough problems to have confidence that we can solve the remaining ones, too
- We are entering a new phase in the RNTuple life cycle
  - Transition to production comes with maintenance responsibility: support, training, bug fixes
  - Expect slower pace regarding new functionality
- We would like to stay focused for 2025: release production APIs (ROOT v6.36), schema evolution, *joins*, *attributes (to be properly scoped)*
  - Lower priority: S3, horizontal merge, checkpoint, C API, intra-event links
- Thank you very much for your continued support and collaboration!
  - We'd like to keep a regular channel with I/O experts of the experiments; and the occasional workshop to bring together I/O experts from multiple experiments

J. Blomer

**Thank you very much for coming! We hope it was useful.**



- ▶ **Idea2 at CERN, ~25 participants over 3 days**
  - LHC experiments, CERN but also universities and research centres
- ▶ Three topics/groups:
  - Tutorials modernisation campaign
  - Extension of ROOT's Python interface
  - Python documentation enhancement
- ▶ **~80 PRs submitted and being integrated!**
- ▶ Some of the obtained results:
  - UHI protocol for ROOT histograms
  - A modern version and structure of ROOT's tutorials
  - RNTuple Python interface
  - Better NumPy integration of graphs and histograms
  - Improved documentation across the board

[Check out the blog post on ROOT's website!](#)





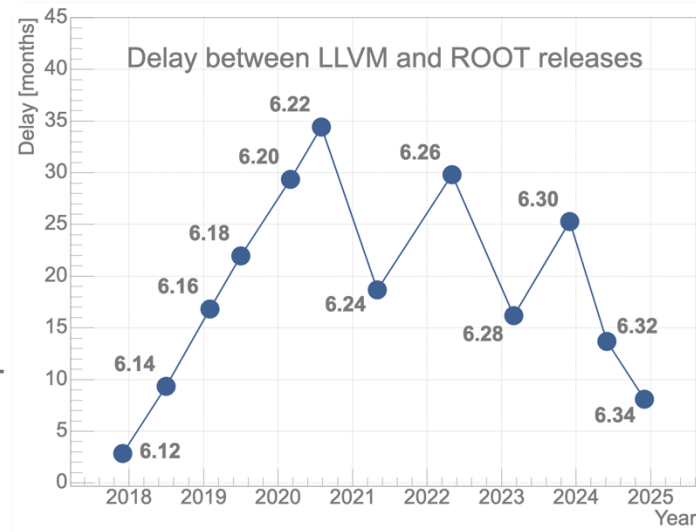
# Proposed Releases in 2024-2025

## Recap of 2024

- ▶ ROOT 6.32.00 Production Release in May 2024
- ▶ ROOT 6.34.00 Development STS Release in November 2024

## Proposal for 2025: timeline analogous to 2024

- ▶ ROOT 6.36.00 Production Release in May 2025
  - Data taking release, long support cycle
- ▶ ROOT 6.38.00 Development Release in November 2025
  - Mainly targeting analysis and exposing new features for all users
  - The following production Release will depend on LS3 plans of experiments (Legacy Processing campaigns, large MC productions, RNTuple commissioning...)





# ROOT Users Workshop: 17-21 Nov 2025

- ▶ **A welcoming, positive and inclusive atmosphere**
- ▶ **An opportunity to shape together the future of ROOT!**
- ▶ A venue for ROOT users, world-class experts of scientific computing and the ROOT core team to exchange ideas and learn from each other
- ▶ A rich program of presentations, tutorials, and most importantly, discussions
- ▶ Strong emphasis on the perspectives of early-career scientists and students

*The registration will be possible via the Indico page of the event, in the next few months*



**In Europe**  
**17-21 November 2025**  
**Save the dates!**



# Q4 Metrics





# ROOT Community and Development

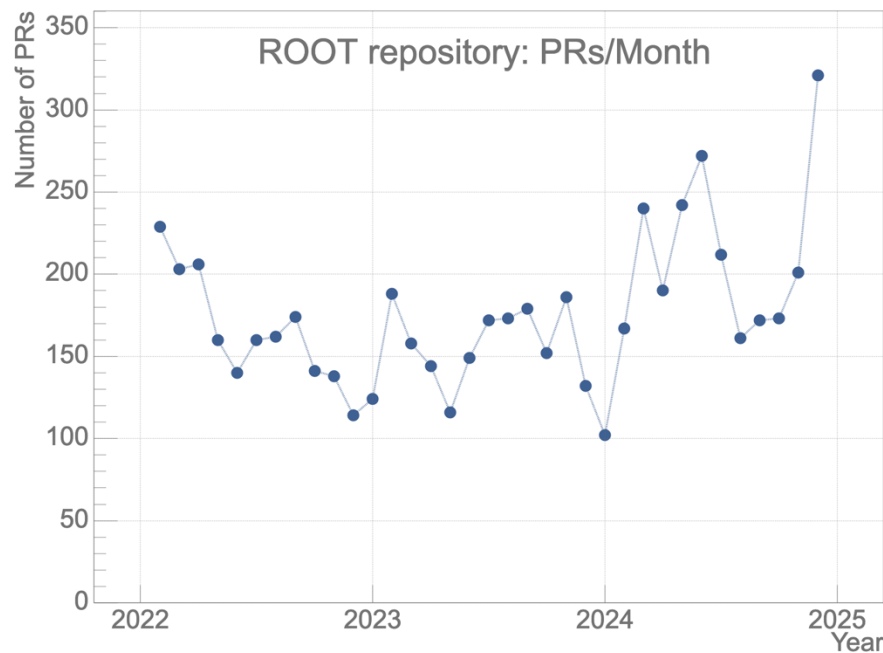
- ▶ ROOT is an **open-source project**, adopting an **open-development** model and **open-planning**
- ▶ **This model pays off: ROOT is supported by a lively community.**

## Contributors/month

<https://www.openhub.net/p/ROOT>



**Many non-core developers contribute to ROOT: your contributions count and are very welcome!**

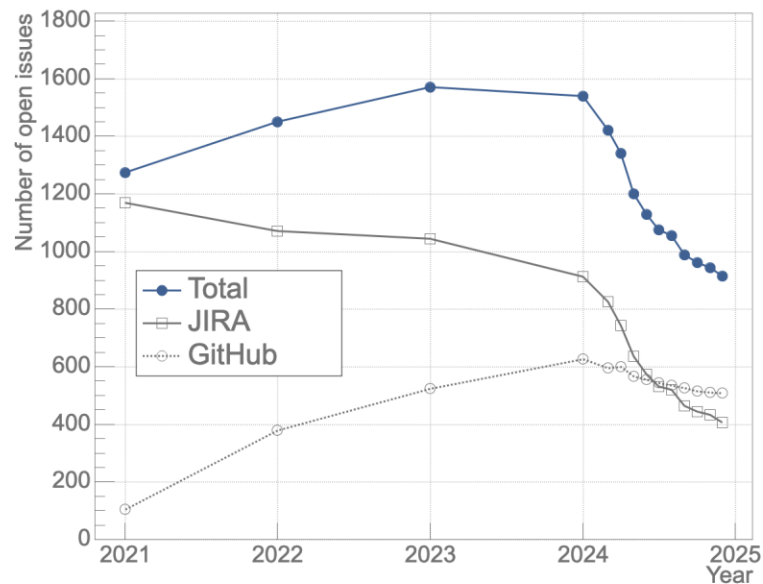




# Number of Open Issues

- ▶ Strong focus in 2024 on reducing number of open items in the tracker
- ▶ Backlog reduction is *implicitly part of the PoW*
- ▶ 2024 so far:
  - 509 new items created, of which 342 solved
  - **1135 solved in total**
  - **40% reduction of # open issues**

ROOT Open Issues, December 2024





# 2024 PoW Completion

	Priority	Completion Status: 0, 5 or 1	
<b>DONE</b>			
<b>PARTIALLY DONE</b>			
<b>NOT DONE</b>			
<b>Builds and Binaries</b>			
1 pip install ROOT for some selected platforms	1	1	
2 Complete transition to GH Actions for builds, adding GPU runners	1	1	
3 Reduce number of services hosted by root.cern with a combination of CERN IT central services	1	0	
4 Win: Replace Debug builds with Release/WithDebinfo in the CI	1	1	
5 Optimise dictionary dependencies to minimise build real time	2	0	
6 Win: Add support for Ninja	2	1	67 %
<b>I/O and TTree</b>			
1 Support std::variant, both in TTree and RNTuple (CMS)	2	0.5	
2 Support writing objects larger than 1GB (TBufferFile > 1 GB, ALICE)	1	0	
3 Complete schema evolution improvements	2	0	
4 Ensure consistency of std::int types across ROOT I/O	2	0	
5 Address residual scaling issues with MT writing	2	0.75	25 %
<b>RNTuple</b>			
1 Complete implementation of merging	1	0.5	
2 Complete implementation of datasets chains	1	1	
3 Limit testing in collaboration with CERN IT	1	1	
4 Follow-up on API review by HEP-CCE	1	1	
5 Implement unsplit ("blobified") encoding	1	1	
6 Support for unaligned friends and joins	1	1	
7 RNTuple: schema evolution	1	0.5	
8 Further develop support for lossy compression with low-precision floats	2	1	
9 Design compression dictionaries and understand implications for the specification	2	1	
10 First implementation of highly-scalable parallel writing	2	1	
11 Organise a Design Workshop to discuss intra-link events, metadata, native SoA layout for events	2	1	91 %
<b>RootFit</b>			
1 Workshop with Experiments: promote features, gather input, speedup integration of RootFit in the existing sw setups	1	1	
2 Numeric integrals in n-dim with CUDA	1	0	
3 Evaluation of custom user functions in CUDA	1	0	
4 Group similar PDFs to speed up evaluation	1	0	
5 Make the new vectorized CPU likelihood evaluation interface the default	1	1	
6 Reduce JITing time for AD in RootFit	1	1	
7 PyROOT: express RooStats configuration with C++-oriented Set* as kwargs	2	1	
8 Integration of Fumili in RootFit	2	0	50 %
<b>RDataFrame</b>			
1 Put existing bulk processing in prod	1	0	
2 DistRDF: reduce memory usage on HTC/Condor Workers	1	1	
3 DistRDF: improve user experience when integrated with notebooks and nb services like SWAN	1	1	
4 Make the TTree → RNTuple transition transparent for analysers	1	1	
5 Further Pythonise the interface	2	0	
6 Deliver varied snapshots	2	0.25	54 %
<b>Math</b>			
1 Python interface: better histos and graph interoperability with NumPy and UHI protocol	1	0	
2 Histos: advance current RHist implementation to one testable by experiments	1	0	
3 Add interface to pass initial error values/cov matrix to Minuit2	1	1	
4 Release a library for Lorentz vector computations on accelerators in SYCL	1	0.5	
5 Deliver plan and prototype of algorithmic improvements when dealing with param constraints in ROOT's minimisers	2	0	
6 PyROOT: Pythonise TF(1,2,3) and numerical algorithms interfaces (e.g. minimisers)	2	0.5	
8 Histograms: Model and prototype of pipelining GPU histogram filling	2	0.5	36 %
<b>ML/AI</b>			
1 Put RBatchGenerator in production	1	0.5	
2 Consolidate RBDT	1	1	
3 Support of integration of SOFIE in experiments Fast Simulation pipelines	1	0	
4 Add support in SOFIE for NVIDIA GPUs in CUDA	1	0	
5 Continue to add support for the ONNX operators requested by experiments	1	0.75	
6 Make HLS4ML interoperable with SOFIE	2	0	
7 Streamline ROOT's inference interface, making it able to use models for Python ML fwks (e.g. Keras/TF) directly	2	0	32 %
<b>Visualisation and UI</b>			
1 Automated placement/tune of plot elements, "Auto Style"	1	0	
2 Add missing features of classic graphics to the web-based one	1	1	
3 Automate web-based graphics test suite	1	1	
4 Add residual missing TEE features to REve, e.g. digit visualisation and text elements overlay	1	1	
5 Visualization of fat ntuples using predefined visual summary data structures	1	0.75	
6 Improve REve window manager and browser, polish render engine	2	0.75	75 %

<b>Interpreters</b>	1 Cling: identify potential Cling codebase reductions through the reuse of parts of clang-repl	1	1
	2 Cling: cppy rebase on top of clang/clang-repl	1	0.25
	3 Migrate PyROOT to the latest Cppyy	1	1
<b>Doc and education</b>	4 Cling: Prototype SYCL support	2	0.5
	1 (Re-)evaluate, update, and improve course material, making it more visible and better organised on the website	1	1
<b>Extra Items</b>	2 (Re-)evaluate luts, eliminating what's outdated, newer features would benefit from a (better) tutorial, improve visibility	1	1
	1 Copyless reading in RNTuple - ALICE	1	1
<b>Substantial items not initially foreseen</b>	2 Physics objects representations out of NanoAOD in RDataFrame - CMS CAT	1	0.5
	3 Bulk Processing + GPU offloading for disRDF - CMS CAT	1	0
	4 Include the open source Tex Gyre Heros clone of Helvetica in root fonts - CMS CAT	2	1
	5 Multithreading-friendly interfaces to the histogram types - CMS CAT/TSG	1	0
	6 A library of matrix operations that can run on GPUs - CMS TSG	1	0
	1 RDataFrame: drastically reduce memory usage to enable very deep computation graphs (O(10K) observables)	1	1
<b>Substantial items not initially foreseen</b>	2 PyROOT: Improve thread-safety of CPython extension (Fixes Ubuntu 24 and distributed RDataFrame)	1	1
	3 Maintenance & update of clangdev-feedstock and root-feedstock (the conda-forge repositories for ROOT packaging)	1	1
	4 RDataFrame: Enable working with missing values (#7713, #9137) (Enabling ATLAS event matching use cases)	1	1
	5 Upgrade to LLVM 18	1	1
6 Add Alma9 ARM runners to CI and address all issues which may result from running on that platform	1	1	100 %

- ▶ A large PoW: 75 items. **61% complete if accounting for extra items**
- ▶ **2023: 55 items (including extra items), 55.4%**
- ▶ 2025 PoW ready, being refined. It will not include ML/AI items, part of the EP-SFT ML4EP Project PoW
- ▶ Will be presented on [January 22](#) and in other slots to allow participation from all timezones. As in 2023, ad-hoc meetings with experiments will be organised when needed.
- ▶ **Open-planning: you can influence the content of the ROOT PoW!**



# Round Table



# CMS Input 1/3: High Priority

- High priority
  - (Core) Schema evolution IORule not working for `std::auto_ptr<T> -> std::unique_ptr<T>` for a split TTree ([cms-sw/cmssw#43923](https://cms-sw/cmssw#43923)). This may be necessary for the planned re-MiniAOD processing of Run 2 UL MC on CMSSW\_15\_0\_X.
  - (CAT) Release of conda packages. It would be desirable if the release of conda packages followed immediately the new releases. E.g. the latest package in conda is 6.32.02, while the latest release is 6.32.04 (in Sep 2024). (<https://codimd.web.cern.ch/s/vdTvYZISf#Release-of-conda-packages>)

Two persons are now working on this issue, and we are making progress:

- A reliable reproducer is under our control
- Hard to estimate an ETA
- **Goal: advance as much as possible before the break**

We will assign to this topic additional effort in 2025, making it part of the PoW. **Contributions from external collaborators, e.g. in CMS, could greatly speedup the process leading to Conda releases.**



# CMS Input: Medium Priority Block

- (Core) More diagnostics from ROOT in “bytecount too large” errors, such as the branch causing the error. These errors have been seen in prompt reconstruction, although the real culprit seems to have been a skim selecting much more events than expected ([cms-sw/cmssw#45089](#), [cms-sw/cmssw#40132](#))
- (Core) Mechanism to read files with `std::auto_ptr<T>` without relying on existence of `std::auto_ptr<T>` ([cms-sw/cmssw#43422](#))
- (Core) RNTuple: Pattern for storing Structure-of-Array data structures (was discussed in the RNTuple workshop)
- (Core) RNTuple: schema evolution support (was discussed in the RNTuple workshop)
- (DQM) Atomic histograms (for Run 4). CMS DQM group is already discussing with the ROOT team, so this item is just to note its priority relative to other items.
- Following three have workarounds
  - (Core) Schema evolution problem with a recursive class when reading a split TTree ([cms-sw/cmssw#43768](#))
  - (Core) Schema evolution problem when removing an intermediate base class ([cms-sw/cmssw#43516](#))
  - (CAT) Towards a transient-persistent layer to read NanoAOD in RDataFrame ([cms-sw/cmssw#45972](#))

We could not deliver as much as we wanted in the medium priority level because of the load generated in other areas.

We believe this is part of the "auto\_ptr item" in the high priority block

The api of RNTuple will likely avoid sophisticated read rules. For writing, we have some ideas and took this as a point to work on. See this talk at the [workshop](#)

A strategy has been identified, reducing the scope of automatic evolution in favour of explicit rules. 6.36 will have an automatic schema evolution, which will do less than TTree's.

This is correct. We are also making progress on a mechanism for TS filling of current histograms.

RNTuple is part of the solution. In the end, we **might** think to a solution that only works with RNTuple.



# CMS Input 3/3: Low Priority

- Low priority
  - (Core) Fix GetMethodWithPrototype returning incorrect function ([root-project/root#7955](#), [root-project/root#16232](#))
  - (Core) Fix for string cut parser sometimes not seeing all methods of an object when using ROOT reflection. This error mode has been worked around such that CMS has not seen it in a long time, but presumably the race condition itself is still there ([cms-sw/cmssw#33084](#))
  - (Core) Infrequent crashes in Cling on x86 ([cms-sw/cmssw#44438](#))
  - (Core) Crashes in Cling on ARM ([cms-sw/cmssw#43802](#), [cms-sw/cmssw#41961](#))
    - CMS has not seen these crashes (at least for a month) in the CMSSW build following ROOT master, whereas in the CMSSW main build (with ROOT 6.30) we see few crashes on every IB test
  - (Core) Rare (non-Cling) crashes on ARM ([cms-sw/cmssw#42070](#), [cms-sw/cmssw#42527](#), [cms-sw/cmssw#42564](#))
  - (Core) A case where rootcling crashes when generating dictionaries for type aliases ([cms-sw/cmssw#44386](#))

Master and 6.34 are indeed the first branches where ARM is treated as a 1<sup>st</sup> class platform.

**Thanks for coordinating the collection of this input. For us, this is very useful. Would other experiments be interested in providing this kind of structured feedback?**