Eur. Phys. J. C (2025) 85:1142
https://doi.org/10.1140/epjc/s10052-025-14571-6

THE EUROPEAN PHYSICAL JOURNAL C

Editorial

The critical importance of software for HEP

Christina Agapopoulou[1], Claire Antel[2], Saptaparna Bhattacharya[3], Steven Gardiner[4], Krzysztof L. Genser[4], James Andrew Gooding[5], Alexander Held[6], Michel Hernandez Villanueva[7], Michel Jouvin[1,a], Tommaso Lari[8], Valeriia Lukashenko[9], Sudhir Malik[10], Alexander Moreno Briceño[11], Stephen Mrenna[4], Inês Ochoa[12], Joseph D. Osborn[7], Jim Pivarski[13], Alan Price[14], Eduardo Rodrigues[15], Richa Sharma[10], Nicholas Smith[4], Graeme Andrew Stewart[2], Anna Zaborowska[2], Dirk Zerwas[16], Maarten van Veghel[17]

[1] Université Paris-Saclay, CNRS/IN2P3, IJCLab, Orsay, France
[2] CERN, Geneva, Switzerland
[3] Southern Methodist University, Dallas, TX 75205, USA
[4] Fermi National Accelerator Laboratory, Batavia, IL 60510, USA
[5] Technische Universität Dortmund, Dortmund, Germany
[6] University of Wisconsin-Madison, Madison, WI, USA
[7] Brookhaven National Laboratory, Upton, NY 11973, USA
[8] INFN Sezione di Milano, Milan, Italy
[9] Physik Institute, Universität Zürich, Winterthurerstrasse 190/Building 36, 8057 Zurich, Switzerland
[10] University of Puerto Rico, Mayagüez, PR, USA
[11] Universidad Antonio Nariño, Ibagué, Colombia
[12] Laboratório de Instrumentação e Física Experimental de Partículas, Lisbon, Portugal
[13] Princeton University, Princeton, NJ, USA
[14] Jagiellonian University, ul. prof. Stanisława Łojasiewicza 11, 30-348 Kraków, Poland
[15] University of Liverpool, Liverpool, UK
[16] DMLab, Deutsches Elektronen-Synchrotron DESY, CNRS/IN2P3, Hamburg, Germany
[17] Nikhef, National Institute for Subatomic Physics, Amsterdam, The Netherlands

Received: 14 June 2025 / Accepted: 14 July 2025

1 Preamble

Particle physics has an ambitious and broad global experimental programme for the coming decades. Large investments in building new facilities are already underway or under consideration. Scaling the present processing power and data storage needs by the foreseen increase in data rates in the next decade for HL-LHC is not sustainable within the current budgets [1,2]. As a result, a more efficient usage of computing resources is required in order to realise the physics potential of future experiments. Software and computing are an integral part of experimental design, trigger and data acquisition, simulation, reconstruction, and analysis, as well as related theoretical predictions. A significant investment in computing and software is therefore critical.

Advances in software and computing, including artificial intelligence (AI) and machine learning (ML), will be key for solving these challenges. Making better use of new processing hardware such as graphical processing units (GPUs) or ARM chips is a growing trend. This forms part of a computing solution that makes efficient use of facilities and contributes to the reduction of the environmental footprint of HEP computing [3]. The HEP community already provided a roadmap for software and computing [4] for the last EPPSU, and this paper updates that, with a focus on the most resource critical parts of our data processing chain.

2 Physics event generators

Monte Carlo Event Generators (MCEGs) that simulate particle collisions, or 'events', to a level that allows direct comparison with experimental data, are indispensable for the planning and analysis of all particle physics experiments.

Prepared by the HEP Software Foundation, with inputs from the HEP community.

This document has been endorsed by the following experiments and communities: ALICE, ATLAS, Belle II, CMS, DUNE, ePIC, LHCb, MCnet, WLCG.

[a] e-mail: michel.jouvin@ijclab.in2p3.fr (corresponding author)

MCEGs are vital to connect theoretical ideas and calculations to experimental results. They provide inputs to detailed software models of detectors, are key components in the evaluation of systematic uncertainties, and frequently play an essential role in the interpretation of those measurements.

As such, the MCEGs buttress the continued success of the experimental programme through sustained improvements responding to increasing demands on their precision, reach, flexibility, and usability. The development, maintenance, validation and tuning of these central assets is driven by a relatively small, but critically important, ecosystem of researchers. The ongoing physics exploration at the (HL-)LHC and other current experiments, and the preparation for future facilities, requires the continuation of this strategic and vibrant research programme. The intersectionality of MCEG developers poses serious challenges in terms of career paths and funding opportunities, and it is imperative that the particle physics community addresses such issues to ensure a sustainable development of MCEGs for existing and future facilities.

As well as the MCEGs themselves, there is a critical ecosystem of tools to exchange data with other software, HepMC3 [5] and LHE [6] in particular; as well as data access via, e.g., LHAPDF [7] and HEPData [8]. Long term, adequate support for these infrastructure tools is vital.

2.1 Large hadron collider

The unprecedented precision in measuring the mass of the W boson [9] and related SM parameters at the LHC illustrates the need for exceptional precision. With HL-LHC datasets, theoretical precision needs to be of $\mathcal{O}(1\%)$ (or even 0.1% for the top-quark mass). This high precision event generation comes at the cost of computing resources, and can reach 20% of the total CPU budget, so considerable R&D is needed to ameliorate this. Significant challenges are outlined below and, although progress has been made, there is still work to be done [10–12].

2.1.1 Reduction of negative weights in event generation

Events with negative weights can occur in MC schemes that include higher order (next-to-leading order, NLO; next-to-next-to-leading order, NNLO; and beyond) corrections to cross sections and can significantly dilute the statistical power of a sample. This statistical loss is most significant for events that have to be propagated through the downstream steps, which are computationally expensive. Positive resampling [13] and cell resampling methods [14] have been developed by the generator theory community – these methods redistribute weights, without introducing bias, thereby significantly reducing the proportion of negative weights. Other methods include the MC@NLO-Δ-scheme [15], an NLO-accurate matching prescription that reduces the number of negatively weighted events. All of these methods are currently being tested by ATLAS and CMS.

2.1.2 Heterogeneous computing

Matrix element calculations, which are usually the most CPU intensive part of higher order MCEG computations, can be offloaded to GPUs. The first benchmarking of the gains from using the GPU version of Madgraph [16–19] for leading-order calculations was performed by the CMS Collaboration [20] and showed improvement up to $\times 7$ in event generation time. The PEPPER (Portable Engine for the Production of Parton-level Event Records) [21] event generator, developed by the Sherpa generator team [22] parallelises the entire parton level event generation and has been tested across a wide range of GPU architectures. The benchmarking exercises performed by experiments are still in nascent stages, with the eventual goal that event generators will be run on GPUs during huge MC production campaigns.

2.1.3 Use of ML in event generation

Neural networks can be used to approximate matrix elements and have been studied in the context of loop-induced diphoton-plus-jets production through gluon fusion [23]. Such approaches to event generation can reduce the computation time by up to $\times 10$. ML-based hadronization models that can replicate the performance of the Pythia 8 or Herwig generators for certain kinematic distributions have been developed [24–27]. These ML models offer the possibility of being directly built from data, so can provide insights into phenomenological models currently in use. Employing ML-based reweighting techniques can alleviate the problem of generating MC samples for several systematic variations – instead of generating multiple samples, only one sample with weights representative of each systematic variation can be sufficient [28]. ML techniques are also being explored for phase-space sampling and unweighting [29].

2.2 Higgs/top/electroweak factories aka Higgs factories

In addition to general-purpose generators, LEP-era generators, which are more specialised to certain processes, have been updated [30–33] and several have moved from Fortran to C++ [34,35]. Modern generators, developed for Higgs factories [36,37] or for LHC physics [22,38], due to their process independent algorithms, have the capability to provide event generation not only for the Standard Model, but also for Beyond the Standard Model (BSM). This feature is already exploited by Belle II in the generation of samples for Dark Matter searches. Specialised tools are necessary, e.g., for providing luminosity spectra [39] based on Guinea Pig [40]. The interface and validation of such tools, as well as adapting the general-purpose Monte Carlo programs to the precision of LEP era MCEGs, is crucial for the physics programme of future Higgs factories.

The high-precision measurements at a Higgs factory poses tremendous challenges. All processes have to be known to at least NNLO electroweak (EW) and some, such as Bhabha scattering, even N^3LO EW [41], which will require the development of new efficient algorithms to automate the calculation of these radiative corrections. These fixed-order corrections must also be supplemented with all-order resummation methods. Currently, the electron parton distribution function (PDF) at leading logarithmic (LL) and next-to-leading logarithmic (NLL) order is known, which must be matched to an exclusive description of the photon phase space, such as a parton shower, to at least NNLL [41–43]. Such universally applicable matching formalism between fixed-order and resummed calculations with exclusive radiation simulations and QED showers is needed to achieve per-mil precision for inclusive and exclusive predictions [44]. A complementary approach is the Yennie–Frautschi–Suura (YFS) resummation, which was crucial to the LEP physics programme as it was the main approach in the KK generator [30], in which both the order-by-order perturbative improvements and resummation of soft logarithms is achieved in one method. This method has been implemented in process independent fashion in Sherpa [45], as an example of how state-of-the-art predictions from the LEP era can benefit from technical advances in LHC MC tools.

Specialised tools are necessary for specific processes, e.g., luminosity determination (BHLumi [32], Babayaga [31]), top threshold. Further developments/calculations are needed to reach the required accuracy. Phase-space sampling is being improved with adaptive multi-channel versions of VEGAS or machine-learning methods, and can utilise GPUs.

Since the last EPPSU, the Key4hep [46,47] software ecosystem has emerged as a common framework supported by and supporting all Higgs factory proposals (FCC, CLIC, ILC, CEPC), and also the Muon collider and Electron-Ion-Collider. It provides coherent interfaces to many generators enabling meaningful systematic comparisons of cross sections, differential distributions and more. Furthermore, scripts to generate events in the Key4hep ecosystem are provided that are of huge utility to the community. As well as continued support for Key4hep, an important goal is to develop automated generator-to-generator comparisons for Higgs factories, i.e., complementing the tests performed by each generator individually.

2.3 Hyper-K and DUNE

In the neutrino community, high-quality event generators are a critical need for next-generation analyses, particularly those pursued by the accelerator neutrino oscillation experiments Hyper-Kamiokande (Hyper-K) and the Deep Underground Neutrino Experiment (DUNE). To make definitive measurements of neutrino and SM properties requires unprecedented percent-level control of systematic uncertainties, including those related to simulations of neutrino-nucleus interactions. These interaction uncertainties, which arise to mitigate event generator mismodeling, are particularly difficult to reduce to the required level because of stringent theoretical demands: full final states must be predicted over orders of magnitude in neutrino energy for multiple targets, and the various interaction modes that contribute are each subject to complicated nuclear effects. Delivering the simulation capability to allow Hyper-K and DUNE [48] to be successful will thus require advances in nuclear theory, as well as corresponding investment in development and maintenance of the neutrino event generators.

Although neutrino event generators are at an earlier stage of development and supported by a smaller workforce, there are many challenges shared with simulation efforts in collider experiments and other branches of HEP [49]. Specific obstacles to the neutrino community must also be overcome. The field is heavily reliant upon nuclear research, which is often funded for other applications. Technical barriers already solved for the LHC still remain an issue for the neutrino community due to a lack of dedicated effort, e.g., there is no common event format (although one has been proposed [50]) and no standardised interfaces for related software. The implementation of new models also typically remains very labour intensive, with adaptation of LHC techniques for streamlining the work only just beginning [51]. Supporting the wide variety of new physics searches envisioned for future experiments, will, in particular, require novel approaches.

2.4 Recommendations

In view of the current state and importance of MC generators (and their growing projected importance for the HL-LHC) and the lack of resources routinely plaguing the generator theory community, the following recommendations are made:

1. Generator theory community funding, including fellowships for students, and ensuring expert continuity by rewarding work on computational aspects of generators with secure positions.
2. Supporting workshops on specific aspects of MC production that help bridge the gap between theory and experiment and provide travel awards for early career scientists.
3. Support for developments of Higgs factory event generators and the Key4hep ecosystem to bring together Monte Carlo generators and experiments into a coherent framework.

4. Ensure support for the neutrino event generators, with their specific needs, and encourage as much sharing as possible with the other communities.
5. Provide sufficient long term support for MCEG infrastructure codes.

3 Detector simulation

As the volume of data collected by high energy experiments increases, so does the need for large, accurate Monte Carlo (MC) datasets. The simulation of the particle propagation in the detectors, and the response of readout channels, is a major component of experiment computing budgets.

3.1 Baseline simulation code needs

Almost every HEP experiment uses the Geant4 [52–54] simulation toolkit. Its development and maintenance over the long timescales, driven by the needs of experiments using the toolkit, are of critical importance and need to be supported. Higher intensity and luminosity require more precision, as does the use of higher granularity calorimeters. The main ongoing developments [55] are the implementation of more detailed EM models and rarer processes, the inclusion of low energy nuclear physics effects, and the continued tuning of string models. The FLUKA.CERN [56,57] hadronic physics code have been made available as an option in Geant4, and use of MC generator code for hadron interaction and decay processes is being explored. The validation of Geant4 physics against thin and thick target experimental data for each release is essential. FLUKA.CERN itself, which is being modernised to C++, remains critical for radiation transport and shielding calculations.

The computing load from electronics simulation and pileup overlay in the digitisation step is heavily dependent on detector design and consumes significant computing for all LHC experiments. Re-use of background induced digitised output has been deployed by all of them.

Since the last EPPSU [58], most LHC experiments have transitioned their simulation applications to multithreaded execution, supported by Geant4, and modernised their code. ALICE uses sub-event parallelism, which is now being implemented in Geant4. ATLAS and CMS have reported [59, 60] throughput increases of a factor 1.5 to 2 at negligible cost in physics accuracy since the beginning of Run 2, from improvements in their own code and in Geant4. This is a process that can, and must, continue.

3.2 Novel computing architectures

Market trends favour computing using GPUs, which can be more energy efficient if used intensively, in addition to porting to more power efficient CPU architectures, such as ARM. There is thus considerable interest in running the compute intensive simulation on GPU-CPU hybrid architectures. Efforts are focusing on the calculation of EM processes on GPUs, which can be up to two thirds of the run time in HEP collider detector simulations. The AdePT [61,62] and Celeritas [63] projects, working with the experiments, have demonstrated the possibility to perform the EM part of simulations in simplified setups on GPUs faster than on CPUs [64,65], while simulating hadronic physics and processing experiment-specific energy deposits on CPUs. Bottlenecks, particularly in the geometry, have been identified that are being addressed by a new surface geometry approach. Further work is needed to completely integrate GPU-based simulation modules into experiment software frameworks, but the goal of using GPUs for MC datasets for the major LHC experiments by the start of Run 4 appears within reach. The other leading candidate for GPU simulation is the simulation of optical photons, which is a computing bottleneck for LHCb and several Intensity Frontier and Dark Matter experiments [66].

3.3 Fast simulation

Since the last Strategy Update [58], fast simulation techniques have been developed and used in the experiments' MC production. Parametrised and generative machine learning (ML) approaches for calorimeter shower simulations are used (e.g., [67]), or planned to be used, in a substantial fraction of MC production by ATLAS, CMS, and LHCb. Generative adversarial networks for background are already developed and planned to be used in Belle II. The use of fast simulation is expected to increase in the next few years, with more applications transitioning from R&D to production. More sophisticated ML methods, e.g., INN [68], CFM [69] are being explored. Fast simulation R&D includes the generation of low-level detector output, tracking with simplified geometry, fast inner detector simulation based on parametrised models (e.g., FATRAS used by ATLAS [70]) or using ML-methods, and the correction of fast simulation output to match detailed simulation training samples. ATLAS's FastChain tool combines fast tracking and fast calorimeter simulation in one workflow. CMS, LHCb, and ALICE are investigating the more radical approach of generating high-level outputs for physics analysis based on the event generators' output, saving greatly on storage. Since fast simulation techniques still need large, accurate simulation samples for training, they do not remove the need for detailed particle tracking simulations, but they can reduce the total simulation time and the size of the simulated samples.

Large experiments have dedicated fast simulation development teams, an approach that can obtain the best performance for a specific application, but smaller experiments cannot afford this level of specialisation. Public datasets for the training of fast simulation models have been provided by the experiments, and an experiment-independent framework to develop ML-based fast calorimeter simulation models has been developed in Geant4. The field is in rapid development and the balance between experiment-specific code and the common code still needs to be found, but it is clear that the development of common frameworks and tools needs to be supported to make the best use of a limited pool of developers and avoid unnecessary duplication of efforts.

Fast simulation and ML-based approaches to mitigate the resource usage of the current pileup mixing approaches are being considered by ATLAS and CMS.

3.4 Future experiments

The simulation requirements for the design of future detectors are very different from those of the running experiments. While large samples are not usually required, it is important to be able to simulate many different design variations, so flexibility and ease of use are essential. Studies for detectors at proposed future colliders are using the common Key4hep [46] framework. Gaseous detector R&D relies heavily on Garfield++ [71] for detailed modelling of signal formation. Long term support of these tools is critical.

The FCC-ee experiments should collect datasets similar or larger in size than that of HL-LHC experiments [72]. They will perform precision measurements in a cleaner environment than hadron colliders, so simulation needs to be at least as fast, but also more accurate than for the HL-LHC. The Muon Collider must handle a large background from beam muon decays, which requires beam background simulation to be overlaid with simulated collision events. FCC-hh experiments will need the extension of physics models used in hadronic interaction simulation to higher energy, as well as overlay of up to 1000 pileup events per signal event and thus huge event, and sample, sizes.

3.5 Recommendations

1. Improve the level of support for critical core simulation software code in Geant4, improving the fidelity and speed of physics models, and ensuring a generational transition before current experts retire as well.
2. Provide suitable long-term funding to maintain new developments which come to fruition.
3. Invest in R&D lines that can explore simulation on accelerated hardware, and move into production if successful.
4. Invest in fast simulation support, particularly using ML, where this can be adapted to and support different experiments and detector studies, as well as benefit current experiments through exploring novel approaches.

4 Reconstruction and software triggers

LHCb [73] and CMS [74] are now successfully operating GPU farms for their Run 3 software trigger systems and ALICE [75] uses such a farm for synchronous reconstruction, online calibration of detectors and compression of the real data. This effort has produced impressive, sustainable production software providing speed-ups, portability and vendor independence [76–78]. The upgrade has enabled ALICE and LHCb to fully adopt real-time analysis strategies, discarding raw data in favour of highly compressed or reconstructed data. This reduces storage needs and meets baseline physics goals, demonstrating that online calibration and detector alignment in quasi real-time is possible. ATLAS [79] and CMS [80] have increased their bandwidth allocation for real-time analysis triggers in Run 3 by factors of 2 to 3, allowing them to collect up to an order of magnitude more physics events beyond baseline for low mass exotic particle resonance searches.

The need for faster reconstruction and data processing algorithms is being widely addressed by ML and AI algorithms. AI tools are already used extensively in HEP offline software and are increasingly found in online software as well as hardware DAQ systems. For example, ATLAS uses, and continuously improves, transformer neural network based heavy flavour jet and tau identification in its high-level trigger (HLT) [81] in Run 3. CMS deployed a variational autoencoder based on an anomaly detection demonstrator in their Level-1 trigger in 2023 [82], making use of HLS4ML [83], a HEP community-driven software package for high-level synthesis of ML algorithms for FPGAs. In Run 3, LHCb has deployed monotonic Lipschitz networks at the HLT stage [84] to select heavy-flavour-quark decays in an efficient and robust way, as well as adding neural network models to their HLT to improve track purity [85].

Many experiments have recognised the compute intensive need for tracking capabilities in the trigger for improved event selection. Since Run 3, LHCb reconstructs tracks on GPUs at the LHC bunch crossing rate, identifying quality particle candidates for offline analysis for every event [86]. This avoided hardware-based first level trigger selections impacting its physics programme, such as measurements for CP violations in heavy flavoured hadron decays. Also from Run 3, CMS runs track reconstruction on GPUs based on the pixel detector data and will make a big advance at the HL-LHC by running tracking for particle flow reconstruction in a hardware-based first level trigger. The next important advancement will be 4D reconstruction: track reconstruction with the use of hit time information. This will be applicable at the HL-LHC and to future hadron, electron-positron, and deep inelastic scattering collider facilities for background mitigation. The tracking R&D area has fostered the A Common Tracking Software (ACTS) project [87], an excellent example of a common software toolkit used by several experiments to tackle tracking challenges as a community. It is already proving a useful platform for common tools, such as traccc [88,89], a demonstrator project for GPU-based tracking.

However, the rapidly evolving software landscape in HEP is highlighting a struggle to keep physicists sufficiently trained to contribute to reconstruction and trigger software in a useful way (e.g., C++ expertise is increasingly hard to find). There is a growing need for a dedicated group of software and physics professionals working together. Proper software engineering leads to immense benefits and is a foundation for software sustainability, as well as efficient and reliable, hence more environmentally sustainable, computing.

The community has identified several key themes in reconstruction and software triggers that need to be intensively explored in the next decade to achieve our physics goals. Not all areas are relevant for every experiment, but combined efforts in the proposed fields will broadly benefit future HEP programmes.

4.1 Enhanced heterogeneous computing software

Utilising heterogeneous architectures, such as GPUs and FPGAs, appears inevitable in several different areas, yet one of the main challenges is anticipating the technological landscape in 5–10 years. While GPUs are one of the most promising accelerators, alternative accelerator and processor technologies, such as RISC-V, IPUs and AI-processors, are evolving rapidly and need to be closely monitored. The community should aim to improve the flexibility of its software in order to explore and adapt to a variety of architectures, without being tied to a specific hardware vendor.

4.2 Real-time analysis

Beginning with Run 3 ALICE has been operating triggerless continuous readout, and future online data acquisition software in HEP and NP trends towards operating triggerless or streaming readouts, to avoid online event selections that would limit primary physics goals. As rates increase, these techniques move closer to the detector, hosted on less flexible systems, which requires very careful system design, as well as operational models and validation. Selective persistency models will likely need to be further exploited to reduce storage pressure, along with tools for real-time detector calibration.

4.3 Integration and support of AI/ML algorithms

Interest in AI/ML solutions is rapidly increasing as they lend themselves well to low latency approaches for real-time analysis and fast data processing. Reaching the latency and throughput requirements at the lowest levels requires specialised expertise, posing a significant challenge and open source collaboration projects should be pursued further. For example, the HLS4ML project [90,91] has provided an easier, faster means of developing ML algorithms for FPGAs, together with the capability to generate bitwise correct C++ representations for simulation, thus reducing the gap in firmware expertise.

4.4 Exploitation of precise timing in the reconstruction sequences

Many HEP experiments are planning, or incorporating, precise timing ($\mathcal{O}$(ps)) detectors, in order to maintain performance in ultra-high density environments (e.g., at Belle-II and HL-LHC). For the HL-LHC and future colliders, reconstruction techniques that use hit level timing (4D reconstruction) can much more efficiently distinguish signal from background. Adapting reconstruction algorithms to keep pace with evolving sensor and detector R&D is a significant challenge and requires community effort.

4.5 Enhanced software maintenance and quality assurance (QA)

Maintaining validated online and offline reconstruction software frameworks is a significant challenge, especially with the move towards heterogeneous systems. Dedicated funding is essential to achieve this. Proper software engineering and common tools for continuous integration and automation of quality assurance tasks will ultimately conserve resources, as will identifying key software developments that are widely applicable. Common software can be more robustly tested and validated, with the benefits shared by many. This will foster a sustainable software environment and at the same time increase computing efficiency to improve physics reach.

4.6 Recommendations

1. Support R&D in heterogeneous architectures along with the associated skill development.
2. Invest in further development of real-time and streaming readout systems with emphasis on reliability, to address growing storage pressures in future ambitious HEP programmes.
3. Support ML/AI developments, in particular platforms for sharing ML/AI methods and tools specific to HEP applications and ML/AI implementations on novel devices.
4. Invest in new algorithm developments using evolving precision timing detector technology to significantly advance particle reconstruction in dense environments.
5. Support common software solutions that encourage collaboration between experiments.

6. Invest in sustainable software development by promoting software maintenance, validation, quality assurance, dedicated software groups and developer training.

5 Data analysis

The data analysis software ecosystem, that is, the common frameworks, toolkits and packages typically employed in HEP data analysis, has generally migrated from pure C++ to a C++/Python hybrid over the last 10 years. The increased prevalence of Python has been driven by usability benefits and changes in language preferences of physicists, with community initiatives such as Scikit-HEP [92] promoting and consolidating Pythonic HEP analysis tools and the "Python in HEP" Developer's Workshop (PyHEP.dev) [93] bringing together experts to form new collaborations. Typically, earlier stages of data analysis use C++ and later stages use Python as a "glue language" to steer frameworks such as ROOT. Further emphasis on interoperability and sustainability have encouraged many tools to implement/enhance their Python compatibility, e.g., through C++-Python wrappers, such as ROOT's Python Interface. To overcome performance limitations, many Python packages have adopted an *array programming* interface, where large arrays of data are passed into a compiled layer for performance, focusing the Python portion of codebases to the task of dispatch and metadata manipulation. A recent addition to the community is a small but growing group of analysts employing Julia. Whilst not expected to become the mainstream in the immediate future, the significant performance benefits against Python and usability benefits against C++ make Julia an appealing language.

In parallel, data analysis frameworks are improving integration with computing infrastructure. The ROOT framework has developed the RDataFrame [94] interface, with improved multithreading and interfaces to Spark [95] and Dask [96], two popular distributed computing technologies. In the Scikit-HEP ecosystem, awkward-array [97] and Coffea [98] have closely integrated Dask to achieve scalability.

Given the wide range of tools available to HEP analysts, interoperability across the software ecosystem is necessary, with the need to build well-specified and interoperable data structures for all stages of analysis. The ROOT RNTuple data serialisation interface, replacing the TTree interface, has a clear specification as well as significant enhancements in support for multithreading and asynchronous I/O [99]. The HEP Statistics Serialisation Standard (HS3) aims to provide a common, framework agnostic serialised description of the components involved in fitting statistical models (models, datasets, likelihoods, etc.) [100].

5.1 Machine learning and automatic differentiation

ML has had a transformative impact on HEP and become a core part of HEP data analysis. Traditionally, ML algorithms such as decision trees and neural networks (NNs), are used to efficiently classify events. More recently, the use of ML in analysis has extended to statistical inference, with results from ATLAS [101,102] using simulation-based inference, wherein ML classifiers are trained on simulation and used to construct statistics, such as likelihood ratios, which can be evaluated on data. We expect ML to keep playing a significant role in the future.

Automatic differentiation (AD), techniques for the fast and precise computation of derivatives of functions, has potential to provide significant performance improvements across HEP analysis. In statistical inference and fitting, where derivatives of functions are evaluated many times, AD has been shown to be significantly faster than existing methods, e.g., in the codegen backend in RooFit [103]. AD also enables applications such as systematics-aware training of NNs.

5.2 Sustainable software development and analysis

Sustainability has become a major focus of data analysis software and is typically discussed in the context of environmental and economic impact, software lifecycles and analysis reproducibility.

Many important software packages in HEP analysis are developed and maintained by early career researchers, who are typically employed on fixed-term contracts of no more than a few years and for whom the time working on software is poorly recognised. Without direct support for, and recognition of, this work, packages can become insufficiently maintained to be useful to the community, with very negative impacts upon analysis, and, by implication, on the experiments' physics programmes and the future usability and preservation of results.

Analysts are increasingly encouraged to make analyses and all related artefacts findable, accessible, interoperable and reusable (FAIR) [104–107]. The expertise and tooling required has developed rapidly over recent years. In particular, developments in the domain of workflow management systems [108–111] have enabled analysts to construct analyses in the form of structured workflows. The REANA reusable analysis platform [112] extends this to provide a data analysis platform from which to develop reusable analyses. Increased functionality and user uptake of CI/CD systems provides tooling to ensure the integrity and reliability of analyses and software alike.

Reinterpretation of results is helped greatly by this approach and is itself supported by important tools such as Rivet [113].

A topic of increasing interest for HEP is the importance of open data (for research and education), and data preservation. This requires effort in all aspects of software, but especially analysis, and does not necessarily have a clear end date, even outlasting the experiments which created the data.

5.3 Analysis facilities and data analysis at scale

With the increasing scale of data to be processed in HEP analyses, it is vital that analysts can employ large scale distributed computing resources. Analysis facilities (AFs) provide an approach to analysis infrastructure in which analysts can interface with these resources whilst also being able to prototype and execute their analyses interactively [114]. Demonstrators such as the Analysis Grand Challenge initiative [115], in which physics analyses are performed to test the technologies which will be required for the HL-LHC, and many prototype AFs have shown promise in this model.

5.4 Recommendations

1. Support the continued development of analysis tools, strongly emphasising interoperability and standards, while also exploring new R&D lines.
2. Ensure recognition and secure positions for the development of analysis software and supporting technical contributions that result in increased usability, efficiency and sustainability.
3. Improve training and infrastructure to make our analyses FAIR.

6 Training and careers

6.1 Current status

Present day HEP experimental collaborations have hundreds or thousands of users. They have internal structures and onboarding processes to help new users learn and jump-start software and physics analysis contributions. While most analysis tools and software frameworks taught are experiment-specific, a considerable amount of training is invested in common software tools and skills like Python, C++, ML, CI/CD, and analysis preservation and reproducibility, etc. There is a growing consensus in the HEP community on the need for common training programmes to bring researchers up to date with new software technologies [116–119]. As a result, in the past 5 years the HSF [120], together with Institute for Research and Innovation in Software for High Energy Physics (IRIS-HEP) [121], CERN EP-SFT group and FIRST-HEP [122], and partnering with The Carpentries [123], and PyHEP [124] have developed material for an introductory, intermediate and advanced HEP software curriculum [125] taught to HEP users. In addition, several of our major conferences have tracks on software training and corresponding efforts. Other specific training efforts include CoDAS-HEP [126], Bertinoro [127], CERN School of Computing [128] and CERN OpenLab Software workshops [129]. Thus far, over 3000 users in HEP, related Nuclear Physics and computing areas have been trained. Current training work has also been published in journals and conference proceedings [130–133].

6.2 Challenges and opportunities

While a lot of progress has been made with respect to finding and establishing common training across HEP, several challenges still remain. Currently, each experiment may have its own training programmes that are specific to its needs, but for HEP wide effort, committed funding sources for HEP have been lacking. A central organisation like the HSF is needed to facilitate cooperation and engage people in common training efforts. Incentives and recognition are essential to be able to recruit mentors and tutors who invest time in training. Scalability and sustainability are key to achieve a steady state, where the scale of training activities can match the number of incoming students each year. New opportunities, such as using Large Language Models for training, need to be investigated. Training scope and curricula must evolve to meet the needs of the community. New instructors and materials developers recruited and trained continuously are required to sustain a long-term effort. Commitments from major HEP labs and experiments to workforce training bring significant value. Diversity and inclusion should be a guiding principle to develop the workforce pipeline and should engage the broader community that it represents.

The benefits of investing in training like this are considerable, directly for HEP and for supporting a set of broad interdisciplinary skills that help research software through, e.g., national and institutional research software engineer groups.

6.3 Career support and recognition

For successful HEP computing, training provides a crucial step to recruit our future workforce and provide wide societal benefits. Most training is led by the early career researchers who are trying to build a career. The current mindset favours visibility and recognition of those working on data analysis projects or detector hardware rather than those invested in software and computing. This must change if we want to meet future challenges over decades and in multiple experiments, which increasingly requires large and long-lived software projects. We must provide such opportunities and provide a career path for those involved in software and computing, including in training, by creating job opportunities at labs and universities on par with those involved in detector construction [4]. This must be supported by increased visibility like publications, conferences opportunities and recognition of these contributions as valid scientific accomplishments.

6.4 Recommendations

1. Create training-the-workforce funding opportunities to engage students, postdocs who can be mentors and tutors for the training programmes.
2. Align software training efforts with other research software bodies to maximise impact.
3. Increase job opportunities for those investing significantly in training and computing.

7 Conclusion

This paper describes the software and computing challenges that are critical for the success of HEP experiments. The main motivation for addressing them is the physics output of current and future experiments with their increased data volume and need for precision. However, this also responds to the requirement to make our computing environmentally sustainable. The main way to address these needs is by improving software efficiency. As many aspects of these challenges are common to several, if not all, experiments, global collaboration around software, as supported by the HSF, has been extremely valuable to the work done in the last 10 years. Continuing the work that the HSF undertakes requires a commitment from labs, universities and funding agencies.

To address all these challenges, the main asset is *skilled people* who can make a *viable career* in our field. Addressing this challenge, to recruit, train and retain scientific software developers, is the key message for the future.

Acknowledgements The editors would like to thank the many people from the HEP community who provided input to this document and made possible this summary on the software and computing challenges faced by HEP experiments. We also thank the many projects and organisations that have provided funding to several colleagues, making possible their contributions. In particular the U.S. National Science Foundation (NSF) Cooperative Agreement PHY-2323298 (IRIS-HEP); the OpenMAPP project, via National Science Centre, Poland under CHIST-ERA programme (NCN 2022/04/Y/ST2/00186); Brookhaven National Laboratory, a U.S. Department of Energy, Office of Science facility; Fermi National Accelerator Laboratory (Fermilab), a U.S. Department of Energy, Office of Science, Office of High Energy Physics HEP User Facility (Fermilab is managed by FermiForward Discovery Group, LLC, acting under Contract No. 89243024CSC000002).

Data Availability Statement This manuscript has no associated data. [Author's comment: There are no external data associated with the manuscript.]

Code Availability Statement This manuscript has no associated code/software. [Author's comment: Code/Software sharing not applicable to this article as no code/ software was generated or analysed during the current study.]

References

1. ATLAS Software and Computing HL-LHC Roadmap. Tech. rep. CERN, Geneva (2022). https://cds.cern.ch/record/2802918
2. CMS Offline Software and Computing. CMS Phase-2 Computing Model: Update Document. Tech. rep. CERN, Geneva (2022). https://cds.cern.ch/record/2815292
3. WLCG Environmental Sustainability Workshop (2024). https://indico.cern.ch/event/1450885
4. HEP Software Foundation. A roadmap for HEP software and computing R&D for the 2020s. Comput. Softw. Big Sci. **3**(1) (2019). https://doi.org/10.1007/s41781-018-0018-8. ISSN:2510-2044
5. A. Buckley et al., The HepMC3 event record library for Monte Carlo event generators. Comput. Phys. Commun. **260**, 107310 (2021). https://doi.org/10.1016/j.cpc.2020.107310. ISSN:0010-4655
6. J. Alwall et al., A standard format for Les Houches event files. Comput. Phys. Commun. **176**(4), 300–304 (2007). https://doi.org/10.1016/j.cpc.2006.11.010. ISSN:0010-4655
7. A. Buckley et al., LHAPDF6: parton density access in the LHC precision era. Eur. Phys. J. C **75**(3), 132 (2015). https://doi.org/10.1140/epjc/s10052-015-3318-8. ISSN:1434-6052
8. E. Maguire, L. Heinrich, G. Watt, HEPData: a repository for high energy physics data. J. Phys: Conf. Ser. **898**(10), 102006 (2017). https://doi.org/10.1088/1742-6596/898/10/102006
9. V. Chekhovsky et al., High-precision measurement of the W boson mass with the CMS experiment at the LHC (2024). arXiv:2412.13872 [hep-ex]
10. A. Valassi et al., Challenges in Monte Carlo Event generator software for high-luminosity LHC. Comput. Softw. Big Sci. **5**(1), 12 (2021). https://doi.org/10.1007/s41781-021-00055-1. ISSN:2510-2044
11. F. Maltoni, S. Su, J. Thaler, TF07 Snowmass report: theory of collider phenomena (2022). arXiv:2210.02591 [hep-ph]
12. J.M. Campbell et al., Event generators for high-energy physics experiments. SciPost Phys. **16**, 130 (2024). https://doi.org/10.21468/SciPostPhys.16.5.130
13. J.R. Andersen et al., A positive resampler for Monte Carlo events with negative weights. Eur. Phys. J. C **80**(11), 1007 (2020). https://doi.org/10.1140/epjc/s10052-020-08548-w. arXiv:2005.09375 [hep-ph]
14. J.R. Andersen, A. Maier, Unbiased elimination of negative weights in Monte Carlo samples. Eur. Phys. J. C **82**(5), 433 (2022). https://doi.org/10.1140/epjc/s10052-022-10372-3. arXiv:2109.07851 [hep-ph]

15. R. Frederix et al., On the reduction of negative weights in MC@NLO-type matching procedures. JHEP **07**, 238 (2020). https://doi.org/10.1007/JHEP07(2020)238. arXiv:2002.12716 [hep-ph]
16. J. Alwall et al., The automated computation of tree-level and next-to-leading order differential cross sections, and their matching to parton shower simulations. JHEP **07**, 079 (2014). https://doi.org/10.1007/JHEP07(2014)079. arXiv:1405.0301 [hep-ph]
17. S. Hageboeck et al., Madgraph5_aMC@NLO on GPUs and vector CPUs experience with the first alpha release. EPJ Web Conf. **295**, 11013 (2024). https://doi.org/10.1051/epjconf/202429511013. arXiv:2312.02898 [physics.comp-ph]
18. A. Valassi et al., Developments in performance and portability for MadGraph5_aMC@NLO. PoS **ICHEP2022**, 212 (2022). https://doi.org/10.22323/1.414.0212. arXiv:2210.11122 [physics.comp-ph]
19. Z. Wettersten et al., Acceleration beyond lowest order event generation. An outlook on further parallelism within MadGraph5_aMC@NLO. EPJ Web Conf. **295**, 10001 (2024). https://doi.org/10.1051/epjconf/202429510001. arXiv:2312.07440 [physics.comp-ph]
20. CMS Collaboration, Quantifying the computational speedup with madgraph4gpu for CMS worflow (2024). https://cds.cern.ch/record/2914584
21. E. Bothmann et al., A portable parton-level event generator for the high-luminosity LHC. SciPost Phys. **17**(3), 081 (2024). https://doi.org/10.21468/SciPostPhys.17.3.081. arXiv:2311.06198 [hep-ph]
22. E. Bothmann et al., Event generation with Sherpa 2.2. SciPost Phys. **7**(3), 034 (2019). https://doi.org/10.21468/SciPostPhys.7.3.034. arXiv:1905.09127 [hep-ph]
23. R. Moodie, Optimising hadronic collider simulations using amplitude neural networks. J. Phys: Conf. Ser. **2438**(1), 012149 (2023). https://doi.org/10.1088/1742-6596/2438/1/012149. arXiv:2202.04506 [hep-ph]
24. P. Ilten et al., Modeling hadronization using machine learning. SciPost Phys. **14**(3), 027 (2023). https://doi.org/10.21468/SciPostPhys.14.3.027. arXiv:2203.04983 [hep-ph]
25. C. Bierlich et al., Reweighting Monte Carlo predictions and automated fragmentation variations in Pythia 8. SciPost Phys. **16**(5), 134 (2024). https://doi.org/10.21468/SciPostPhys.16.5.134. arXiv:2308.13459 [hep-ph]
26. A. Ghosh et al., Towards a deep learning model for hadronization. Phys. Rev. D **106**(9), 096020 (2022). https://doi.org/10.1103/PhysRevD.106.096020. arXiv:2203.12660 [hep-ph]
27. J. Chan et al., Fitting a deep generative hadronization model. JHEP **09**, 084 (2023). https://doi.org/10.1007/JHEP09(2023)084. arXiv:2305.17169 [hep-ph]
28. A. Hayrapetyan et al., Reweighting simulated events using machine-learning techniques in the CMS experiment (2024). arXiv:2411.03023 [hep-ex]
29. S. Badger et al., Machine learning and LHC event generation. SciPost Phys. **14**(4), (2023), ed. by A. Butter, T. Plehn, S. Schumann, p. 079. https://doi.org/10.21468/SciPostPhys.14.4.079. arXiv:2203.07460 [hep-ph]
30. S. Jadach, B.F.L. Ward, Z. Was, The precision Monte Carlo event generator KK for two fermion final states in e+e− collisions. Comput. Phys. Commun. **130**, 260–325 (2000). https://doi.org/10.1016/S0010-4655(00)00048-5. arXiv:hep-ph/9912214
31. C.M.C. Calame et al., The BABAYAGA event generator. Nucl. Phys. B Proc. Suppl. **131** (2004), ed. by Marco Incagli, G. Venanzoni, pp. 48–55. https://doi.org/10.1016/j.nuclphysbps.2004.02.008. arXiv:hep-ph/0312014
32. S. Jadach et al., Monte Carlo program BHLUMI-2.01 for Bhabha scattering at low angles with Yennie–Frautschi–Suura exponentiation. Comput. Phys. Commun. **70**, 305–344 (1992). https://doi.org/10.1016/0010-4655(92)90196-6
33. A. Denner et al., Electroweak radiative corrections to e+e→WW→4 fermions in double-pole approximation—the RacoonWW approach. Nucl. Phys. B **587**(1), 67–117 (2000). https://doi.org/10.1016/S0550-3213(00)00511-3. ISSN:0550-3213
34. S. Jadach et al., Multi-photon Monte Carlo event generator KKMCee for lepton and quark pair production in lepton colliders. Comput. Phys. Commun. **283**, 108556 (2023). https://doi.org/10.1016/j.cpc.2022.108556. arXiv:2204.11949 [hep-ph]
35. T. Sjöstrand et al., An introduction to PYTHIA 8.2. Comput. Phys. Commun. **191**, 159–177 (2015). https://doi.org/10.1016/j.cpc.2015.01.024. arXiv:1410.3012 [hep-ph]
36. W. Kilian, T. Ohl, J. Reuter, WHIZARD: simulating multi-particle processes at LHC and ILC. Eur. Phys. J. C **71**, 1742 (2011). https://doi.org/10.1140/epjc/s10052-011-1742-y. arXiv:0708.4233 [hep-ph]
37. E. Bothmann et al., Event generation with Sherpa 3. J. High Energy Phys. **2024**(12), 156 (2024). https://doi.org/10.1007/JHEP12(2024)156. arXiv:2410.22148 [hep-ph]. ISSN:1029-8479
38. S. Frixione et al., Lepton collisions in MadGraph5_aMC@NLO (2021). arXiv:2108.10261 [hep-ph]
39. T. Ohl, CIRCE version 1.0: beam spectra for simulating linear collider physics. Comput. Phys. Commun. **101**, 269–288 (1997). https://doi.org/10.1016/S0010-4655(96)00167-1. arXiv:hep-ph/9607454
40. D. Schulte, Beam-beam simulations with Guinea-Pig. eConf **C980914** (1998), ed. by Kwok Ko, Robert D. Ryne, pp. 127–131. https://cds.cern.ch/record/382453/files/ps-99-014.pdf
41. ECFA. ECFA Higgs/electroweak/top factory study, in ed. by A. Robson, C.C. Leonidopoulos (2025)
42. V. Bertone et al., The partonic structure of the electron at the next-to-leading logarithmic accuracy in QED. JHEP **03**, 135 (2020). https://doi.org/10.1007/JHEP03(2020)135. arXiv:1911.12040 [hep-ph]. [Erratum: JHEP **08**, 108 (2022)]
43. S. Frixione, Initial conditions for electron and photon structure and fragmentation functions. JHEP **11**, 158 (2019). https://doi.org/10.1007/JHEP11(2019)158. arXiv:1909.03886 [hep-ph]
44. S. Frixione et al., Initial state QED radiation aspects for future e^+e^- colliders, in *Snowmass 2021* (2022). arXiv:2203.12557 [hep-ph]
45. F. Krauss, A. Price, M. Schönherr, YFS resummation for future lepton–lepton colliders in SHERPA. SciPost Phys. **13**(2), 026 (2022). https://doi.org/10.21468/SciPostPhys.13.2.026. arXiv:2203.10948 [hep-ph]
46. G. Ganis, C. Helsens, V. Völkl, Key4hep, a framework for future HEP experiments and its use in FCC. Eur. Phys. J. Plus **137**(1), 149 (2022). https://doi.org/10.1140/epjp/s13360-021-02213-1. ISSN:2190-5444
47. A. Sailer et al., Towards a Turnkey Software Stack for HEP experiments. EPJ Web Conf. **245**, 10002 (2020). https://doi.org/10.1051/epjconf/202024510002
48. DUNE Collaboration, DUNE software and computing research and development for European particle physics strategy update (2025). https://docs.dunescience.org/cgi-bin/sso/ShowDocument?docid=33192
49. J.M. Campbell et al., Event generators for high-energy physics experiments (2024). arXiv:2203.11110 [hep-ph]
50. S. Gardiner, J. Isaacson, L. Pickering, NuHepMC: a standardized event record format for neutrino event generators (2024). arXiv:2310.13211 [hep-ph]
51. J. Isaacson et al., Novel event generator for the automated simulation of neutrino scattering. Phys. Rev. D **105**, 096006 (2022). https://doi.org/10.1103/PhysRevD.105.096006

52. J. Allison et al., Recent developments in Geant4. Nucl. Instrum. Methods Phys. Res. Sect. A **835**, 186–225 (2016). https://doi.org/10.1016/j.nima.2016.06.125. ISSN:0168-9002
53. J. Allison et al., Geant4 developments and applications. IEEE Trans. Nucl. Sci. **53**(1), 270–278 (2006). https://doi.org/10.1109/TNS.2006.869826
54. S. Agostinelli et al., GEANT4—a simulation toolkit. Nucl. Instrum. Methods A **506**, 250–303 (2003). https://doi.org/10.1016/S0168-9002(03)01368-8
55. Geant4 Input to the EPPSU (2024). https://indico.cern.ch/event/1475445/
56. C. Ahdida et al., New capabilities of the FLUKA multi-purpose code. Front. Phys. **9** (2022). https://doi.org/10.3389/fphy.2021.788253. ISSN:2296-424X
57. G. Battistoni et al., Overview of the FLUKA code. Ann. Nucl. Energy (Oxf.) **82** (2024). https://doi.org/10.1016/j.anucene.2014.11.007
58. The European Strategy Group, Deliberation document on the 2020 update of the European strategy for particle physics. Tech. rep. Geneva (2020). https://doi.org/10.17181/ESU2020Deliberation. https://cds.cern.ch/record/2720131
59. ATLAS Collaboration, Software and computing for Run 3 of the ATLAS experiment at the LHC (2024). arXiv:2404.06335 [hep-ex]
60. CMS Offline and Computing Public Results (2025). https://twiki.cern.ch/twiki/bin/view/CMSPublic/CMSOfflineComputingResults
61. G. Amadio et al., Offloading electromagnetic shower transport to GPUs. J. Phys: Conf. Ser. **2438**(1), 012055 (2023). https://doi.org/10.1088/1742-6596/2438/1/012055. ISSN:1742-6596
62. AdePT: Accelerated demonstrator of electromagnetic Particle Transport. https://github.com/apt-sim/AdePT
63. S.R. Johnson et al., *Celeritas: Accelerating Geant4 with GPUs*, vol. 295 (Oak Ridge National Laboratory (ORNL), Oak Ridge, 2024). https://doi.org/10.1051/epjconf/202429511005. https://www.osti.gov/biblio/2361202
64. Accelerating detector simulations with Celeritas: performance improvements and new capabilities. https://indi.to/JtRK9
65. AdePT-Enabling GPU electromagnetic transport with Geant4. https://indi.to/k3jCM
66. S. Blyth, Meeting the challenge of JUNO simulation with Opticks: GPU optical photon acceleration via NVIDIA® OptiXTM. EPJ Web Conf. **245**, 11003 (2020). https://doi.org/10.1051/epjconf/202024511003
67. ATLAS Collaboration, AtlFast3: the next generation of fast simulation in ATLAS. Comput. Softw. Big Sci. **6**, 7 (2022). https://doi.org/10.1007/s41781-021-00079-7. arXiv:2109.02551 [hep-ex]
68. D. Kim et al., INN: a method identifying clean-annotated samples via consistency effect in deep neural networks (2021). arXiv:2106.15185 [cs.LG]
69. A. Tong et al., Improving and generalizing flow-based generative models with minibatch optimal transport (2024). arXiv:2302.00482 [cs.LG]
70. S. Hamilton et al., The ATLAS fast track simulation project. Tech. rep. CERN, Geneva (2011). https://cds.cern.ch/record/1341541
71. Garfield++ toolkit. https://garfieldpp.web.cern.ch/garfieldpp/
72. G. Ganis, FCC Physics Workshop: computing resources status and plans (2025). https://indico.cern.ch/event/1439509/contributions/6289563/
73. R. Aaij et al., A comprehensive real-time analysis model at the LHCb experiment. J. Instrum. **14**(04), P04006 (2019). https://doi.org/10.1088/1748-0221/14/04/P04006
74. A. Hayrapetyan et al., Development of the CMS detector for the CERN LHC Run 3. J. Instrum. **19**(05), P05064 (2024). https://doi.org/10.1088/1748-0221/19/05/P05064
75. S. Acharya et al., ALICE upgrades during the LHC Long Shutdown 2. J. Instrum. **19**(05), P05062 (2024). https://doi.org/10.1088/1748-0221/19/05/P05062
76. M. Concas, A vendor-unlocked ITS GPU reconstruction in ALICE. EPJ Web Conf. **295**, 02002 (2024). https://doi.org/10.1051/epjconf/202429502002
77. R. Aaij et al., Allen: a high-level trigger on GPUs for LHCb. Comput. Softw. Big Sci. **4**, 7 (2020). https://doi.org/10.1007/s41781-020-00039-7
78. A. Matthes et al., Tuning and optimization for a variety of many-core architectures without changing a single line of implementation code using the Alpaka library (2017). arXiv:1706.10086
79. G. Aad et al., The ATLAS trigger system for LHC Run 3 and trigger performance in 2022. J. Instrum. **19**(06), P06029 (2024). https://doi.org/10.1088/1748-0221/19/06/P06029
80. CMS Collaboration, Enriching the physics program of the CMS experiment via data scouting and data parking (2024). arXiv:2403.16134 [hep-ex]
81. ATLAS Collaboration, Configuration, performance, and commissioning of the ATLAS *b*-jet triggers for the 2022 and 2023 LHC data-taking periods (2025). arXiv:2501.11420 [hep-ex]
82. CMS Collaboration, Anomaly detection in the CMS global trigger test crate for Run 3 (2023). https://cds.cern.ch/record/2876546
83. FastML Team, HLS4ML. Version v1.0.0 (2024). https://doi.org/10.5281/zenodo.14344847
84. N. Nolte, O. Kitouni, M. Williams, Robust and provably monotonic networks. Mach. Learn.: Sci. Technol. **4**(3), 15 (2023). https://doi.org/10.1088/2632-2153/aced80. arXiv:2112.00038 [physics.ins-det]
85. A. de Oyanguren Campos et al., A streamlined neural model for real-time analysis at the first level of the LHCb trigger (2024). https://indico.cern.ch/event/1338689/contributions/6015440/
86. R. Aaij et al., A comparison of CPU and GPU implementations for the LHCb experiment run 3 trigger. Comput. Softw. Big Sci. **6**(1), 1 (2022). https://doi.org/10.1007/s41781-021-00070-2. arXiv:2105.04031 [physics.ins-det]
87. X. Ai et al., A common tracking software project. Comput. Softw. Big Sci. **6**, 8 (2022). https://doi.org/10.1007/s41781-021-00078-8
88. B. Yeo et al., CTD2022: traccc—GPU Track reconstruction demonstrator for HEP (2023). https://doi.org/10.5281/zenodo.8119769
89. ACTS: demonstrator tracking chain for accelerators (traccc). https://github.com/acts-project/traccc
90. FastML Team, fastmachinelearning/hls4ml. Version v1.0.0 (2024). https://doi.org/10.5281/zenodo.1201549. https://github.com/fastmachinelearning/hls4ml
91. J. Duarte et al., Fast inference of deep neural networks in FPGAs for particle physics. JINST **13**(07), P07027 (2018). https://doi.org/10.1088/1748-0221/13/07/P07027. arXiv:1804.06913 [physics.ins-det]
92. E. Rodrigues et al., The Scikit HEP Project overview and prospects, in *EPJ Web of Conferences*, vol. 245, ed. by C. Doglioni et al. (2020), p. 06028. https://doi.org/10.1051/epjconf/202024506028. ISSN:2100-014X
93. A. Alshehri et al., PyHEP.dev 2024 Workshop Summary Report, August 26–30 2024, Aachen, Germany (2024). arXiv:2410.02112 [hep-ex]
94. P. Danilo et al., RDataFrame: easy parallel ROOT analysis at 100 threads. EPJ Web Conf. **214**, 06029 (2019). https://doi.org/10.1051/epjconf/201921406029
95. M. Zaharia et al., Apache spark: a unified engine for big data processing. Commun. ACM **59**(11), 56–65 (2016). https://doi.org/10.1145/2934664. ISSN:0001-0782
96. M. Rocklin, Dask: parallel computation with blocked algorithms and task scheduling, in *Proceedings of the 14th Python in Sci-*

ence Conference, ed. by K. Huff, J. Bergstra (2015), pp. 126–132. https://doi.org/10.25080/Majora-7b98e3ed-013
97. J. Pivarski, P. Elmer, D. Lange, Awkward arrays in Python, C++, and Numba, in *EPJ Web of Conferences*, vol. 245, ed. by C. Doglioni et al. (2020), p. 05023. https://doi.org/10.1051/epjconf/202024505023. ISSN:2100-014X
98. N. Smith et al., Coffea columnar object framework for effective analysis, in *EPJ Web of Conferences*, vol. 245, ed. by C. Doglioni et al. (2020), p. 06012. https://doi.org/10.1051/epjconf/202024506012. ISSN:2100-014X
99. J. Blomer et al., ROOT's RNTuple I/O subsystem: the path to production. EPJ Web Conf. **295**, 06020 (2024). https://doi.org/10.1051/epjconf/202429506020
100. The HS3 Committee, HS3-HEP Statistics Serialization Standard. https://github.com/hep-statistics-serialization-standard/hep-statistics-serialization-standard
101. ATLAS Collaboration, Measurement of off-shell Higgs boson production in the $H^* \to ZZ \to 4\ell$ decay channel using a neural simulation-based inference technique in 13 TeV *pp* collisions with the ATLAS detector. Rep. Prog. Phys. **88**(5), 057803 (2025). https://doi.org/10.1088/1361-6633/adcd9a. ISSN:1361-6633
102. The ATLAS Collaboration, An implementation of neural simulation-based inference for parameter estimation in ATLAS. Rep. Prog. Phys. **88**(6), 067801 (2025). https://doi.org/10.1088/1361-6633/add370
103. S. Hageboeck, What the new RooFit can do for your analysis. PoS **ICHEP2020**, 910 (2021). https://doi.org/10.22323/1.390.0910. arXiv:2012.02746 [physics.data-an]
104. M.D. Wilkinson et al., The FAIR guiding principles for scientific data management and stewardship. Sci. Data **3**(1), 160018 (2016). https://doi.org/10.1038/sdata.2016.18. ISSN:2052-4463
105. Y. Chen et al., A FAIR and AI-ready Higgs boson decay dataset. Sci. Data **9**, 31 (2022). https://doi.org/10.1038/s41597-021-01109-0
106. J. Duarte et al., FAIR AI models in high energy physics. Mach. Learn. Sci. Technol. **4**(4), 045062 (2023). https://doi.org/10.1088/2632-2153/ad12e3. arXiv:2212.05081 [hep-ex]
107. E.A. Huerta et al., FAIR for AI: an interdisciplinary and international community building perspective. Sci. Data **10**(1), 487 (2023). https://doi.org/10.1038/s41597-023-02298-6. ISSN:2052-4463
108. M.R. Crusoe et al., Methods included: standardizing computational reuse and portability with the Common Workflow Language. Commun. ACM **65**(6), 54–63 (2022). https://doi.org/10.1145/3486897. ISSN:0001-0782
109. F. Mölder et al. Sustainable data analysis with Snakemake [version 2; peer review: 2 approved]. F1000Research **10**(33) (2021). https://doi.org/10.12688/f1000research.29032.2
110. M. Rieger, End-to-end analysis automation over distributed resources with Luigi analysis workflows, in *EPJ Web of Conferences*, vol. 295, ed. by R. De Vita et al. (2024), p. 05012. https://doi.org/10.1051/epjconf/202429505012. ISSN:2100-014X
111. K. Cranmer, L. Heinrich, Yadage and Packtivity—analysis preservation using parametrized workflows. J. Phys: Conf. Ser. **898**(10), 102019 (2017). https://doi.org/10.1088/1742-6596/898/10/102019
112. T. Šimko et al., REANA: a system for reusable research data analyses. EPJ Web Conf. **214**, 06034 (2019). https://doi.org/10.1051/epjconf/201921406034
113. C. Bierlich et al., Robust independent validation of experiment and theory: rivet version 3. SciPost Phys. **8**, 026 (2020). https://doi.org/10.21468/SciPostPhys.8.2.026
114. D. Ciangottini et al., Analysis facilities white paper (2024). arXiv:2404.02100 [hep-ex]
115. A. Held, O. Shadura, The IRIS-HEP analysis grand challenge. PoS **ICHEP2022**, 235 (2022). https://doi.org/10.22323/1.414.0235
116. HEP Software Foundation et al., HEP software foundation community white paper working group—training, staffing and careers (2019). arXiv:1807.02875 [physics.ed-ph]
117. Snowmass 2021—Community Engagement Frontier (2022). https://www.slac.stanford.edu/econf/C210711/Engagement.html
118. K.A. Assamagan et al., Snowmass 2021—community engagement frontier report (2022). arXiv:2211.13210 [physics.soc-ph]
119. J. Hogan, A. Karadzhinova-Ferrer, S. Malik, Summary report of the topical group on career pipeline and development (CommF2) community engagement frontier Snowmass 2021 (2022). arXiv:2209.10114 [physics.ed-ph]
120. HSF Training Activity. https://hepsoftwarefoundation.org/activities/training.html
121. IRIS-HEP Training, Education and Outreach. https://iris-hep.org/ssc.html
122. Framework for Integrated Research Software Training in High Energy Physics. https://first-hep.org
123. The Carpentries. https://carpentries.org
124. HSF PyHEP—Python in HEP. https://hepsoftwarefoundation.org/activities/pyhep.html
125. HSF Training Center. https://hsf-training.org/training-center/
126. Computational and Data Science for High Energy Physics. https://codas-hep.org
127. INFN Efficient Scientific Computing School. https://esc.infn.it
128. CERN School of Computing. https://csc.web.cern.ch
129. CERN OpenLab Software Workshops. https://indico.cern.ch/category/3169/
130. S. Malik et al., Train to sustain. EPJ Web Conf. **295**, 05023 (2024). https://doi.org/10.1051/epjconf/202429505023
131. S. Malik et al., Software training outreach in HEP. EPJ Web of Conf. **295**, 08018 (2024). https://doi.org/10.1051/epjconf/202429508018
132. S. Malik et al., Software training in HEP. Comput. Softw. Big Sci. **5**, 22 (2021). https://doi.org/10.1007/s41781-021-00069-9
133. A. Reinsvold Hall et al., Training and onboarding initiatives in high energy physics experiments. Front. Big Data **8** (2025). https://doi.org/10.3389/fdata.2025.1497622. ISSN:2624-909X