The LHCb ultra-fast simulation option, Lamarr

design and validation

Lucio Anderlini[1], *Matteo* Barbetti[1,2,*], *Simone* Capelli[3,4], *Gloria* Corti[5], *Adam* Davis[6], *Denis* Derkach[7], *Nikita* Kazeev[7], *Artem* Maevskiy[7], *Maurizio* Martinelli[3,4], *Sergei* Mokonenko[7], *Benedetto G.* Siddi[8], and *Zehua* Xu[9] on behalf of the LHCb Simulation Project

[1]Istituto Nazionale di Fisica Nucleare (INFN), Sezione di Firenze, Italy
[2]Department of Information Engineering (DINFO), University of Firenze, Italy
[3]Istituto Nazionale di Fisica Nucleare (INFN), Sezione di Milano-Bicocca, Italy
[4]Department of Physics, University of Milano-Bicocca, Italy
[5]European Organization for Nuclear Research (CERN), Switzerland
[6]Department of Physics and Astronomy, University of Manchester, United Kingdom
[7]Affiliated with an institute covered by a cooperation agreement with CERN
[8]Department of Physics, University of Ferrara, Italy
[9]Laboratoire de Physique de Clermont (LPC), Université Clermont Auvergne, France

Abstract. Detailed detector simulation is the major consumer of CPU resources at LHCb, having used more than 90% of the total computing budget during Run 2 of the Large Hadron Collider at CERN. As data is collected by the upgraded LHCb detector during Run 3 of the LHC, larger requests for simulated data samples are necessary, and will far exceed the pledged resources of the experiment, even with existing fast simulation options. An evolution of technologies and techniques to produce simulated samples is mandatory to meet the upcoming needs of analysis to interpret signal versus background and measure efficiencies. In this context, we propose Lamarr, a Gaudi-based framework designed to offer the fastest solution for the simulation of the LHCb detector. Lamarr consists of a pipeline of modules parameterizing both the detector response and the reconstruction algorithms of the LHCb experiment. Most of the parameterizations are made of Deep Generative Models and Gradient Boosted Decision Trees trained on simulated samples or alternatively, where possible, on real data. Embedding Lamarr in the general LHCb Gauss Simulation framework allows combining its execution with any of the available generators in a seamless way. Lamarr has been validated by comparing key reconstructed quantities with Detailed Simulation. Good agreement of the simulated distributions is obtained with two-order-of-magnitude speed-up of the simulation phase.

1 Introduction

The LHCb experiment [1] has been originally designed to study rare decays of particles containing b and c quarks produced at the Large Hadron Collider (LHC). The LHCb detector is a single-arm forward spectrometer covering the pseudorapidity range of $2 < \eta < 5$, that

*e-mail: matteo.barbetti@cern.ch

includes a Tracking system and a Particle Identification (PID) system [2]. The Tracking system provides high-precision measurements of the momentum p of charged particles and the position of primary vertices. Different types of charged hadrons are separated using the response of two ring-imaging Cherenkov (RICH) detectors. Photons, electrons and hadrons are identified by the calorimeter system relying on an electromagnetic calorimeter (ECAL) and a hadron calorimeter (HCAL). Finally, a dedicated system named MUON identifies muons alternating layers of iron and multi-wire proportional chambers. The RICH, calorimeters and MUON detectors are part of the PID system.

Interpreting signal, rejecting background contributions and performing efficiency studies requires to have a full understanding of its data sample, from the high-energy collisions to the set of physics processes responsible for the detector high-level response. This kind of studies greatly benefits from the use of simulated samples. At LHCb, the simulation production mainly relies on the Gauss framework [3] that implements the generation and simulation phases, and is based on the Gaudi processing framework [4]. The high-energy collisions and all the physics processes that produce the set of particles (e.g., muons, pions, kaons or protons) able to traverse the LHCb spectrometer are simulated during the *generation phase* using software like Pythia8 [5] and EvtGen [6]. The radiation-matter interactions between the detector materials and the traversing particles are reproduced during the *simulation phase* that aims to compute the energy deposited in the active volumes and relies on the Geant4 toolkit [7]. Then, a separate application converts the energy deposits into raw data compatible with the real one collected by LHCb.

The simulation of all the physics events occurring within the detector is the major consumer of CPU resources at LHCb, having used more than 90% of the total computing budget during LHC Run 2. The upgraded version of the experiment is designed to collect one-order-of-magnitude larger data samples during Run 3. Meeting the upcoming and future requests for simulated samples is not sustainable relying only on the traditional *detailed simulation*. For this reason, the LHCb Collaboration is spending great efforts in modernizing the simulation software stack through the novel experiment-independent framework Gaussino[1] [8, 9] on which a newer version of Gauss will be built on, and in developing faster simulation options, some of which also powered by machine learning algorithms [10–13].

2 Fast simulation VS. ultra-fast simulation

Simulating all the physics processes of interest for LHCb is extremely expensive in terms of computing resources, especially the Geant4-based step that is the major CPU-consumer. Speeding up the computation of the energy deposits or, more generally, the detector response is mandatory to satisfy the demand for simulations expected for Run 3 and those that will follow. Actually, this is a shared problem across the High Energy Physics (HEP) community that is collectively facing it, including by exploiting the latest achievements in Computer Science and adapting *deep generative models* to parameterize the low-level response of the various experiments [14–16]. The literature refers to this kind of strategies with the term *fast simulation*. Fast simulations share their data processing scheme and the reconstruction step with the detailed simulation (as depicted in Figure 1), and are proven capable of reducing the computation cost of a simulated sample up to a factor of 20.

To meet the upcoming and future requests for simulated samples, the LHCb Collaboration is also considering a more radical approach based on the so-called *ultra-fast simulation* paradigm. In this case, the aim is to directly reproduce the high-level response of the detector relying on a set of parameterizations developed to transform generator-level particles

[1] Visit https://gaussino.docs.cern.ch for additional details.

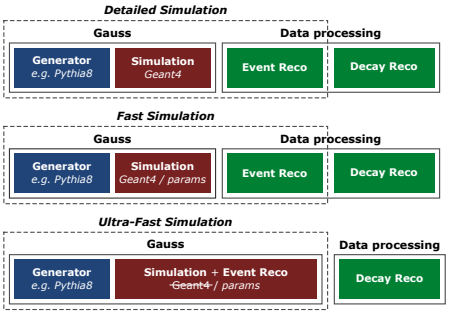

Figure 1. Schematic representation of the data processing flow in the *detailed* (top), *fast* (center) and *ultra-fast* (bottom) simulation paradigms.

information into reconstructed physics objects as schematically represented in Figure 1 (bottom). Such parameterizations can still be built using generative models, like *Generative Adversarial Networks* (GAN), proven to succeed in reproducing the high-level response of the LHCb detector [17] and offering reliable synthetic simulated samples [18]. Following pioneering studies on the ultra-fast simulation of the electromagnetic calorimeter based on GANs [19], the CMS Collaboration has recently started developing a full-scope ultra-fast simulation based on *Normalizing Flow*, named FlashSim [20].

3 Lamarr: the LHCb ultra-fast simulation framework

Lamarr [12, 13] is the official ultra-fast simulation framework for LHCb, able to offer the fastest options for simulation. Originating from the attempt of an LHCb customized version of Delphes [21, 22], Lamarr is an independent project retaining only the inspiration of its modular layout from Delphes. In particular, the Lamarr framework consists of a pipeline of modular parameterizations, most of which based on machine learning algorithms, designed to take as input the particles generated by the physics generators and provide as output the high-level response of the various LHCb sub-detectors.

The Lamarr pipeline can be logically split in two separated chains according to the charge of the generated particles. We expect that charged particles leave a mark in the Tracking system that Lamarr characterizes in terms of acceptance, efficiency and resolution as described in Section 3.1. The reconstructed tracking variables are then used to compute the response of the PID system for a set of traversing charged particles (muons, pions, kaons or protons) as detailed in Section 3.2. In case of neutral particles (e.g., photons), the calorimeters play a key role and, since multiple photons can concur to the energy of a single calorimetric cluster, parameterizing particle-to-particle correlation effects is of major relevance. The solutions under investigation are reported in Section 3.3. The Lamarr pipelines described above are shown in Figure 2.

3.1 Tracking system

One of the aims of the LHCb Tracking system is to measure the momentum p of charged particles (i.e., electrons, muons, pions, kaons and protons), exploiting the deflection of their trajectories due to the dipole magnet located in between the tracking detectors. Hence, the first step of the *charged chain* reported in Figure 2 is the propagation through the magnetic field of the particles provided by the physics generators. Lamarr parameterizes the particle trajectories as two rectilinear segments with a single deflection point (inversely proportional to the transverse momentum p_T), implementing the so-called *single p_T kick* approximation.

The next step requires to select the subset of tracks that fall within the LHCb geometrical acceptance and that have any chance to be reconstructed. To this end, Lamarr uses *Gradient*

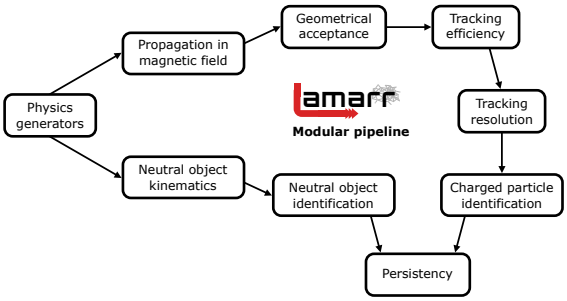

Figure 2. Scheme of the LAMARR modular pipeline. According to the charge of the particle provided by the physics generator, two sets of parameterizations are defined: the charged particles are passed through the Tracking and PID models, while the neutral ones follow a different path where the calorimeter modeling plays a key role.

Boosted Decision Trees (GBDT) trained to learn the fraction of candidates that are in the acceptance as a function of the kinematic information provided by the physics generators. Given a generated track in acceptance, we ask whether the latter will be reconstructed and, in case of positive answer, which tracking detectors are involved in the reconstruction procedure. LAMARR statistically infers such information, namely the tracking efficiency, relying on *neural networks* trained to perform a multi-class classification according to the track kinematics. A major effort is ongoing to improve the performance of the efficiency model on the basis of the type of tracks and particle species (i.e., electrons, muons or hadrons).

At this point, LAMARR disposes of the subset of the generated particles that can be considered as reconstructed tracks, but their kinematics and geometry are still identical to those provided by the physics generators. The smearing of these features, mimicking the effect of the reconstruction, is achieved using GANs. Driven by a *binary cross-entropy* loss function and powered by *skip connections*, GANs succeed in describing the resolution effects due to, for example, multiple scattering phenomena, only relying on track kinematic information at generator-level as input conditions. A similar GAN-based architecture is used to provide the correlation matrix obtained from the Kalman filter adopted in the reconstruction algorithm to define the position, slope and curvature of each track.

Stacking the parameterizations described above, LAMARR is able to provide the high-level response of the LHCb Tracking system. The resulting reconstructed quantities can be further processed using the LHCb analysis software to combine the parameterized tracks into decay candidates as depicted by the green slot in Figure 1 (bottom).

3.2 Particle identification system

To accomplish the LHCb physics program, disposing of a high-performance PID system is crucial since it allows for discriminating the various particle species that traverse the detector. LAMARR provides parameterizations for the majority of the charged particles for which the PID detectors are relevant (i.e., muons, pions, kaons or protons). Specialized parameterizations for the electrons, encoding the multiple scattering and Bremsstrahlung emission contributions in the interaction with the detector materials, is planned as future development.

Identifying these subset particles involves mainly the RICH and MUON detectors, while the role played by the calorimeters is minor. In general, we expect that the response of the PID system depends only on the specie of the traversing particle, its kinematics, and the detector occupancy. According to these dependencies, LAMARR provides the high-level response for both the detectors using GAN-based models properly conditioned [11, 18]. Given the particle specie from the physics generators, its kinematic information results from the LAMARR Tracking modules, while the detector occupancy is described by the total number of tracks traversing the detector.

In real data, the combination of the responses from RICH detectors, calorimeters, MUON system and a binary muon-identification criterion implemented via FPGA and named `isMuon` allows to compute the higher-level response of the PID system, referred to as GlobalPID variables. The parameterization of the GlobalPID variables still relies on conditioned GANs, adding as input what results from the `RichGAN` and `MuonGAN` models. The binary output of a neural-network-based implementation of `isMuon` is used as additional input feature, while no explicit calorimeters contribution is defined leaving the missing information problem to the generator *latent space*.

GAN-based models, driven by a *Wasserstein distance* loss function and trained using a Lipschitz-constrained discriminator [23], succeed in describing the high-level response of the RICH and MUON systems. Chaining together different GANs, LAMARR is also able to provide the higher-level response of the LHCb PID system, injecting an implicit contribution from the calorimeters.

3.3 Electromagnetic calorimeter

Providing a parameterization for the electrons requires describing the response to Bremsstrahlung photons by the LHCb ECAL detector. Since interested by a multitude of secondary particles, the detailed simulation of the calorimeter system is the most computationally expensive step in the simulation pipeline. The latter is a shared problem across the HEP community, that is investing great efforts in tuning deep generative models to properly parameterize the energy deposited in the calorimeter cells [10, 14–16]. Such studies belong to the fast-simulation paradigm that aims to reduce the GEANT4 use, providing models for the low-level response of the various experiments.

The current version of LAMARR provides a simplified parameterization for the LHCb calorimeter, designed for detector studies and based on a fast-simulation approach. Disposing information at the calorimeter cell level requires running reconstruction algorithms to obtain analysis-level quantities that may become rather CPU-expensive for high-multiplicity events. In addition, since non-physical strategies are used to simulate the energy deposits (as is the case for GANs), there is no certainty that the reconstruction software stack can correctly reproduce the expected distributions for the high-level variables [24]. Hence, the LAMARR project is actively working to provide an ultra-fast solution for the ECAL detector.

Reproducing the calorimeter high-level response is a non-trivial task since traditional generative models rely on the hypothesis that an unambiguous relation between the generated particle and the reconstructed object exists[2]. Instead, the presence of merged π^0 and Bremsstrahlung photons may lead to having n generated particles responsible for m reconstructed objects (in general with $n \neq m$). A strategy to face this particle-to-particle correlation problem can be built using techniques designed in the context of Language Modeling, describing the calorimeter simulation as a *translation problem*. To this end, *Graph Neural Network* (GNN) [25] and *Transformer* [26] models are currently under investigation.

Both the models are designed to process a sequence of n generated photons and infer the kinematics of a sequence of m reconstructed clusters. The non-trivial correlations between any particles of the source sequence (photons) and the target one (clusters) rely on the *attention mechanism* [26, 27]. To improve the quality of the resulting parameterizations, the training of both GNN and Transformer-based models is driven by an adversarial procedure (similarly to what occurs for GANs). The discriminator is currently implemented through a *Deep Sets* model [28], while further studies are ongoing to replace it with a second Transformer [29]. Considering the complexity of the problem, the preliminary results are promising as depicted in Figure 3, where the joint action of Transformer and Deep Sets succeeds

[2] To a first approximation, the response of the Tracking and PID systems satisfy this condition.

in deriving the energy distribution on the ECAL face. The center of the calorimeter has not active material since is used to host the LHC beam pipe. It should be pointed out that no constraints are applied to the model output to reproduce such conditions, and that the empty space shown in Figure 3 (right) is the result of the adversarial training procedure.

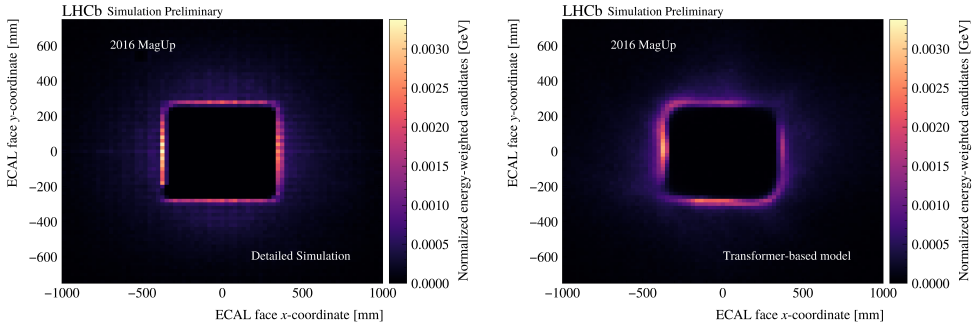

Figure 3. Distribution of the (x, y)-position of the reconstructed clusters on the LHCb ECAL face for a 2000×1500 mm^2 frame placed around the center. The geometrical information is combined with the energy signature properly weighting each bin entry. What obtained from detailed simulation is reported on the left, while the predictions of an adversarial trained Transformer model is shown on the right. The corresponding LHCB-FIGURE is in preparation.

4 Validation campaign and timing performance

The ultra-fast philosophy at the base of the LAMARR framework is being validated by comparing the distributions obtained from machine-learnt models trained on detailed simulation and the ones resulting from standard simulation strategies. In particular, we will briefly discuss the validation studies performed for the charged particles pipeline using simulated $\Lambda_b^0 \to \Lambda_c^+ \mu^- \bar{\nu}_\mu$ decays with $\Lambda_c^+ \to pK^-\pi^+$. The semileptonic nature of the Λ_b^0 decay requires an interface with dedicated generators, in this case EVTGEN. Deeply studied by LHCb, this decay channel includes in its final state the four charged particle species parameterized in the current version of LAMARR, namely muons, pions, kaons and protons.

The validation of the LAMARR Tracking modules is depicted in Figure 4 (left) where the agreement between the Λ_c^+ invariant mass distribution resulting from the ultra-fast paradigm and the one obtained from detailed simulation proves that the decay dynamics is well reproduced and the resolution effects correctly parameterized. To show the good performance of the LAMARR PID models, a comparison between the selection efficiencies for a tight requirement on a multivariate proton classifier is shown in Figure 4 (right).

Comparing the CPU time spent per event by GEANT4-based production of $\Lambda_b^0 \to \Lambda_c^+ \mu^- \bar{\nu}_\mu$ samples and the one needed by LAMARR, we estimate a CPU reduction of two-order-of-magnitude only for the simulation phase. Interestingly, since the generation of b-baryons is exceptionally expensive, PYTHIA8 becomes the major consumer of CPU resources in the ultra-fast paradigm. A further speed-up can be reached reducing the cost for generation, for example using a *Particle Gun* that simulates directly the signal particles without going through the high-energy collisions, not needed since LAMARR parameterizes the detector occupancy. Even in these physics-simplified settings, the ultra-fast philosophy succeeds in reproducing thee distributions obtained from detailed simulation [12].

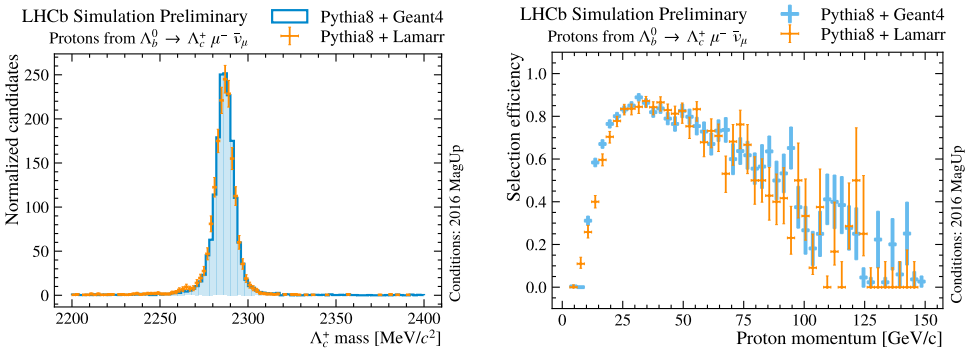

Figure 4. Validation plots for $\Lambda_b^0 \to \Lambda_c^+ \mu^- \bar{\nu}_\mu$ decays with $\Lambda_c^+ \to pK^-\pi^+$ simulated with Pythia8, EvtGen and Lamarr (orange markers) and compared with detailed simulation samples relying on Pythia8, EvtGen and Geant4 (cyan shaded histogram). Reproduced from LHCB-FIGURE-2022-014.

5 Integration with the LHCb simulation framework

To be integrated within the LHCb software stack, the parameterizations provided by Lamarr need to be queried from a C++ application, running in the Gaudi framework. Traditional deployment strategies were found to lead to unacceptably large overheads due to the presence of different multi-threading schedulers and context switching issues. Hence, a custom deployment strategy was preferred: models trained with `scikit-learn` and Keras are converted into compatible C code using the `scikinC` toolkit [30], and then distributed through the LHCb Computing Grid via the CERN VM file-system (`cvmfs`) [31].

The modular layout of Lamarr enables a variety of studies and developments on the single parameterizations, providing a unique and shared infrastructure for validation and performance measurements. While crucial for applications within LHCb, the integration with Gaudi and Gauss makes the adoptions of Lamarr unappealing for researchers outside of the LHCb community. The SQLamarr package[3] aims to mitigate this problem, providing a stand-alone ultra-fast simulation framework with minimal dependencies. Based on SQLite3, SQLamarr provides a set of classes and functions for loading data from physics generators and defining pipelines from compiled models. An integration between SQLamarr and Gaussino is currently under investigation with the aim of providing ultra-fast parameterizations following the experiment-independent philosophy of the newest LHCb simulation framework, named Gauss-on-Gaussino[4] [8, 9].

6 Conclusion

An evolution of the LHCb software stack and the simulation techniques are mandatory to meet the upcoming and future demand for simulated samples expected for Run 3 and those that will follow. Ultra-fast-based solutions will play a key role in reducing the pressure on pledged CPU resources, without compromising unreasonably the description of the uncertainties introduced in the detection and reconstruction phases. Such techniques, powered by deep generative models, are provided to LHCb via the novel Lamarr framework. Well integrated with the physics generators within the Gauss framework, Lamarr delivers two

[3] Visit https://lamarrsim.github.io/SQLamarr for additional details.
[4] Visit https://lhcb-gauss.docs.cern.ch/Futurev5 for additional details.

pipelines according to the charge of the generated particle. The statistical models for the Tracking and the charged PID systems have been deployed and validated with satisfactory results on $\Lambda_b^0 \to \Lambda_c^+ \mu^- \bar{\nu}_\mu$ decays. Several models are currently under investigation for the neutral pipeline, where the translation problem approach offers a viable solution to face the particle-to-particle correlation problem. Further development of the integration between Lamarr and the LHCb simulation framework is one of the major ongoing activities to put the former in production and make its parameterizations available to the HEP community.

Acknowledgements

This work is partially supported by ICSC – Centro Nazionale di Ricerca in High Performance Computing, Big Data and Quantum Computing, funded by European Union – NextGenerationEU.

References

[1] A.A. Alves, Jr. et al. (LHCb), JINST **3**, S08005 (2008)
[2] R. Aaij et al. (LHCb), Int. J. Mod. Phys. A **30**, 1530022 (2015), 1412.6352
[3] M. Clemencic et al. (LHCb), J. Phys. Conf. Ser. **331**, 032023 (2011)
[4] G. Barrand et al., Comput. Phys. Commun. **140**, 45 (2001)
[5] T. Sjostrand, S. Mrenna, P.Z. Skands, Comput. Phys. Commun. **178**, 852 (2008), 0710.3820
[6] D.J. Lange, Nucl. Instrum. Meth. A **462**, 152 (2001)
[7] J. Allison et al., IEEE Trans. Nucl. Sci. **53**, 270 (2006)
[8] M. Mazurek, G. Corti, D. Müller, Comput. Inform. **40**, 815–832 (2021), 2112.04789
[9] M. Mazurek, M. Clemencic, G. Corti, PoS **ICHEP2022**, 225 (2023)
[10] V. Chekalina et al., EPJ Web Conf. **214**, 02034 (2019), 1812.01319
[11] A. Maevskiy et al. (LHCb), J. Phys. Conf. Ser. **1525**, 012097 (2020), 1905.11825
[12] L. Anderlini et al., PoS **ICHEP2022**, 233 (2023)
[13] M. Barbetti (2023), ACAT'22, 2303.11428
[14] M. Paganini, L. de Oliveira, B. Nachman, Phys. Rev. D **97**, 014021 (2018), 1712.10321
[15] C. Krause, D. Shih, Phys. Rev. D **107**, 113003 (2023), 2106.05285
[16] O. Amram, K. Pedro (2023), 2308.03876
[17] F. Ratnikov et al., Nucl. Instrum. Meth. A **1046**, 167591 (2023)
[18] L. Anderlini et al. (LHCb), J. Phys. Conf. Ser. **2438**, 012130 (2023), 2204.09947
[19] P. Musella, F. Pandolfi, Comput. Softw. Big Sci. **2**, 8 (2018), 1805.00850
[20] F. Vaselli et al. (CMS), Tech. rep. (2023), https://cds.cern.ch/record/2858890
[21] J. de Favereau et al. (DELPHES 3), JHEP **02**, 057 (2014), 1307.6346
[22] B.G. Siddi, EPJ Web Conf. **214**, 02024 (2019)
[23] D. Terjék (2020), ICLR'20, 1907.05681
[24] A. Rogachev, F. Ratnikov, J. Phys. Conf. Ser. **2438**, 012086 (2023), 2207.06329
[25] F. Scarselli et al., IEEE Trans. Neural. Netw. **20**, 61 (2009)
[26] A. Vaswani et al. (2017), NeurIPS'17, 1706.03762
[27] S. Brody, U. Alon, E. Yahav (2022), ICLR'22, 2105.14491
[28] M. Zaheer et al. (2017), NeurIPS'17, 1703.06114
[29] K. Lee et al. (2022), ICLR'22, 2107.04589
[30] L. Anderlini, M. Barbetti, PoS **CompTools2021**, 034 (2022)
[31] P. Buncic et al., J. Phys. Conf. Ser. **219**, 042003 (2010)