PROCEEDINGS OF SCIENCE

PoS(EPS-HEP2025)077

CMS FlashSim: how ML powers end-to-end simulation for HEP

Francesco Vaselli,[b,c,*] **Filippo Cattafesta,**[b,c] **Patrick Asenov,**[a,c] **Andrea Rizzi**[a,c] **and on behalf of the CMS Collaboration**

[a] *University of Pisa*

[b] *Scuola Normale Superiore Pisa*

[c] *INFN Sezione di Pisa*

E-mail: francesco.vaselli@cern.ch

The CMS Collaboration developed an end-to-end ML based simulation that can speed up the time for production of analysis samples of several orders of magnitude with a limited loss of accuracy. Detailed event simulation at the LHC is crucial for physics analyses and it is currently taking a large fraction of computing budget. Because the CMS experiment is adopting a common analysis level format (the NANOAOD) for a larger number of analyses, such an event representation is used as the target of this ultra fast simulation, which we call FlashSim. Generator level events, from PYTHIA or other generators, are directly translated into NANOAOD events at several hundred Hz rate with FlashSim. We show how training FlashSim on a limited number of full simulation events is sufficient to achieve very good accuracy on larger datasets for processes not seen at training time. With this work, we aim to update the community about recent and relevant developments behind the FlashSim framework.

The European Physical Society Conference on High Energy Physics (EPS-HEP2025)
7-11 July 2025
Marseille, France

[*] Speaker

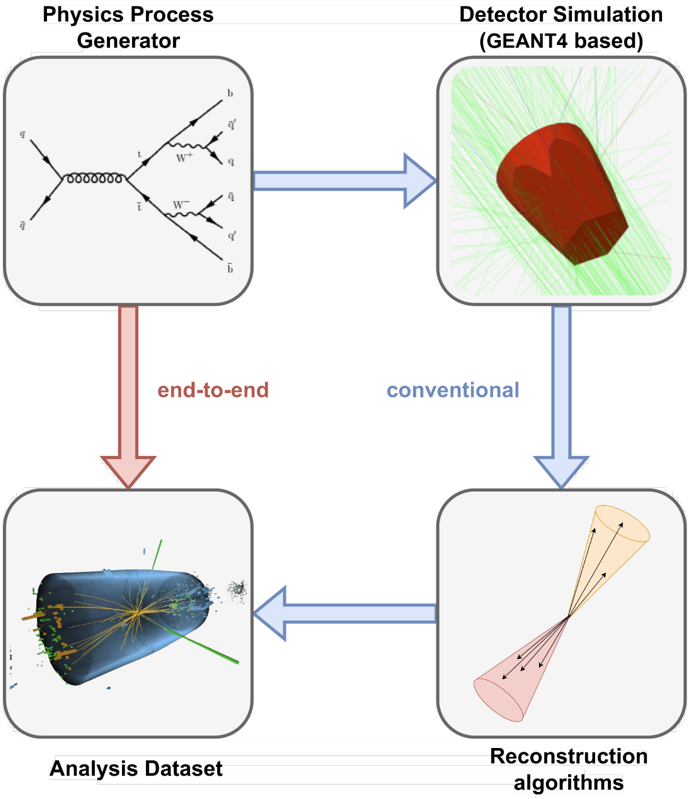
Physics Process Generator
Detector Simulation (GEANT4 based)
end-to-end
conventional
Analysis Dataset
Reconstruction algorithms

Figure 1: Different paths to a NANOAOD simulation via computing demanding full simulation or with end-to-end approach discussed in this report. From [8].

1. Introduction

In this report we summarize the current status of CMS FlashSim, an end-to-end simulation framework developed for the CMS experiment [1] at the Large Hadron Collider (LHC). This framework leverages Machine Learning (ML) techniques to address the computational challenges associated with event simulation ([3],[4]). The computational cost of conventional simulation approaches is significant, and this cost is only exacerbated by the increasing event numbers and single-event complexity anticipated in the High-Luminosity LHC (HL-LHC) era. The growing resource requirements make the investigation into ML-based alternative methods a high priority.

In this first attempt to realize an end-to-end simulation (see Figure 1) with ML we decided to target the simpler format of NANOAOD (1-2 Kb/event of information) so that the ML models would need to learn a limited number of correlations between top level quantities such as particle kinematics and output of taggers and ID algorithms used to discriminate between different particles. The availability of such a simple format is hence one of the enabling elements for the implementation of FlashSim.

2. Framework and Algorithms

The CMS FlashSim framework aims to provide a fast, end-to-end simulation method, and it is based on the following aspects: The framework should provide analysis agnostic simulation by

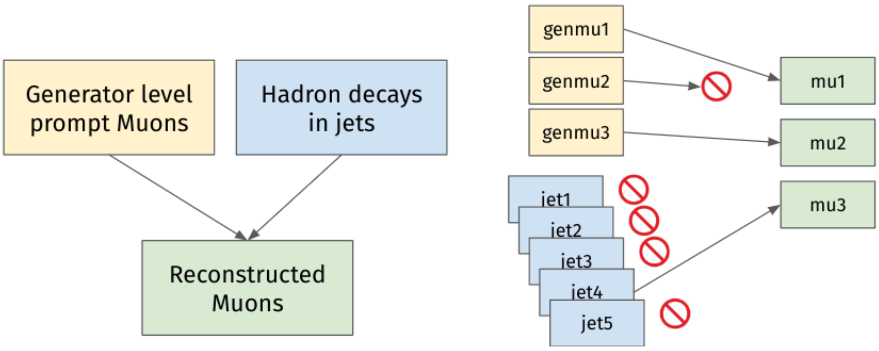

Figure 2: FlashSim architecture and object sources example. Muons can be generated from signal prompt muons or from decays/misreconstructions from jets. For each generated muon and each jet the efficiency model is asked the probability to yield a reconstructed muon and such a probability is compared with a random number uniformly sampled in [0,1]. If the probability is higher than the thrown value, a reconstructed object is generated and its properties are produced by the properties-model.

targeting the NANOAOD format as its baseline. In addition, it is designed to be independent of specific samples, learning the detector's response to different types of generated particles. A key objective is to be orders of magnitude faster than existing simulation techniques while maintaining a reasonable level of accuracy.

A reconstructed object can originate from multiple sources, such as a genuine signal, or from particles with similar signatures, detector interactions, or fake objects. FlashSim is hence designed to model each of these cases separately, where every object is handled in two steps: an efficiency model, which dictates if an object will be reconstructed, and a properties model, which generates the individual properties and features of the reconstructed object (see Figure 2). Variables for each object should display good agreement in both 1D distributions and correlations. For this reason we explored a recently developed technique in the field of Generative AI, the so called Flow Matching [9] regime for training Flow Matching models. We don't discuss this approach here in detail, but we instead refer the interested reader to the relevant reference.

Efficiency models are trained as simple binary classifiers, with cross-entropy loss. At inference time, the output is interpreted as the probability that a given input object is reconstructed and a simple uniform random number is generated and compared with such a probability (see Figure 3).

The current FlashSim prototype covers all object properties for the majority of the NANOAOD collections. For each object the main sources of *signals* and *backgrounds* are considered as starting point. A list is shown in Table 1.

3. Results

3.1 Training

The results presented in this report are obtained with a model trained using a mixture of samples with different signatures, covering the corners of the phase space, with a total of 4M events. The training dataset includes $t\bar{t}$, Drell-Yan (with $100 < H_T < 200$ GeV or with 2 jets and $200 < M_{ll} < 1400$ GeV), and a set of signal models spanning double Higgs production with Higgs

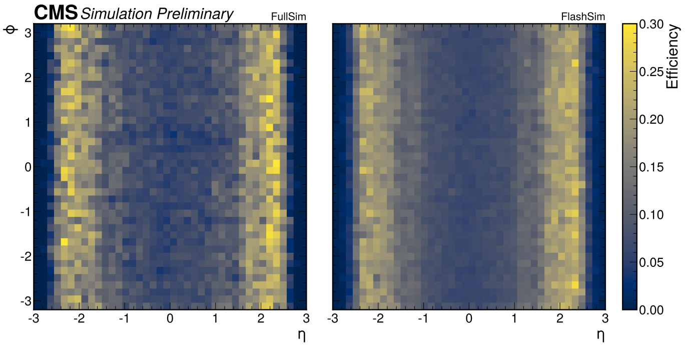

Figure 3: Example of efficiency map as function of η and ϕ for the full simulation target (left) and the learned FlashSim surrogate (right). From [3].

Physics object	Sources (one NN model for each source)	Number of simulated attributes per object
Jets	Generator Jet Fake from pile-up	39
Muons	Generator Muons Fake from Jets/PU Duplicates	53
Electrons	Generator Electrons Generator Photons (prompt) Jets	48
Photons	Generator Photons (prompt) Generator Electrons Jets	22
MET	GenMET and HT	25
FatJets	Generator AK8 Jets	53
SubJets	Generator AK8 SubJets	13
Tau	Reconstructed Jets with a GenTau RecoJets without a GenTau	27
Secondary Vertices	Jets with Heavy Flavour Light Jets Taus	16
Non MET scalars (e.g. PV)	Various event level inputs	16
FSR Photons	GenMuon/RecoMuon	6

Table 1: Objects implemented in the current version of FlashSim and their sources

bosons decays to b quarks, unknown resonances decaying to a Higgs boson and a 500 TeV mass resonance, high mass resonances in various beyond-the-SM models (including SUSY models).

A full list of validation plots obtained with this training can be found in Ref. [3].
As shown in Figure 4, FlashSim was able to replicate some detector features that are present in the simulation, but in some case missed some details in the tails of the distributions.

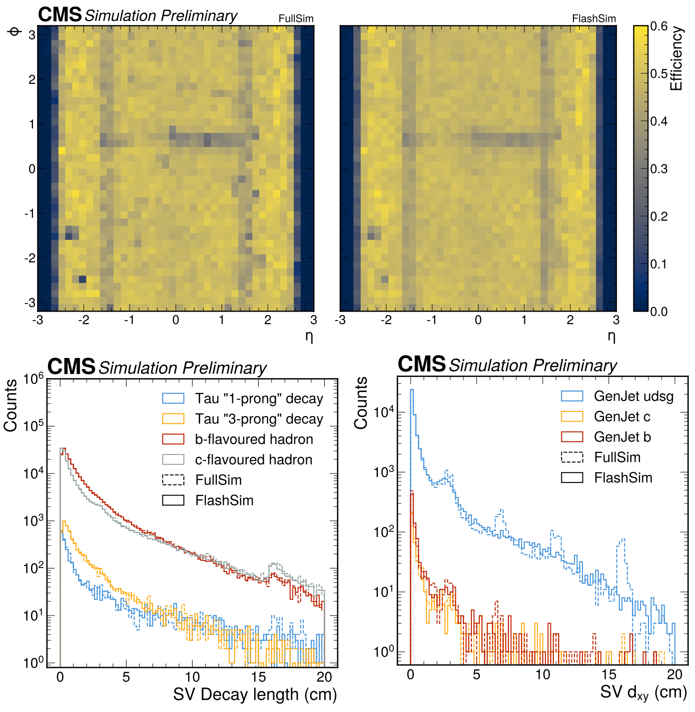

Figure 4: Example of results obtained after the training with 4M events. In the top row we can observe that detector features in the efficiency map present in full simulation (left) are reproduced by FlashSim (right). In the bottom figures the Secondary Vertex decay length in 3D for Tau signal (left) and in 2D for generic jet background (right) is shown: it can be noted that the spikes corresponding to pixel layers positions visible in full simulation are only partly reproduced by FlashSim [3], this can be improved with more training data and/or more complex models. From [3].

3.2 Accurate Conditioning and Correlations

A key strength of the model is its ability to produce different distributions based on the conditioning values as shown in Figure 5 (left).

Flow matching in addition is very accurate in capturing the multidimensional correlations both between the conditioning variables and the resulting properties and among the properties as shown in Figure 5 (right).

4. Complete Event Simulation and Toy Analyses

A complete simulation of a NANOAOD event involves several steps, which must be repeated for every object and source. The process includes extracting the conditioning information, running the efficiency model, running the properties model, and merging the output from various sources. To enable parallelization, RDataFrame and PyTorch were used in this simulation chain, leveraging all available GPUs. Using this chain, we simulated more than 100 millions events, also drawn from physical processes never seen during training.

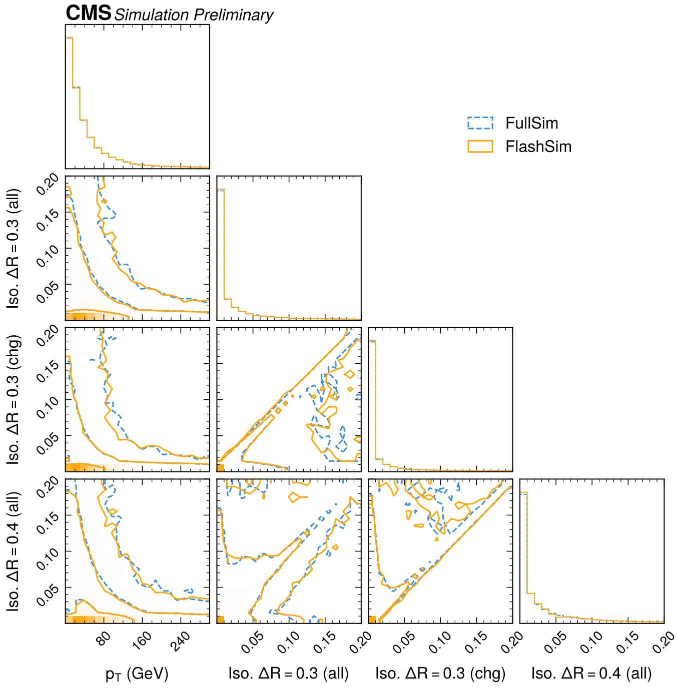

Figure 5: Example of the accuracy of the conditioning process (top) and correlations among variables (bottom). Figure [3].

4.1 Derived Quantities

The ability to simulate complete NANOAOD events allows us to compare derived quantities and perform typical analysis steps. Two toy analyses have been implemented: a VBF $H \rightarrow \mu\mu$ analysis very similar to the one used in $H \rightarrow \mu\mu$ CMS evidence paper [11] and $ZH \rightarrow llbb$. These analyses were tested from individual observables level down to the final DNN output, comparing FlashSim with Full Simulation on both physics processes used during training (but on original generator-level events) and on completely unseen processes. Results can be seen in Figure 6 where satisfying agreement is observed for all variables.

5. Speed and Bottlenecks

The current implementation of FlashSim achieves event simulation speeds ranging from 10 Hz to 1 kHz, using 20 properties models and 20 efficiency models starting from existing generated samples as summarized in Table 2. The performance varies depending on the hardware and on the accuracy of the ODE integration needed during inference of Flow Matching generative algorithms.

6. Conclusions

This work shows the successful implementation of an end-to-end simulation prototype for the CMS experiment, leveraging Machine Learning and targeting the NANOAOD data format. The results show a good balance between accuracy and speed, although further tuning might still be

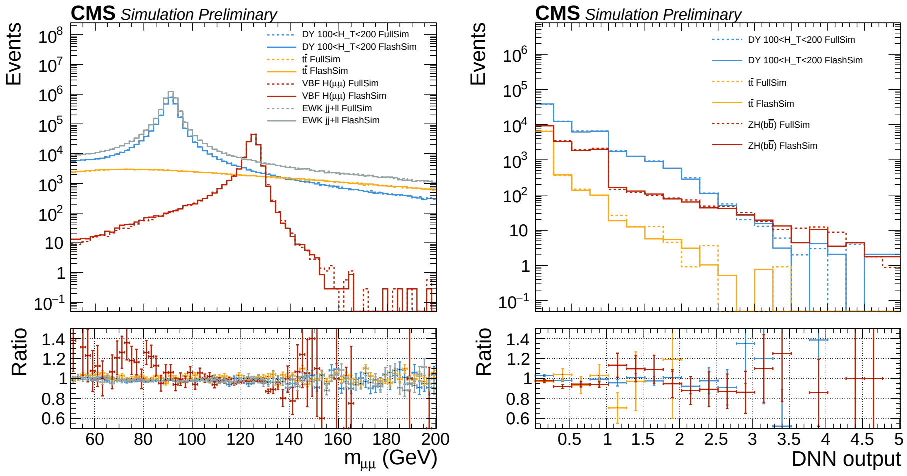

Figure 6: Results of the toy analysis: on the left the dimuon mass in the VBF $H \to \mu\mu$ is shown; on the right the DNN for the Hbb analysis is compared for multiple samples [3].

Table 2: Event simulation speed for different hardware

Processor	ODE accuracy (time steps)	Event simulation rate
GPU 3060	100	325 Hz
GPU 3060	20	690 Hz
CPU 1-core	100	15 Hz
CPU 1-core	20	60 Hz
CPU 4-core	20	120 Hz

needed. The performed toy analyses display good accuracy for derived quantities as well. To avoid the generator to be the bottleneck in the simulation, we introduced the oversampling technique. Further studies will be performed in order to make FlashSim production ready in the coming years.

References

[1] CMS Collaboration, The CMS Experiment at the CERN LHC, JINST 3 (2008) S08004, doi:10.1088/1748-0221/3/08/S08004.

[2] CMS Offline Software and Computing. CMS Phase-2 Computing Model: Update Document. Technical report, CERN, Geneva, 2022.

[3] CMS Collaboration, FlashSim (DP Note), CMS DP-2024/080

[4] F. Vaselli, A. Rizzi, F. Cattafesta, G. Cicconofri on bealf of the CMS Collaboration, FlashSim prototype: an end-to-end fast simulation using Normalizing Flow, CMS-NOTE-2023-003

[5] CMS Collaboration, Reweighting simulated events using machine-learning techniques in the CMS experiment, CMS-MLG-24-001, CERN-EP-2024-269, arXiv:2411.03023 (submitted to CSBS)

[6] G. Petrucciani, A. Rizzi, C. Vuosalo for the CMS Collaboration, Mini-AOD: A New Analysis Data Format for CMS, J.Phys.Conf.Ser. 664 (2015) no.7, 072052

[7] A. Rizzi, G. Petrucciani, M. Peruzzi on behalf of the CMS Collaboration, A further reduction in CMS event data for analysis: the NANOAOD format, EPJ Web Conf. 214 (2019) 06021

[8] F. Vaselli, F. Cattafesta, P. Asenov and A.Rizzi, End-to-end simulation of particle physics events with Flow Matching and generator Oversampling, Mach. Learn.: Sci. Technol. **5** 035007, arXiv:2402.13684

[9] Y. Lipman et al. , Flow Matching for Generative Modeling, arXiv:2210.02747

[10] Emiel Hoogeboom, How to build E(n) Equivariant Normalizing Flows, for points with features?

[11] The CMS collaboration., Sirunyan, A.M., Tumasyan, A. et al. Evidence for Higgs boson decay to a pair of muons. J. High Energ. Phys. **2021**, 148 (2021). https://doi.org/10.1007/JHEP01(2021)148