Denoising Diffusion to Accelerate Detector Simulation

Principal Investigator: Kevin Pedro (FNAL)
Co-investigators: Javier Duarte (UCSD), Daniel Elvira (FNAL), Lindsey Gray (FNAL), Philip Harris (MIT)

Introduction

The CMS detector simulation, which uses Geant4, consumed 40% of grid CPU at the beginning of Run 2 [1], even including various technical optimizations and physics-preserving approximations that improve its efficiency by a factor of 4–6 compared to the default [2]. In the HL-LHC era, the CPU time to simulate an event will increase by a factor of 3 or more [3, 4], because of the more complex detector geometry and the more detailed physics models needed to reproduce the precise measurements of the upgraded detector. The high level of physical accuracy in modern detector simulations is a crucial ingredient in high energy physics research, and its importance should not be understated. Statistical uncertainty arising from the limited sizes of simulated signal and background samples already poses a significant challenge for many analyses. CMS needs an accurate simulation with decreased CPU usage to match the growing LHC datasets, and there is a relatively small amount of serious R&D in this area compared to topics such as reconstruction. The PI previously led the CMS integration of the GeantV prototype [3], which concluded that the achievable speedup in classical, rule-based, CPU-bound simulation engines was too limited [5].

Artificial intelligence (AI) and machine learning (ML) are promising avenues to solve this problem. In particular, ML algorithm inference can be massively accelerated using coprocessors, such as GPUs—including those available from HPCs—and FPGAs. This functionality is already available in the CMS software via SONIC (Services for Optimized Network Inference on Coprocessors) [6–8], for which the PI is the lead developer. The PI has been L3 convener of the CMS ML4Sim (machine learning for simulation) group since 2020. The PI has also served as co-convener of the Detector Simulation working group in the HEP Software Foundation (HSF) and co-convener of the Theoretical Calculations and Simulation Topical Group in the Snowmass Computing Frontier. In these roles, he gathered input from and provided direction to the entire HEP community [9–12] and organizes the CMS efforts.

This proposal follows an earlier proposal, for which the PI was awarded support for 0.5 FTE-year of postdoc effort. There, we proposed to use ML denoising to enhance fast, low-quality simulation output, producing an improved final result. Strong results from a first prototype were obtained and will be summarized below. Oz Amram was hired to deliver the results outlined in the proposal milestones; he started in September 2022 with substantial ML expertise, including a leading entry [5] in the LHC Olympics [13]. The lessons learned from the prototype and the timing of Oz's arrival motivated a shift in the project's direction to use cutting-edge diffusion architectures from industry image generation. We have achieved impressive initial success with diffusion, and we will continue this approach in the second year, applying it to the CMS FastSim and High Granularity Calorimeter (HGCal).

Previous Work

Denoising, related to inpainting and super-resolution techniques, has been used in industry to reduce the need for expensive Monte Carlo ray-tracing in computer animation [14]. Generative adversarial networks (GANs) have been the most commonly explored option in ML for detector simulation, even deployed in production at ATLAS [15] and LHCb [16].

However, their reliability is ultimately limited by insurmountable mathematical difficulties: their training is not guaranteed to converge, they may suffer from mode collapse and vanishing gradients, and the validity of extrapolation beyond the training dataset is unclear. Instead, starting from the basic knowledge of particle showers encoded in fast simulations and enhancing the output with ML may provide an easier and more reliable solution. The initial proposal to use denoising for this purpose had four elements:

1. Determine parameter modifications to use Geant4 as a lower-quality fast simulation;
2. Generate particle shower training data;
3. Design and train the denoising algorithm;
4. Implement the final product in the CMS software using SONIC.

Given the long timescales (here, 16 months) involved in hiring a postdoc with the requisite expertise to deliver a novel AI application, the PI proceeded to work on the first three elements, largely in parallel, with available effort from undergraduate interns and master's students. Some preliminary results were already included in the first proposal: on element 1, modifying the Geant4 production cut was shown to reduce the computing time by a factor of two; and on element 3, a convolutional neural network (CNN) successfully denoised 2D shower images with noise artificially added. Since then, elements 1–3 were completed. Modifications to other Geant4 parameters (element 1), including Russian Roulette [17] energy thresholds for neutrons and photons and the energy threshold to use simplified chord-finding magnetic field propagation, were found to reduce the computing time by 4–7% each. At the same time, photon showers in the CMS ECAL were generated with a modified production cut as training data (element 2). Finally, the denoising algorithm was trained and optimized, including data preprocessing and architecture modifications, and the results for various shower-related variables were assessed using ensemble comparisons and an event-by-event concordance correlation metric. Figure 1 shows selected results; in particular, the observed agreement in per-pixel energy down to 0.1 MeV is difficult for even very complicated GAN architectures to achieve without extensive postprocessing. Though integration into CMSSW via SONIC has not yet been achieved, the inference speed of the CNN was measured at 38 shower images per second on CPU and up to 8200 on an Nvidia P100 GPU, orders of magnitude faster than Geant4. These results were published [18].

The premise of element 1, using a modified Geant4 as a fast simulation, was based on the hypothesis that the additional information associated with each energy deposit by the full simulation engine could be used as extra input features to improve the output of the denoising CNN. Amram's first task was to test this hypothesis; he examined features including the time associated with each hit, the step length, and momentum differences, both as individual features and combined in a custom deep set architecture to derive a new optimized feature as an additional channel for each pixel input in the shower image. The deep set approach produced minor gains in the concordance correlation scores, but otherwise the additional information did not prove particularly useful. In addition, parameter modifications to Geant4 could not produce a fast enough simulation to be viable for HL-LHC, without in-depth changes such as rewriting physics lists. This early feedback was important to be able to modify the approach while still utilizing the principle of denoising.

In August 2022, a company called Stability AI released their image generation algorithm, Stable Diffusion, to the public, following on the heels of other new image generators such as DALL·E 2, Imagen, and Midjourney. This new breed of generative ML produced results far more detailed than previous algorithms. Notably, they all used variations of an architecture

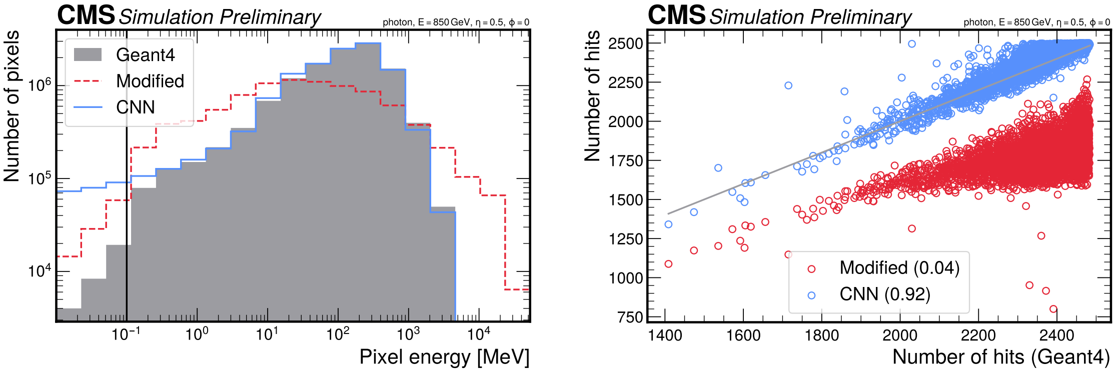

Figure 1: Left: comparison between the modified simulation, denoising CNN output, and GEANT4 for distributions of the per-pixel energy. Right: the per-image comparison of the number of hits above threshold with the concordance correlation between GEANT4 and the others listed in parentheses.

called diffusion models [19], which had only just started to be explored in particle physics [20]. Diffusion is a sophisticated denoising process: small amounts of noise are repeatedly added to an image, and a network is trained to remove this added noise. Once the training is completed, new images can be generated by iteratively subtracting small amounts of noise from random input until an image is generated. Learning this iterative process is more effective than learning a one-shot transformation all the way to a fully denoised image. Around the same time, the CaloChallenge [21] was announced, providing the first public, standardized datasets for ML-based simulation. Starting with these relatively simple datasets facilitated rapid iteration and improvements to the ML algorithm, described below, as well as objective comparisons to other approaches using various metrics [22].

Amram had an operational diffusion network within 2 months, dubbed “CaloDiffusion”, and competitive results only 1 month after that. The baseline model itself represents an improvement over the previous denoising prototype, using a more sophisticated U-net architecture with 3D convolutions. Improvements to the baseline architecture include preprocessing, cylindrical convolutions that properly handle calorimeter geometries, conditioning the convolutions on positional information, and optimizing the noise- and sampling-related parameters. We also found denoising diffusion, which directly predicts the added noise in the image, simpler and better performing than the score-based variant used in Ref. [20]. In order to handle CaloChallenge Dataset 1, which mimics the ATLAS calorimeter’s irregular geometry, we developed a learned, invertible embedding to convert it into a regular geometry on which convolutions can be performed. Figure 2 shows the latest results, which have been submitted to the CaloChallenge and presented at CHEP [23], and will be published soon. These generated showers achieve excellent scores of 0.6–0.7 on the CaloChallenge metric, the area under the curve for a classifier trained to distinguish between real and generated showers (where 0.5 indicates complete indistinguishability), and perform similarly well in other metrics such as the Fréchet particle distance [22].

Proposal

There are several immediate improvements that will be made to the existing CaloDiffusion algorithm. The agreement in global quantities, such as total energy per shower, can be

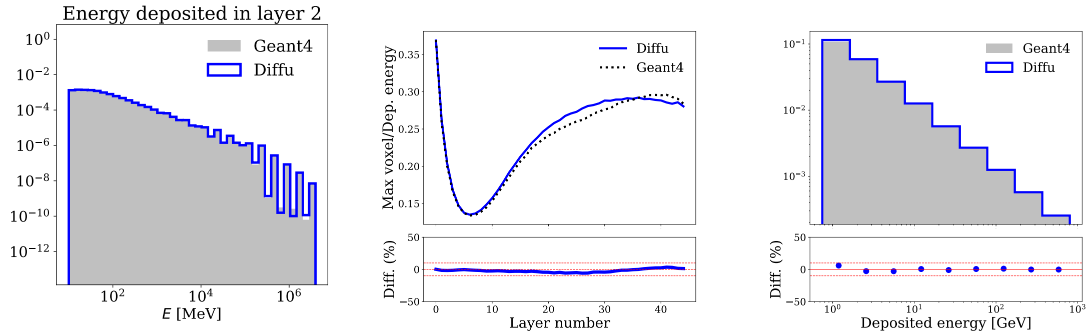

Figure 2: Comparisons of various physical quantities for CaloChallenge datasets 1 (left), 2 (middle), and 3 (right).

improved by adding an ensemble term to the loss function. CaloDiffusion inference, though still faster than GEANT4, is slower by default than other generative ML algorithms because many iterations of the denoising process are needed to produce high quality output. We will explore recent techniques such as consistency models [24] to distill the denoising operation of the trained network, reducing the number of iterations needed while preserving quality as measured by the metrics described above. In addition, the computational burden of each denoising step can be reduced by denoising in a lower-dimensional latent space [25], an approach we have already started to pursue. Latent diffusion may further help apply this method to more complicated detector geometries.

So far, CaloDiffusion has been used in a fully generative mode, creating particle shower output from random noise. However, as discussed in the previous proposal, denoising a low quality shower from a fast simulation should be more reliable, by incorporating basic physical knowledge of shower shapes. Given the conclusion of element 1 of the previous proposal, that using GEANT4 as a fast simulation is not achievable, we will instead use the existing CMS FastSim application [26]. We have already obtained initial results from a technique called Cold Diffusion [27] to learn transitions between any two types of images, using "average" showers computed from the CaloChallenge datasets as a stand-in for actual FastSim. Some deficits in this technique compared to standard diffusion may be overcome using mixture density networks [28], which will be a priority item. This hybrid approach using FastSim is synergistic with another project by the PI to refine high-level FastSim outputs using ML, replacing coarse, manual correction factors [29]. By iteratively refining FastSim at different processing stages (after the simulation step and at the end-stage analysis step), each refinement task becomes progressively easier to learn, leading to even higher quality. Further, CaloDiffusion should need fewer denoising steps for FastSim, speeding up inference.

The hybrid approach will be first explored with the existing CMS calorimeter systems and applied to photons, electrons, and hadrons. However, the real motivation for ML-based fast simulation in CMS is the HGCal, which is the primary driver of increased time to simulate HL-LHC events. The HGCal has an extremely complicated geometry with hexagonal cells of varying materials and sizes, which frustrates standard convolutional operators, such as those used in CaloDiffusion. The learned mapping used for CaloChallenge Dataset 1 may be extendable to this problem. Alternatively, graph convolutions may be used, with the layer-by-layer conditioning approach from Ref. [30] to reduce memory usage.

We anticipate integration of at least one approach, fully generative or hybrid, into CMSSW, with SONIC employed to enable accelerated GPU inference anywhere, by the end of the award period. This implies the availability of ML-based simulation to the CMS collaboration by the end of Run 3, ensuring sufficient statistical precision to carry out any legacy analysis using the entire LHC dataset and advance preparation for the Run 4 detector upgrades. Amram is an ideal candidate to carry out this integration because of his background as Pixel Offline Software Convener. Coupled with the PI's experience as HCAL CMSSW Convener, Upgrade Software Coordinator, and Deputy Release Manager, this team will bring high-quality ML-based simulation to large-scale production and official usage. The CMS-specific algorithm development and results will be published via the CMS ML Group (MLG).

Timeline & Milestones

Because Amram started in September 2022 while this renewal is submitted in May 2023, less than a full year later, milestones for 15 subsequent months (5 quarters) are listed in Table 1 below. These deliverables build on the previous work discussed above.

Month	Milestone	Deliverables
3	First results	• Conference presentations of results • Publication of CaloDiffusion algorithm • Overall CaloChallenge publication
6	CaloDiffusion improvements	• Global performance • Faster inference (distillation, latent diffu.)
9	Hybrid approach	• Cold Diffusion w/ Mixture Density Nets • Adapt to CMS calorimeters, particle types
12	HGCal	• Convolutions in complicated geometry • Validate generative and hybrid results
15	Implementation	• CMSSW integration w/ SONIC • Publication of CMS results via MLG

Table 1: Timeline of milestones and deliverables for the proposal.

Mentoring & Supervision

Many detector simulation and AI experts are resident at Fermilab or nearby institutions (Argonne, University of Chicago, etc.) to provide feedback and guidance. Fermilab hosts the LHC Physics Center, through which we can collaborate with university students and postdocs. Amram has already demonstrated his leadership capabilities by supervising a Fermilab AI Associate, Ashia Lewis, who contributed to ML algorithm exploration for CaloDiffusion. The Fermilab CMS group has established a highly successful postdoc mentoring program that includes annual reports and regular communication with both a direct supervisor and a mentor who looks out for the postdoc's broader career interests. The CMS ML4Sim meeting, run by the PI, provides opportunities to get feedback from the collaboration. Amram has given and will continue to give conference presentations, as well as preparing publications.

References

[1] HEP Software Foundation, "HEP Software Foundation Community White Paper Working Group - Detector Simulation", `arXiv:1803.04165`.

[2] K. Pedro, "Current and future performance of the CMS simulation", *Eur. Phys. J Web Conf.* **214** (2019) 02036, `doi:10.1051/epjconf/201921402036`.

[3] K. Pedro, "Integration and Performance of New Technologies in the CMS Simulation", *Eur. Phys. J. Web Conf.* **245** (2020) 02020, `doi:10.1051/epjconf/202024502020`, `arXiv:2004.02327`.

[4] J. Apostolakis et al., "Detector Simulation Challenges for Future Accelerator Experiments", *Front. in Phys.* **10** (2022) 913510, `doi:10.3389/fphy.2022.913510`.

[5] G. Amadio et al., "GeantV: Results from the Prototype of Concurrent Vector Particle Transport Simulation in HEP", *Comput. Softw. Big Sci.* **5** (2021) 3, `doi:10.1007/s41781-020-00048-6`, `arXiv:2005.00949`.

[6] J. Duarte et al., "FPGA-accelerated machine learning inference as a service for particle physics computing", *Comp. Soft. Big Sci.* **3** (2019) 13, `doi:10.1007/s41781-019-0027-2`, `arXiv:1904.08986`.

[7] J. Krupa et al., "GPU coprocessors as a service for deep learning inference in high energy physics", *Mach. Learn. Sci. Tech.* **2** (2021) 035005, `doi:10.1088/2632-2153/abec21`, `arXiv:2007.10359`.

[8] D. S. Rankin et al., "FPGAs-as-a-Service Toolkit (FaaST)", in *Proceedings of Sixth International Workshop on Heterogeneous High-performance Reconfigurable Computing (H2RC20).* 2020. `arXiv:2010.08556`. doi:10.1109/H2RC51942.2020.00010.

[9] A. Adelmann et al., "New directions for surrogate models and differentiable programming for High Energy Physics detector simulation", in *Snowmass 2021.* 3, 2022. `arXiv:2203.08806`.

[10] S. Banerjee et al., "Detector and Beamline Simulation for Next-Generation High Energy Physics Experiments", `arXiv:2203.07614`.

[11] P. Boyle, K. Pedro, and J. Qiang, "CompF2: Theoretical Calculations and Simulation Topical Group Report", `arXiv:2209.08177`.

[12] V. D. Elvira et al., "The Future of High Energy Physics Software and Computing", in *Snowmass 2021.* 10, 2022. `arXiv:2210.05822`.

[13] G. Kasieczka et al., "The LHC Olympics 2020 a community challenge for anomaly detection in high energy physics", *Rept. Prog. Phys.* **84** (2021) 124201, `doi:10.1088/1361-6633/ac36b9`, `arXiv:2101.08320`.

[14] S. Bako et al., "Kernel-predicting convolutional networks for denoising Monte Carlo renderings", *ACM Trans. Graph.* **36** (2017) `doi:10.1145/3072959.3073708`.

[15] ATLAS Collaboration, “AtlFast3: the next generation of fast simulation in ATLAS”, *Comput. Softw. Big Sci.* **6** (2022) 7, doi:10.1007/s41781-021-00079-7, arXiv:2109.02551.

[16] M. Barbetti, “Lamarr: LHCb ultra-fast simulation based on machine learning models deployed within Gauss”, in *21th International Workshop on Advanced Computing and Analysis Techniques in Physics Research: AI meets Reality.* 3, 2023. arXiv:2303.11428.

[17] CMS Collaboration, “Upgrades for the CMS simulation”, *J. Phys. Conf. Ser.* **608** (2015) 012056, doi:10.1088/1742-6596/608/1/012056.

[18] CMS Collaboration, “Denoising Convolutional Networks to Accelerate Detector Simulation”, *J. Phys. Conf. Ser.* **2438** (2023) 012079, doi:10.1088/1742-6596/2438/1/012079, arXiv:2202.05320.

[19] J. Ho, A. Jain, and P. Abbeel, “Denoising diffusion probabilistic models”, arXiv:2006.11239.

[20] V. Mikuni and B. Nachman, “Score-based generative models for calorimeter shower simulation”, *Phys. Rev. D* **106** (2022) 092009, doi:10.1103/PhysRevD.106.092009, arXiv:2206.11898.

[21] M. F. Giannelli et al., “Fast Calorimeter Simulation Challenge”, 2022. https://calochallenge.github.io/homepage/.

[22] R. Kansal et al., “Evaluating generative models in high energy physics”, *Phys. Rev. D* **107** (2023) 076017, doi:10.1103/PhysRevD.107.076017, arXiv:2211.10295.

[23] O. Amram and K. Pedro, “Fast and Accurate Calorimeter Simulation with Diffusion Models”, 2023. https://indico.jlab.org/event/459/contributions/11736/.

[24] Y. Song, P. Dhariwal, M. Chen, and I. Sutskever, “Consistency models”, arXiv:2303.01469.

[25] R. Rombach et al., “High-resolution image synthesis with latent diffusion models”, arXiv:2112.10752.

[26] S. Sekmen, “Recent Developments in CMS Fast Simulation”, *PoS* **ICHEP2016** (2016) 181, doi:10.22323/1.282.0181, arXiv:1701.03850.

[27] A. Bansal et al., “Cold diffusion: Inverting arbitrary image transforms without noise”, arXiv:2208.09392.

[28] C. M. Bishop, “Mixture density networks”, 1994. https://research.aston.ac.uk/en/publications/mixture-density-networks.

[29] S. Bein et al., “Refining fast simulation using machine learning”, 2023. https://indico.jlab.org/event/459/contributions/11725/.

[30] S. Diefenbacher et al., “L2LFlows: Generating High-Fidelity 3D Calorimeter Images”, arXiv:2302.11594.