DeepMuon: Generating Cosmic-Ray Muons Based on Optimal Transport

Authors: Wang, Mr. Ao-Bo, Su, Mr. Bohua, Pan, Mr. Chu-Cheng, Dong, Dr. Xiang, Sun, Mr. Yu-Chang, Hu, Mr. Yu-Xuan, Cheng, Mr. Ao-Yan, Cai, Prof. Hao, Fan, Prof. Xi-Long, Cai, Prof. Hao

Date: 2026-02-15T12:32:58+00:00

Abstract

Deep learning can learn distributions directly from data, enabling fast event simulation without hand-crafted designs or costly sampling. Yet many mainstream generators, such as diffusion models, require sophisticated architectures and multi-step denoising, which reduces throughput. Neural generators also struggle with sharp, high-kurtosis distributions common in physics, such as the cosmic-ray muon energy spectrum. Cosmic-ray muon simulation underpins non-destructive imaging, positioning and navigation, timing, and cryptography, and is traditionally performed with Monte Carlo or parametric models. Here we propose DeepMuon, a deep learning-based cosmic-ray muon generator that targets both efficiency and accuracy while addressing high-kurtosis fitting. We first apply an inverse Box–Cox transformation to reduce energy-spectrum kurtosis and simplify the learning task. We then optimize the generator with the Sliced Wasserstein Distance loss, achieving high-fidelity, one-step cosmic-ray muon generation with only a 2-layer Transformer encoder. DeepMuon also learns muon distribution patterns from limited data, enabling simulation of real detector-measured distributions. At sea level, DeepMuon substantially accelerates muon generation relative to CRY. We further develop a DeepMuon-based pipeline to directly simulate underwater muon distributions and compare against CRY, demonstrating suitability for simulation and imaging tasks. For more details about our open-source project, please visit:https://github.com/wangab0/deepmuon.

Full Text

Preamble

DeepMuon: Generating Cosmic-Ray Muons Based on Optimal Transport∗ Ao-Bo Wang,1, † Bohua Su,1, † Chu-Cheng Pan,1 Xiang Dong,1 Yu-Chang Sun,1

Yu-Xuan Hu,1 Ao-Yan Cheng,1 Hao Cai,1, ‡ and Xi-Long Fan1, § 1School of Physics and Technology, Wuhan University, Wuhan 430072, People' s Republic of China Deep learning can learn distributions directly from data, enabling fast event simulation without hand-crafted designs or costly sampling. Yet many mainstream generators, such as diffusion models, require sophisticated architectures and multi-step denoising, which reduces throughput. Neural generators also struggle with sharp, high-kurtosis distributions common in physics, such as the cosmic-ray muon energy spectrum. Cosmic-ray muon simulation underpins non-destructive imaging, positioning and navigation, timing, and cryptography, and is traditionally performed with Monte Carlo or parametric models. Here we propose DeepMuon, a deep learning-based cosmic-ray muon generator that targets both efficiency and accuracy while addressing high- kurtosis fitting. We first apply an inverse Box–Cox transformation to reduce energy-spectrum kurtosis and simplify the learning task. We then optimize the generator with the Sliced Wasserstein Distance loss, achiev- ing high-fidelity, one-step cosmic-ray muon generation with only a 2-layer Transformer encoder. DeepMuon also learns muon distribution patterns from limited data, enabling simulation of real detector-measured distri- butions. At sea level, DeepMuon substantially accelerates muon generation relative to CRY. We further develop a DeepMuon-based pipeline to directly simulate underwater muon distributions and compare against CRY, demonstrating suitability for simulation and imaging tasks. For more details about our open-source project, please visit:https://github.com/wangab0/deepmuon.

Keywords: cosmic muon generator, muon tomography, muon radiography, deep learning

INTRODUCTION

Deep learning models have been widely applied in high- energy physics event generation[10, 11, 17]. Previous deep learning-based generative algorithms primarily utilized mod- els such as Variational Autoencoders (VAEs)[13], Generative Adversarial Networks (GANs)[12], or Diffusion Models[23].

GANs consist of a generator and a discriminator engaging in a competitive adversarial process: the generator produces sim- ulated data while the discriminator attempts to distinguish it from real data. VAEs, on the other hand, employ an encoder to map input data into a probabilistic latent space and a de- coder to reconstruct the data, optimizing the evidence lower bound. In contrast, Diffusion Models operate by gradually adding noise to the data and then training a neural network to reverse this process, iteratively denoising random noise to reconstruct the target distribution.

However, these approaches face significant challenges in practice. GANs are notoriously difficult to train due to con- vergence instability and mode collapse[21], often failing to accurately capture the full phase space of complex physi- cal events. While VAEs offer more stable training, they frequently suffer from posterior collapse and tend to gen- erate blurred samples that lack the high-frequency details required for precision physics[22]. Furthermore, Diffusion Models require a large number of iterative steps to denoise ∗ The National Science Foundation of China (nos.11735010, U1932108, U2032102 and 12061131006) provided the funding for this project. Their financial support was instrumental in the realization of our research goals.

The numerical calculations in this paper have been done on the supercom- puting system in the Supercomputing Center of Wuhan University. † These authors contributed equally to this work. ‡ hcai@whu.edu.cn § xilong.fan@whu.edu.cn the data, resulting in extremely slow generation speeds that hinder rapid simulation. Consequently, these frameworks typically rely on complex, parameter-heavy architectures to achieve high fidelity, making them computationally expensive for high-dimensional distributions. In contrast to the afore- mentioned architectures, generative models grounded in Op- timal Transport (OT) theory, specifically those utilizing the Sliced Wasserstein Distance (SWD) as a loss function, have demonstrated exceptional precision in learning and generat- ing high-dimensional physical event distributions[24]. Cru- cially, SWD serves a dual purpose: it functions not only as a robust optimization objective but also as a direct met- ric for quantifying distributional discrepancies[9]. This du- ality ensures that the convergence of the model is intrinsi- cally linked to the statistical consistency of the generated data, thereby endowing the training outcome with explicit physical significance.The theoretical superiority of this ap- proach stems from the Optimal Transport framework, which quantifies the distance between probability distributions as the minimal cost required to transport mass from one distri- bution to the other. Unlike the Kullback-Leibler divergence employed in mainstream likelihood-based generative mod- els, the Wasserstein distance derived from OT is symmetric and satisfies the triangle inequality. More importantly, it pro- vides a meaningful distance metric and non-vanishing gradi- ents even when the supports of the generated and real dis- tributions are disjoint (non-overlapping). These mathemati- cal properties constitute the rigorous theoretical basis for uti- lizing SWD to measure distribution differences in physical event generation.Computationally, SWD approximates the in- tractable high-dimensional Wasserstein distance by leverag- ing the Radon Transform.

It projects the two probability distributions onto multiple one-dimensional subspaces and aggregates the Wasserstein distances computed within these projections. Compared to direct comparison in the high- dimensional original space, this method significantly reduces computational complexity while preserving the ability to cap- ture precise distributional discrepancies [14].Formally, for two probability measures α and β, SWD is defined as in Eq. 1:

SW D(α, β) = (cid:90) S d−1 OTq (R$\theta(\alpha)$, R$\theta(\beta)$) dθ, where Rθ denotes the projection along direction θ and OTq is the one-dimensional Wasserstein distance (efficiently com- puted via sorting).

However, deep learning models struggle to directly learn sharp distributions with high kurtosis. This is because gen- erative neural networks were originally designed for image generation tasks, where the output is typically the pixel inten- sity values of an image, which range from 0 to 255. In gen- eral, the distribution of pixel values is fairly uniform across different intensity ranges. In high-energy particle generation, the goal is to generate high-dimensional distributions, includ- ing energy and angular distributions. The energy distribution, in particular, may exhibit high kurtosis, as seen in the cosmic ray muon energy spectrum. The energy range of cosmic ray muons spans widely: below 1 GeV, the spectrum is nearly flat; between 1 GeV and 10 GeV, it begins to decay; and be- tween 10 GeV and 100 GeV, it decays rapidly, resulting in a highly kurtotic energy spectrum. Simply scaling the energy spectrum to the 0-1 range for model output would further ex- acerbate the kurtosis, and the dense distribution would make it difficult for the deep learning model to extract meaningful features. Therefore, an appropriate kurtosis transformation is required to reduce the kurtosis of the energy spectrum and allow the SWD model to better fit the high-kurtosis data.

The Box-Cox transformation[8] is a commonly used tech- nique to adjust the distribution of data to more closely re- semble a normal distribution. Through experimentation, we found that the inverse Box-Cox transformation effectively re- duces the kurtosis of the cosmic ray muon energy distribu- tion, transforming it into a relatively smoother distribution that is more suitable for SWD model learning. Based on this, we developed a machine learning method called DeepMuon, which combines SWD with the Box-Cox transforma- tion. This method successfully addresses the challenge of generating high-kurtosis distributions using SWD and pro- duces cosmic ray muon distributions across various scenarios, demonstrating significant practical value.

Primary cosmic rays are primarily composed of atomic nu- clei, with 86% consisting of hydrogen nuclei (protons), 12% of helium nuclei, and 1% of carbon, nitrogen, oxygen, and iron nuclei, along with 1% electrons. Additionally, there are trace amounts of high-energy gamma rays, neutrinos, and antiprotons. When these primary cosmic rays interact with atoms in the Earth' s atmosphere, they produce a large num- ber of secondary particles. At sea level, the cosmic rays are composed of these secondary particles, and among the sec- ondary cosmic rays that reach the Earth' s surface, 80% of the charged particles are muons. These muons hit the ground at a frequency of approximately 170 Hz per square meter[1]. As detector technologies have advanced, cosmic-ray muons have found a wide range of applications. For example, Procureur et al. utilized cosmic-ray muon imaging technology to discover a previously unknown, 9-meter-long corridor-like structure, termed the "North Face Corridor" (NFC), located behind the Chevron structure on the North Face of Khufu' s Pyramid[2].

This method, known as muon radiography or muon absorp- tion radiography, infers the material composition of an object by measuring the muon flux passing through it. It is com- monly used for imaging large structures. Another technique, muon tomography, was developed by a team at Los Alamos in 2003[3]. This method uses the scattering angle of muons to create a three-dimensional image of an object by comparing the incoming and outgoing muon beams. Both muon radio- graphy and muon tomography rely heavily on simulations of cosmic ray muon trajectories as they pass through different materials[19]. Currently, there are two main types of simula- tion tools used for cosmic ray muon generation:

1. Monte Carlo-based cosmic ray muon generators such

as CORSIKA[5] and CRY[4]. These Monte Carlo models simulate the physical processes involved in cos- mic ray muon production, including the generation of primary cosmic rays, the decay of secondary cosmic rays, and their interactions with the atmosphere. Since the Monte Carlo method simulates the entire physical process, it is computationally expensive. However, it allows for flexible modifications of physical models and other parameters, and the simulation results also include other cosmic ray fluxes besides muons.

2. Parametric model-based cosmic ray muon generators

such as CMSCGEN[6] and EcoMug[7]. A paramet- ric model generation method is based on experimental data, from which semi-empirical formulas are derived to describe the muon flux under specific conditions.

These formulas can be used directly or fit to Monte Carlo data through polynomial fitting. For example, CMSCGEN is a parametric model obtained by polyno- mial fitting to data generated by CORSIKA.

Unlike traditional methods, DeepMuon adopts an innova- tive data-driven approach to generate cosmic-ray muon dis- tributions. This deep learning model is capable of learning the characteristics of muon distributions directly from vari- ous data sources, including both simulation data and real de- tector data, allowing it to generate muon data consistent with target distributions with exceptional efficiency. While Monte Carlo methods and parametric modeling approaches require constructing specific physical models to generate muon distri- butions, deep learning methods bypass the need for an explicit physical model, capturing and generating realistic cosmic-ray muon distributions directly through data learning. This data- driven approach holds particular practical value for applica- tions involving real detector data.

In our study, we trained the DeepMuon model using muon data at sea level generated by the Cosmic-ray Shower Library (CRY) simulation. To evaluate the model' s ability to generate cosmic-ray muon distributions under varying conditions and to further explore the potential of deep learning in cosmic-ray muon imaging, we constructed a pipeline using DeepMuon as the muon source. GEANT4[18], a widely-used particle physics simulation framework, specializes in simulating the propagation, interaction, and decay of particles within matter, accurately modeling the interactions of particles, including cosmic-ray muons, with various materials. In our pipeline, we employed GEANT4 to simulate the distribution of muons passing through different water depths and compared the sim- ulation results with those directly obtained using CRY and GEANT4.

Muon distributions under different water depths exhibit distinct kurtosis features, resulting from varying attenuation and scattering processes during muon propagation in water.

As water depth increases, the muon penetration rate gradu- ally declines, causing observable changes in the kurtosis of the muon distribution. In our simulations, DeepMuon suc- cessfully learned these variations, generating muon distribution data consistent with the conditions at different water depths. By simulating underwater muon distributions across various depths, our experiments demonstrated that the Deep- Muon model effectively captures and reproduces these kurto- sis changes, affirming its effectiveness in modeling complex environments.

Compared to brute-force simulation methods based on GEANT4, our pipeline significantly enhances simulation speed. In particular, when compared with traditional Monte Carlo cosmic-ray generators like CRY, our pipeline' s results closely align with CRY's simulation outcomes, indicating that DeepMuon can accurately model cosmic-ray muon distribu- tions with higher simulation efficiency. This efficiency ad- vantage provides potential application value in scenarios requiring extensive muon data.

The following sections will provide a detailed descrip- tion of the model architecture and the preprocessing meth- ods applied to muon data. We will also explain how the model achieves acceleration of GEANT4-based simulations and demonstrate the model' s performance and fit to muon dis- tribution results.

II. METHOD A. Data Preprocessing To train our cosmic muon generation model, we utilized cosmic muon distribution data generated by CRY to construct our training dataset. Given that the spatial distribution of cosmic muons at sea level is approximately uniform, we focused on using only their energy and angular distributions for model training.

The energy distribution of cosmic muons is highly peaked, as shown in Figure 1 [Figure 1: see original paper], which illustrates the energy spectrum generated by CRY. Such dense distributions exhibit minimal variation in feature space, making it challenging for the model to learn them directly. Additionally, the figure reveals a step- like feature in the energy spectrum produced by CRY. This artifact is likely caused by the discretization introduced by CRY' s use of precomputed input tables derived from MC- NPX, from which cosmic ray samples are drawn. Similar characteristics have been observed in other studies that use CRY for muon generation[16]. To address this challenge, we Fig. 1. The distribution of cosmic ray muons at sea level generated by CRY is highly skewed (left figure). After applying the inverse Box-Cox transformation, the distribution exhibits improved statisti- cal characteristics (right figure). introduced the Box-Cox transformation, a commonly used statistical technique aimed at adjusting the shape of a data distribution. The Box-Cox transformation helps improve the statistical properties of the data, making it closer to a normal distribution. In practice, this transformation is often applied to data with high kurtosis and skewness, which can otherwise lead to misleading results during analysis. The primary goal of the Box-Cox transformation is to modify the shape of the data distribution, making it more suitable for subsequent sta- tistical analysis and modeling. The Box-Cox transformation is defined as follows: y(λ) = (cid:40) xλ−1 log x, if $\lambda = 0$ if $\lambda = 0$ Here, x represents the input data, and λ is an adjustable pa- In our approach, we apply the inverse Box-Cox rameter. trans- formation to the energy data. The inverse transformation is defined as follows: (cid:40) (λy(λ) + 1)1/λ, ey(λ), if $\lambda = 0$ if $\lambda = 0$ Through experimentation, we found that setting the Box- Cox transformation parameter λ to 9 yields the most effec- tive transformation results. After applying this transforma- tion, the energy data requires further processing to meet the model' s input require- ments. First, we subtract the mean from the energy values. This step reduces the magnitude of the en- ergy and helps the transformed distribution become closer to a Gaussian-like shape, which makes it easier for the model to learn the underlying features of the data. Additionally, a Gaussian-like distribution helps mitigate potential training is- sues such as vanishing or exploding gradients.

After mean subtraction, we apply the tanh function to map the energy values into the range (-1,1). This step compresses the input data into a finite range, contributing to the model' s stable training. Once the energy data is mapped to the (-1,1) interval, we combine the processed energy values with the di- rectional angles to form a feature vector. This feature vector is then fed into the neural network model for further process- drawn, and the SWD Loss is computed to update θ via back- propagation, driving the generated distribution to converge to- wards the physical reality.

B. Formal Formulation and Model Architecture

1. Problem Formulation

Formally, we define the cosmic ray muon generation task as learning a parameterized mapping Gθ : Z → X , where θ represents the learnable parameters of the neural network.

The objective is to approximate the true high-dimensional dis- tribution of cosmic ray muons, denoted as pdata(x), by min- imizing the discrepancy between the generated distribution pθ(x) and pdata(x).

Let z be a latent variable sampled from a fixed prior dis- tribution pz(z), which we choose as a standard normal distri- bution N (0, I). The generator Gθ transforms this noise prior into the target physical space: xgen = Gθ(z), z N (0, I) The optimal parameters θ∗ are obtained by minimizing the SWD between the generated distribution and the ground truth distribution obtained from CRY simulations: θ∗ = arg min LSW D(pdata, Gθ#pz) where Gθ#pz denotes the push-forward measure of the prior through the generator.

2. Architecture Details

Figure 2 [Figure 2: see original paper] illustrates the overall structure of our DeepMuon model. The architecture is designed to be lightweight yet effective, consisting of three primary components: a linear embedding layer, a Transformer encoder, and an Multi-Layer Perceptron (MLP) projector.

The input to the model is a batch of noise vectors sampled from the standard normal distribution, with a shape of (C × 100), where C represents the batch size. This latent noise is first projected by the linear embedding layer into a high- dimensional feature space of size (C × 3 × 1024), serving as the input for the subsequent attention mechanism.

The core of the network is a two-layer, eight-head Trans- former encoder. It captures the complex dependencies within the feature space without the need for recurrent connections.

The encoder' s output is then passed to the MLP projector, which projects the transformed latent distribution to the tar- get physical observables. The final output has dimensions (C × 1024 × 3), where each of the 1024 elements represents a generated cosmic ray muon instance. The last dimension corresponds to the three key physical attributes of each muon: energy (E), zenith angle, and azimuth angle (ϕ).

To ensure physical fidelity, we utilized a dataset of 100 mil- lion sea-level cosmic ray muon instances generated by CRY for training. In each training iteration, a batch of samples is C. Model Performance Figures 3–4 present the performance of our sea-level muon generation model. We compared the energy distributions of muons generated by our model with those produced by CRY, and also examined muon fluxes at different zenith angles (0° and 60°). The distribution plots show a high degree of consis- tency between the data generated by our model and the CRY- generated target data.

To further validate our model' s ability to accurately fit smooth, physically faithful distributions, we also generated 3.5 × 105 cosmic-ray muons with CORSIKA and trained a separate DeepMuon model on this dataset. Unlike CRY, CORSIKA produces a continuous energy spectrum free of discretization artifacts, making it an ideal benchmark for eval- uating the intrinsic fitting precision of the SWD-based ap- proach. As shown in Fig. 5 [Figure 5: see original paper], the agreement between Deep- Muon and the CORSIKA reference is excellent: the over- all energy distribution yields \$ \$2/ndf = 134.6/148 with a Kolmogorov-Smirnov p-value of 0.914, while the energy dis- tributions at zenith angles of 15° and 45° give \$ \$2/ndf = 83.6/78 (p = 0.311) and \$ \$2/ndf = 90.7/78 (p = 0.154), respectively. These results demonstrate that the SWD loss enables DeepMuon to achieve high-precision modeling of cosmic-ray muon distributions. However, as detailed in Sec. II D, CORSIKA' s generation speed is orders of magnitude slower than CRY, making it impractical to produce the large-scale training datasets required by our pipeline, particu- larly for obtaining underwater muon distributions, which de- mand the generation of a vast number of incident muons. We therefore adopt CRY-generated data as the primary training data source throughout this work.

To prevent overfitting to the non-physical step-like ar- left), tifacts inherent in the CRY simulation (see Fig. 6 [Figure 6: see original paper], we implemented a spectral monitoring criterion. Theoret- ically, approximating these discontinuities with continuous functions induces high-frequency oscillations (Gibbs phenomenon). Consequently, we tracked the power spectral den- sity above a cutoff frequency fc, which was chosen at the high-frequency jump point of the CRY distribution (i.e., the onset of the non-physical power surge). Training was halted when the high-frequency noise fraction exceeded a statistical baseline ($\mu + 3\sigma$) established during the early smooth-fitting phase. This ensures the model captures the continuous phys- ical envelope while rejecting the discretization noise (Fig. 6, right).

In contrast to simulated data, real-world data is often more limited. We do not have access to large datasets of real cos- mic muons comparable to the 100 million samples generated by CRY for training. To address this, we tested our model using a much smaller subset of CRY-generated data (6,000 samples) and compared the model' s output with CRY' s full dataset (100 million samples). As shown in the figures, even Fig. 2. The schematic overview of the DeepMuon model. The generator $G\theta$ takes a standard normal distribution N (0, I) as the prior input. A linear embedding layer projects the noise into the latent space, followed by a Transformer encoder that extracts global dependencies. Finally, an MLP projection head maps the latent features to the physical space of cosmic ray muons (Energy, Zenith, Azimuth).

Fig. 3 [Figure 3: see original paper]. From left to right show the energy distribution for all events, the energy distribution of muons at a 1° zenith angle, and the energy distribution of muons at a 60° zenith angle, respectively. with a limited amount of training data, our model was able to accurately capture the correct distribution. This suggests that our model is capable of learning the true cosmic muon distri- bution even when working with smaller, real-world datasets.

Inference Speed and Performance Analysis We benchmarked the generation efficiency of DeepMuon against standard simulation tools CORSIKA and CRY. All evaluations generated 2.048 × 107 cosmic ray muon events using an AMD Ryzen 7 5800H CPU (16 threads) and an Fig. 4 [Figure 4: see original paper]. Even with a limited amount of data, DeepMuon is capable of effectively learning the distribution of cosmic ray muons.

Fig. 5. Performance of DeepMuon trained on CORSIKA-generated data (3.5 × 105 muons). From left to right: the overall energy distribution ($ 2/ndf = 134.6/148, KSp = 0.914), theenergydistributionata15°zenithangle(2/ndf = 83.6/78, p = 0.311), andtheenergydistributionata45°zenithangle($2/ndf = 90.7/78, p = 0.154).

Fig. 6. Overfitting prevention via spectral monitoring. Left: The model (red) approximates the physical envelope, ignoring the step-like artifacts of the training data (blue). Right: Training is halted when the high-frequency noise remains within the 3σ safety threshold.

NVIDIA RTX 3090 GPU.Table 1 details the computational performance. Leveraging GPU parallelism, DeepMuon com- pleted the task in merely 16 seconds. Even when restricted to CPU inference, DeepMuon (73 seconds) significantly out- performed CRY (305 seconds). In stark contrast, the compu- tationally intensive CORSIKA simulation required approxi- mately 11 hours (39,701 seconds).

Consequently, DeepMuon achieves a speedup of 19× over CRY and 2481× over CORSIKA on GPU. Notably, even on CPU hardware, our model is approximately 4× faster than CRY. This acceleration is critical for data-intensive applications like underwater muon navigation and absorption radiography, where rapid generation of large-scale datasets is a prerequisite.

While CORSIKA offers a physically smoother energy spectrum, its slow generation speed is impractical for training deep learning models requiring billions of samples. Despite minor "step-like" artifacts, CRY provides a necessary compromise between accuracy and speed, justifying its use as the training target for large-scale experiments.

Table 1. Performance comparison for generating 2.048$×$107 cosmic ray muon events. DeepMuon demonstrates significant speedups on both CPU and GPU hardware compared to traditional methods. Note that CORSIKA and CRY do not support GPU acceleration.

Generator CORSIKA DeepMuon DeepMuon Hardware CPU (16 threads) CPU (16 threads) CPU (16 threads) Speedup 1× (Baseline) 130× 544× 2481× Time (s) 39,701 Imaging Simulation The conventional simulation process for cosmic-ray muon imaging typically involves using cosmic-ray muon gener- ators, such as CORSIKA or CRY, to produce cosmic-ray muons, followed by simulating their passage through mate- rials in GEANT4. As an example, we substitute CRY with DeepMuon as the cosmic-ray muon source and further simu- late the passage of these muons through water in GEANT4, exploring the feasibility of using a deep learning-based muon source for imaging simulations.

To simulate underwater muon distribution data, we con- structed a cubic world of dimensions 60m × 60m × 10m in GEANT4, filled with water. At the top of this world, we intro- duced cosmic-ray muons generated by DeepMuon and simu- lated their passage through the water using GEANT4. Ad- ditionally, to demonstrate DeepMuon' s capability to simulate muon distributions beyond just those at sea level, we designed an underwater muon distribution simulation pipeline. In this setup, DeepMuon directly generates muon distributions at a specific water depth, with further simulation in GEANT4 pro- viding data for even deeper underwater muon distributions.

Specifically, we first train our model using surface muon data generated by CRY, simulating their passage through wa- ter in GEANT4 to obtain underwater muon distributions. We then use this data to further train a model for underwater muon generation, repeating this process to create models for different depths. The pipeline consists of two key steps:

1. Training DeepMuon to learn the underwater muon dis-

tribution simulated by GEANT4.

2. Integrating DeepMuon into GEANT4 to complete the

simulation for deeper underwater muon distributions.

Specifically, we first trained the DeepMuon model using muon distribution data generated by CRY at sea level. Once the DeepMuon muon generator was established, we used it as input for the particle gun in GEANT4. In GEANT4, we created a 60m x 60m x 10m simulation space filled with wa- ter. The particle gun randomly emitted muons from the top of this volume, while a muon detector at the bottom recorded the distribution of muons that had passed through 10 meters of water. After obtaining this distribution data, we used it to train a new DeepMuon model for deeper water conditions, and repeated the process by using this updated model as input for the next simulation step in GEANT4.

1. Result

Using the pipeline described above, we obtained Deep- Muon models that generate underwater muon distributions at depths of 0m, 10m, 20m, 30m, 40m, and 50m. For com- parison, we used CRY to generate sea-level muons and then employed a brute-force method in GEANT4 to simulate the distribution of these muons after passing through 10m and 50m of water. The results were then compared with the 10m and 50m DeepMuon models obtained through our pipeline.

The comparison results are shown in Figures 7–8. It can be observed that at a depth of 10m, the statisti- cal yield from the CRY-generated muons remains sufficiently high. The muon distribution generated directly by our model at 10m depth is highly consistent with the 10m underwater muon distribution obtained from the CRY+GEANT4 simula- tion. This indicates that our model can also serve as a muon source for cosmic ray muon imaging simulations.

However, at a depth of 50m, the time required for the CRY+GEANT4 direct simulation becomes prohibitively long, and the muon flux reaching the detector significantly decreases, making it difficult to observe a smooth and com- plete distribution. As a result, we could only generate a lim- ited number of cases for plotting. Despite this limitation, the overall trend of the distribution still shows a high degree of consistency with our model. Since the generation of deep- water muon distributions in our model relies on the results from shallower depths, this suggests that the distribution pro- duced by DeepMuon during the propagation process in the pipeline does not suffer significant distortion. Therefore, we can conclude that our model is not only capable of simulating muon distributions at sea level, but it also performs well in simulating underwater muon distributions at various depths.

Fig. 7 [Figure 7: see original paper]. Energy distribution of cosmic ray muons at a depth of 10 meters underwater.

Fig. 8 [Figure 8: see original paper]. Energy distribution of cosmic ray muons at a depth of 50 meters underwater.

III. CONCLUSION In this paper, we design DeepMuon, a deep learning–based cosmic-ray muon generator that differs from traditional Monte Carlo and parametric models. Leveraging the optimal- transport-based SWD loss, we enable one-step generation of high-fidelity cosmic-ray muon distributions from a normal distribution using a simple architecture with only a two-layer Transformer encoder. We introduced the Box-Cox transfor- mation to reduce the kurtosis of the cosmic ray muon energy distribution, thereby achieving a distribution with statistical properties close to a normal distribution. This addresses the challenge of deep learning models struggling to learn sharp distributions and provides a feasible solution for generating high-kurtosis distributions in high-energy physics.

DeepMuon is capable of directly learning the distribution data of cosmic ray muons, without the need to incorporate additional physical constraints to achieve accurate simula- tions. We trained separate models for sea-level muon distribu- tions and underwater muon distributions at different depths, demonstrating that DeepMuon, while achieving ultra-high generation speed, can also accurately simulate cosmic ray muon distributions with extremely high precision in various scenarios. This suggests that deep learning methods can serve as an alternative to Monte Carlo and parameterized models for cosmic ray muon simulations, while also highlighting the tremendous potential of SWD in high-energy physics event generation.

Furthermore, we demonstrated that models trained with a small number of examples can still effectively learn the distri- bution of cosmic ray muons, indicating that our model has the potential to directly learn from real data collected by muon detectors, thus leading to higher simulation accuracy. At its core, our model is a distribution transformation model using an optimal transport loss function. We have shown that it can flexibly transform uniform distributions into various types of cosmic ray muon distributions. This implies that in future work, we can not only use it as a generation model, but also as a distribution transformation model to simulate the effects of water layers, rock formations, or other structures on any cosmic ray muon distribution, or to model the impact of high- energy particle beams on detectors in particle physics experi- ments.

IV. BIBLIOGRAPHY [1] L. Dorman, Cosmic rays in the Earth' s atmosphere and under- ground, vol. 303 (Springer Science & Business Media, 2013) [2] S. Procureur, K. Morishima, M. Kuno, et al., Precise charac- terization of a corridor-shaped structure in Khufu' s Pyramid by observation of cosmic-ray muons. Nature Communications, 14(1): 1144 (2023) [3] K.N. Borozdin, G.E. Hogan, C. Morris, et al., Radiographic imaging with cosmic-ray muons. Nature, 422(6929): 277–277 (2003) [4] C. Hagmann, D. Lange, D. Wright, Cosmic-ray shower gen- erator (CRY) for Monte Carlo transport codes, in 2007 IEEE nuclear science symposium conference record, vol. 2 (IEEE, 2007), pp. 1143–1146 [5] D. Heck, T. Pierog, J. Knapp, CORSIKA: An air shower simu- lation program. Astrophysics Source Code Library, ascl–1202 (2012) [6] P. Biallass, T. Hebbeker, Parametrization of the cosmic muon flux for the generator CMSCGEN. arXiv preprint arXiv:0907.5514 (2009) [7] D. Pagano, G. Bonomi, A. Donzella, et al., EcoMug: An Ef- ficient COsmic MUon Generator for cosmic-ray muon appli- cations. Nucl. Instrum. Methods Phys. Res. A, 1014: 165732 (2021). doi: 10.1016/j.nima.2021.165732 [8] J. Osborne, Improving your data transformations: Applying the Box-Cox transformation. Practical Assessment, Research, and Evaluation, 15(1) (2010) [9] C.-C. Pan, X. Dong, Y.-C. Sun, et al., Event Generation and Consistence Test for Physics with Sliced Wasserstein Distance. arXiv preprint arXiv:2310.17897 (2023) [10] S. Otten, S. Caron, W. de Swart, et al., Event generation and statistical sampling for physics with deep generative models and a density information buffer. Nature communications, 12(1): 2985 (2021) [11] Y. Alanazi, N. Sato, P. Ambrozewicz, et al., A survey of machine learning-based physics event generation. Technical re- port, Thomas Jefferson National Accelerator Facility (2021) [12] I. Goodfellow, J. Pouget-Abadie, M. Mirza, et al., Generative adversarial nets. Advances in neural information processing systems, 27 (2014) [13] D.P. Kingma, Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013) [14] G. Peyré, M. Cuturi, Computational Optimal Transport. arXiv preprint arXiv:1803.00567 (2020) [15] G. Carminati, M. Bazzotti, A. Margiotta, et al., Atmospheric MUons from PArametric formulas: a fast GEnerator for neu- trino telescopes (MUPAGE). Computer Physics Communica- tions, 179(12): 915–923 (2008) [16] N. Su, Y. Liu, L. Wang, et al., A comparison of muon flux models at sea level for muon imaging and low background ex- periments. Frontiers in Energy Research, 9: 750159 (2021) [17] A.-Y. Cheng, H. Cai, S. Chen, et al., Application of deep learn- ing methods combined with physical background in wide field of view imaging atmospheric Cherenkov telescopes. Nuclear Science and Techniques, 35(4): 84 (2024) [18] S. Agostinelli, J. Allison, K. Amako, et al., Geant4—a simulation toolkit. Nucl. Instrum. Methods Phys. Res. A, 506(3): 250–303 (2003). doi: 10.1016/S0168-9002(03)01368-8 [19] H.K.M. Tanaka, C. Bozza, A. Bross, et al., Muography. Nature Reviews Methods Primers, 3(1): 88 (2023) [20] S. Kolouri, G.K. Rohde, H. Hoffmann, Sliced wasserstein dis- tance for learning gaussian mixture models, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recog- nition (2018), pp. 3427-3436 [21] R. Di Sipio, M.F. Giannelli, S.K. Haghighat, et al., A Generative-Adversarial Network Approach for the Simulation of QCD Dijet Events at the LHC. Proc. Sci. LeptonPhoton, 50 (2019) [22] S. Otten, S. Caron, W. de Swart, et al., Event generation and statistical sampling for physics with deep generative mod- els and a density information buffer. Nature communications, 12(1): 2985 (2021) [23] J. Ho, A. Jain, P. Abbeel, Denoising diffusion probabilistic models. Advances in neural information processing systems, 33: 6840-6851 (2020) [24] C.-C. Pan, X. Dong, Y.-C. Sun, et al., SwdFold: A Reweight- ing and Unfolding method based on Optimal Transport Theory. arXiv preprint arXiv:2406.01635 (2024)

Note: Figure translations are in progress. See original paper for figures.