Machine Learning: Science and Technology

PAPER • OPEN ACCESS

Photon detection probability prediction using one-dimensional generative neural network

To cite this article: Wei Mu *et al* 2022 *Mach. Learn.: Sci. Technol.* **3** 015033

IOP Publishing Mach. Learn.: Sci. Technol. 3 (2022) 015033 https://doi.org/10.1088/2632-2153/ac58e2

MACHINE LEARNING Science and Technology

CrossMark

OPEN ACCESS

RECEIVED
20 September 2021

REVISED
18 November 2021

ACCEPTED FOR PUBLICATION
25 February 2022

PUBLISHED
18 March 2022

PAPER

Photon detection probability prediction using one-dimensional generative neural network

Wei Mu*, **Alexander I Himmel** and **Bryan Ramson**

Neutrino Division, Fermi National Accelerator Laboratory, Wilson Street and Kirk Road, Batavia, IL 60510, United States of America
* Author to whom any correspondence should be addressed.

E-mail: wmu@fnal.gov

Keywords: 1D generative neural network, photon detection, liquid argon detector, neutrino, dark matter, level of detail

Abstract

Photon detection is important for liquid argon detectors for direct dark matter searches or neutrino property measurements. Precise simulation of photon transport is widely used to understand the probability of photon detection in liquid argon detectors. Traditional photon transport simulation, which tracks every photon using the Geant4 simulation toolkit, is a major computational challenge for kilo-tonne-scale liquid argon detectors and GeV-level energy depositions. In this work, we propose a one-dimensional generative model which efficiently generates features using an OuterProduct-layer. This model bypasses photon transport simulation and predicts the number of photons detected by particular photon detectors at the same level of detail as the Geant4 simulation. The application to simulating photon detection systems in kilo-tonne-scale liquid argon detectors demonstrates this novel generative model is able to reproduce Geant4 simulation with good accuracy and 20 to 50 times faster. This generative model can be used to quickly predict photon detection probability in huge liquid argon detectors like ProtoDUNE or DUNE.

1. Introduction

Liquid argon (LAr) is a popular detector medium for direct dark matter searches [1–3] and neutrino property measurements [4–6]. Its excellent scintillation properties have been employed by dark matter search experiments for energy reconstruction and background rejection. LAr scintillation light detection will advance the physics goals of neutrino experiments by improving detector calorimetric and position resolution, enabling the study of astrophysical neutrinos and enhancing the physics reach of oscillation analyses. In order to take the advantage of information provided by LAr scintillation light, photon detection probability in LAr detectors must be well understood.

Traditionally, photon detection probability is explored by simulating the transport of photons in detectors using Geant4 [7]. However, such simulation is extremely challenging for experiments using huge LAr detectors that record GeV-level energy depositions due to limited computing resources. Modern machine learning techniques have enabled new ways to emulate the results from full Geant4 simulation and a generative model is one of the most promising approaches for learning the true distribution from training samples so as to generate new data points with variation [8]. However, while generative models based on deep neural networks (DNNs) have shown great promise in generating accurate predictions, they can be too slow when deployed without GPUs, motivating the development of a novel generative model which can efficiently predict photon detection probabilities at lower computational cost.

In this paper, we demonstrate that a one-dimensional generative neural network (1D GENN) is capable of bypassing photon transport simulation and rapidly predicting precise photon detection probabilities based only on the scintillation vertex. We first discuss common features of photon detection systems for LAr detectors and propose a general architecture for the GENN in section 2. Subsequently, we instantiate and train GENN models specifically for photon detection systems of two large scale LAr detectors. In section 3, we evaluate the performance of the GENN-based photon detection probability predictions, in comparison to full Geant4 simulations, on a number of metrics including: the capability to smoothly interpolate, the precision of the prediction, and the speed of inference on CPUs. In addition, we evaluate the scalability of the GENN model to both higher precision and to larger systems. Finally, we summarize our work and discuss the potential generalization of the GENN model in section 4.

2. One-dimensional generative model

Aiming at building a fast and precise generative model for photon detection probability prediction, we extract common features from photon detection systems of LAr detectors, propose a novel network architecture that is able to conditionally generate the photon detection probabilities based on the scintillation vertex, and instantiate the model for two large scale LAr detectors.

2.1. Model architecture

Common photon detection systems of LAr detectors consist of a series of photon detectors distributed in arrays and deployed on the 'walls' of the detector. Photons, emitted from a vertex where a particle deposits energy illuminate specific photon detectors and form a hit pattern on the photon detection system. The number of total detected photons, being correlated to the deposited energy of the particle, can be used as a calorimetric energy measurement and improve the detector energy resolution. The distribution of the illuminated photon detectors, which depends on the location of the scintillation vertex, can be used to locate the two-dimensional (2D) position of the interaction point. The Geant4 toolkit uses a Monte-Carlo (MC) method to simulate photon transport inside a detector and produce the number of photons detected by particular photon detectors, which predicts the photon detection probability of the detector. This MC-based method is challenging for huge LAr detectors and GeV-level energy depositions, so the prediction from the full Geant4 simulation might be emulated, in a more efficient way, by generative models. They can directly map the energy deposition, with specific amount and position, to the hit pattern on the photon detection system.

Recently, generative models using DNNs have shown the ability to randomly produce new high resolution images that mimic the details of true images [9–11]. However, DNNs inference on CPUs is slow. Millions of simulated events with billions of hit pattern images are required for physics analyses, suggesting significant computing resources. Given that modern LAr detectors will deploy thousands of photon detectors and the photon detection systems of LAr detectors will generate hit pattern images consisting of only thousands of pixels, it could be a costly and time consuming process which is far less than the images processed by common generative models. In addition, the images to be generated strongly depend on the location of the scintillation vertex, which makes it possible to fill in features to an image in a more efficient way. This motivated us to explore whether generative models based on shallow neural networks with novel architectures could reproduce the features of the photon detection systems of LAr detectors.

Considering a LAr detector in a three-dimensional (3D) Cartesian coordinate system, a photon detector array is deployed perpendicular to the x-axis, where photon detectors are aligned in an $m \times n$ array along the y-axis and z-axis. Assuming a certain number of photons are emitted from a scintillation vertex (x_s, y_s, z_s), it is a reasonable intuition that the number of detected photons by particular photon detectors depends on the coordinate x_s and the distribution of the illuminated photon detectors along row or column in the yz-plane is determined by the coordinate y_s or z_s.

According to the common photon detection system design, we propose a general GENN architecture with three input layers, named pos_x, pos_y, and pos_z, each having one neuron corresponding to the coordinate of the scintillation vertex, x_s, y_s , and z_s. The GENN has one output layer, named vis_{full}, with number of neurons equal to the number of photon detectors. There are three first hidden layers, connecting to the three input layers respectively. The hidden layer connected to pos_x, named vis_{int}, has one neuron as a factor to normalize the number of total detected photons. The hidden layer connected to pos_y or pos_z, named vis_{col} or vis_{row}, has n or m neurons, being used to predict the photon detection probabilities by detectors in each column or row. The outer product of vis_{col} and vis_{row}, composing an OuterProduct-layer and named vis_{tmp}, represents the 2D-distribution of the photons in the photon detector array, which is unrolled without particular concern for ordering to 1D data structure. The multiplication of vis_{int} and vis_{tmp}, forming a Multiply-layer named vis_{arr}, reflects the features of the hit pattern on the photon detector array and predicts the photon detection probability of each detector corresponding to a particular scintillation vertex. Finally the Multiply-layer is mapped to the output layer, so that the distribution of the photon detectors is able to be decoupled from their sequence number. In practice, the number of neurons on vis_{col} and vis_{row} (n or m) can be tuned and optional layers can be introduced before the output layer, such as Normalization layers, to get optimal network architecture (see section 3.4). The general GENN architecture we proposed is illustrated in figure 1.

Unlike common generative models, which use the upsample (UpSampling2D) layer or the transpose convolutional (Conv2DTranspose) layer for the input dimension expansion, the model proposed in this

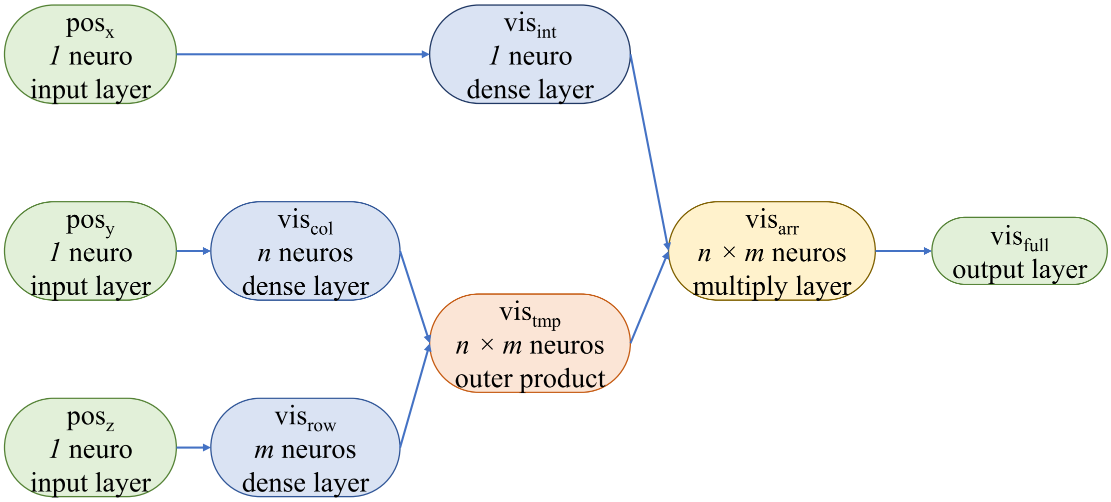

Figure 1. General GENN architecture. pos_y-layer and pos_z-layer are used to predict the photon distribution on the photon detection system, while pos_x-layer is a normalization factor for the number of total detected photons. OuterProduct layer expands the dimension of the inputs.

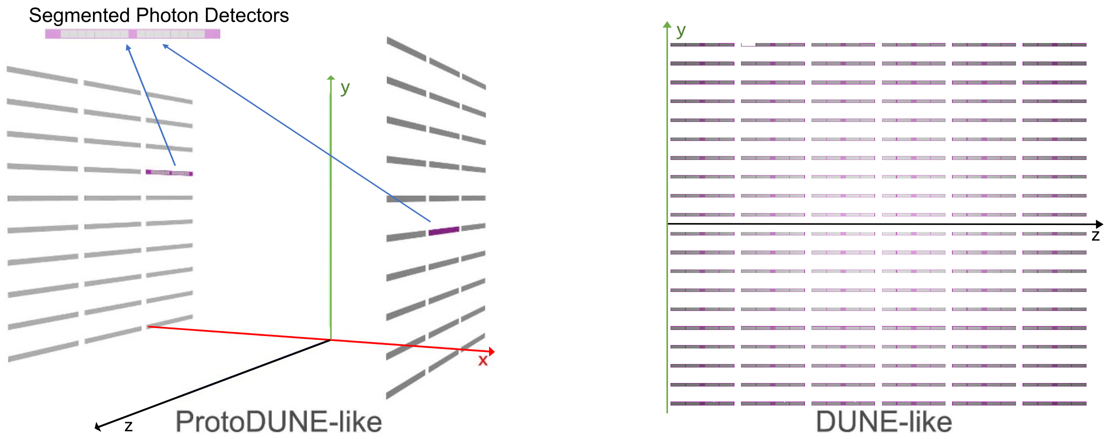

Figure 2. Photon detection systems in ProtoDUNE-like (left) and DUNE-like geometries (right).

work uses a OuterProduct layer to produce features and expand the dimension of inputs, which significantly reduces the inference time on CPUs. This model starts with a 3D position, passes 1D data from layer to layer, and finally gives rise to a prediction in a 1D data structure on the output layer. We will discuss the advantage of the 1D data structure in section 2.2.

This general GENN model has been applied to photon detection systems of ProtoDUNE-like [4, 12, 13] and DUNE-like geometries [14]. Two photon detection arrays are deployed on two 'walls' in ProtoDUNE-like geometry. In each array, two different types of photon detectors are used. 30 bar-like light collectors are aligned in 3×10 array [15], where one bar is segmented into 16 silicon photomultipliers, as showed in figure 2 on the left. Therefore, in each array, there are 29 bar-like and 16 segmented photon detectors. In total, 90 photon detectors are deployed in this geometry. The active volume sits between the photon detection systems, with an opaque wall halfway between them, so each wall detects the photons emitted on its side of the geometry. In the DUNE-like geometry (a segment of a DUNE detector), 480 photon detectors are deployed on a single 'wall' in the middle of the volume, forming a yz-plane. The photon detectors are rectangular and aligned with the z-direction forming 20 wide-spaced rows, each of which has 24 photon detectors aligned in the y-direction placed end-to-end in six groups, as shown in figure 2 on the right. This photon detection system sits in the middle of the active volume, and so is able to detect photons emitted from both sides.

We instantiate 1D GENN models for above two photon detection systems to verify the stability and generalizability of this general model. The networks have been implemented in the framework of

KERAS ([16]) on top of TensorFlow ([17]), and the code for the two instantiated models is available on GitHub ([18]).

2.2. Loss function

Neural networks are generally trained using the gradient descent algorithm, which requires a loss-function to calculate model error and update model parameters using the back propagation algorithm. Common generative models generate 2D images, which makes it challenging to choose an optimal loss-function for model training since there is no consensus as to which measure best captures the feature difference between two 2D images. This motivates the generative model to be trained in the generative adversarial network (GAN) framework [19, 20], where a discriminator network is used to evaluate the output from the generative model. The GAN framework has led to success in many applications, but objective and quantitative evaluation of GAN generative model remains an open question. So far, there is no general agreement upon algorithms to find the Nash-equilibrium between the discriminator and the generative model [21], and it relies on human eyes to determine performance of the generative model, which results in an unstable training procedure. We solve this problem by training the GENN models with a clearly defined loss-function instead of within the GAN framework.

It is natural to assume generative models would learn features from 2D images in the similar way as human brain. However, neural networks might view data in a different way and learn features from flattened 2D images: even non-spatially-representative 1D-vectors. In this case, the problem to be solved by generative models can be considered a multiple regression prediction problem, and the 1D data structure allows more straightforward similarity metrics [22–24].

The most commonly used metric for a multiple regression problem, mean squared error (MSE) or mean absolute error (MAE), tends to produce 'regression to the mean' situation, therefore, we design a loss-function based on physics performance. As the value of each element of the 1D-vector predicted by the GENN model is photon detection probability, the Kullback–Leibler (KL) divergence (D_{KL}) provides a good starting point, which is defined as:

$$D_{\mathrm{KL}}(P||Q) = \sum_{x} P(x) \log \frac{P(x)}{Q(x)}, \tag{1}$$

where $P(x)$ corresponds to the 'test' distribution, and $Q(x)$ the 'true' distribution. However, the KL-divergence has a problematic property that D_{KL} diverges in regions where $Q(x)$ has non-null value and $P(x)$ has null value. In this application, this condition occurs when the photon detectors 'really', according to the simulation, detect a non-zero number of photons but the model predicts no photons are detected. Regions distant from the photon detectors might have quite low photon detection probability, but typically not zero, which is why a situation where the simulation predicts a small but non-zero probability, while the model predicts exactly zero. As this condition can occur frequently, this loss function is not appropriate for this case. In addition, because D_{KL} is value-weighted by the probability $P(x)$, the D_{KL}-loss-function leads to a selection bias where the error measured by low-value elements is underrated.

Motivated by one of the variations of KL-divergence: Jensen–Shannon (JS) divergence, we propose a loss-function using a variational KL-divergence D_{vKL}, defined as:

$$D_{\mathrm{vKL}}(P||Q) = \left| \sum_{x} \Big(P(x) - Q(x)\Big) \log \frac{P(x)}{Q(x)} \right|, \tag{2}$$

where x stands for the sequence number of photon detectors, $P(x)$ the 'test' 1D-vector from GENN model prediction, and $Q(x)$ the 'true' 1D-vector from Geant4 simulation. The D_{vKL}-loss-function is symmetric and benefits from all the advantages which a symmetric metric has for generative model training [25]. It is well behaved no matter whether either $P(x)$ or $Q(x)$ is small or not, so that it satisfies the required properties for a strict metric for similarity measurement. Because D_{vKL} is value-weighted by the difference between $P(x)$ and $Q(x)$, minimising D_{vKL} avoids producing samples that are very unlikely under $Q(x)$. Empirically, this loss function makes the model to converge to the condition where the similarity between $P(x)$ and $Q(x)$ is good and acceptable. The training of the sample GENN models using the D_{vKL}-loss-function proves D_{vKL} successfully captures properties of this problem.

2.3. Training

We train the two GENN models using samples from Geant4 simulation. We first generate 1000 000 light sources, uniformly distributed within the detectors' active volume, at vertices (pos_x, pos_y, pos_z). From each vertex, 1000 000 photons are emitted. Those photons are transported inside the active volume, fully simulated by Geant4, and a certain number of photons are detected by particular photon detectors. For each

Table 1. 3D-position of the photon detectors (PDs) being studied. Three groups of light sources are produced and uniformly distributed along x, y, and z direction.

ProtoDUNE-like		DUNE-like	
PD#	Position (cm)	PD#	Position (cm)
10	363, 568, 347	100	0.05, 591, 781
14	363, 331, 347	107	0.05, 155, 781
17	363, 94, 347	344	−0.05, −280, 781
05	363, 272, 579	005	0.05, 280, 1357
29–44	363, 272, 428 ∼ 265	095	0.05, 280, 843
24	363, 272, 115	185	0.05, 280, 3169
Position of Light Sources (cm)			
−400 ∼ 400, 420 ± 10, 345 ± 10		−400 ∼ 400, 300 ± 10, 780 ± 10	
150 ± 10, 0 ∼ 600, 345 ± 10		150 ± 10, −600 ∼ 600, 780 ± 10	
150 ± 10, 420 ± 10, 0 ∼ 700		150 ± 10, 300 ± 10, 0 ∼ 1400	

scintillation vertex, we get an 'image' of the hit pattern on the photon detectors, which is organized as a 1D-vector. We produce the images by simulating the photon transport in ProtoDUNE-like and DUNE-like geometries using the LArSoft toolkit [26] which uses Geant4. Combining the scintillation vertex and the image, we build the training samples in the structure: $('x' : pos_x, 'y' : pos_y, 'z' : pos_z, 'image' : image)$. We emphasize that the sequence for elements in the 1D-vector for 'image' is plainly organized by the sequence number of the photon detectors instead of flattening the human-friendly spatial-representative image.

We train the GENN models with a batch size of 4096, using the *Adam* optimizer [27] on the Wilson Cluster ([28]) at Fermilab with two NVIDIA Tesla K40 GPUs. In order to help the model converge quickly and stably, we use a learning rate scheduler to reduce the learning rate during training. The learning rate is initialized as: lr= 0.0002, and decays following: $\mathrm{lr} = \mathrm{lr} \times 0.997^{\mathrm{epoch}}$. In order to help the model get out of potential saddle points, we reset the learning rate to its initial value every 1000 epochs. The training procedure is monitored with the metric 'mean absolute error of the validation sample' to avoid overtraining. The training procedure stops automatically when the monitored metric 'validation sample loss' stops improving after 500 epochs, which was typically less than 10 000 epochs and within eight hours.

3. Performance

Once the networks were trained, we evaluated the GENN models' feature-learning capability and bench-marked their performance within the LArSoft framework.

3.1. Feature-learning capability

A photon detector observes more photons when a light source is closer to it. Hence, the probability of photon being detected by the particular photon detector, or the so-called 'visibility', is higher. Consequently, the visibility of a photon detector reaches a peak when the light source moves towards it and reduces when that moves away. In addition, both the GENN prediction and Geant4 simulation are expected to see a smooth variation in the total amount of light detected by particular photon detectors when the light source passes by.

In order to verify the models have learned desired characteristics, we compose three groups of light sources in LArSoft, which are inside detectors' active volume and distributed uniformly along x, y, and z direction respectively, and predict the visibility using GENN and Geant4. For each group, we study the visibilities on three photon detectors. The position of the photon detectors and the light sources are shown in table 1. The visibilities, produced by GENN and Geant4, are plotted in figure 3 for comparison.

Figure 3 shows the visibilities of the photon detectors change smoothly when the light source 'moves', which matches the general trends of what is produced from full Geant4 simulation. This test indicates that the GENN models are able to learn features from the dataset and interpolate smoothly, instead of only memorizing the training samples. Figure 3 indicates GENN model can generalize what it has learned and predict probabilities of photon detected by particular photon detectors with precision similar to Geant4's.

3.2. Prediction precision

We evaluate the precision of the GENN prediction in the LArSoft framework. After being trained, the models and weights are frozen to computable graphs which are loaded by with the TensorFlow C++ API. We simulate the transport of samples representative of the physics of interest to neutrino experiments in Geant4 and record the tracks, where we use 200 MeV monoenergetic muons starting from a specific position in the

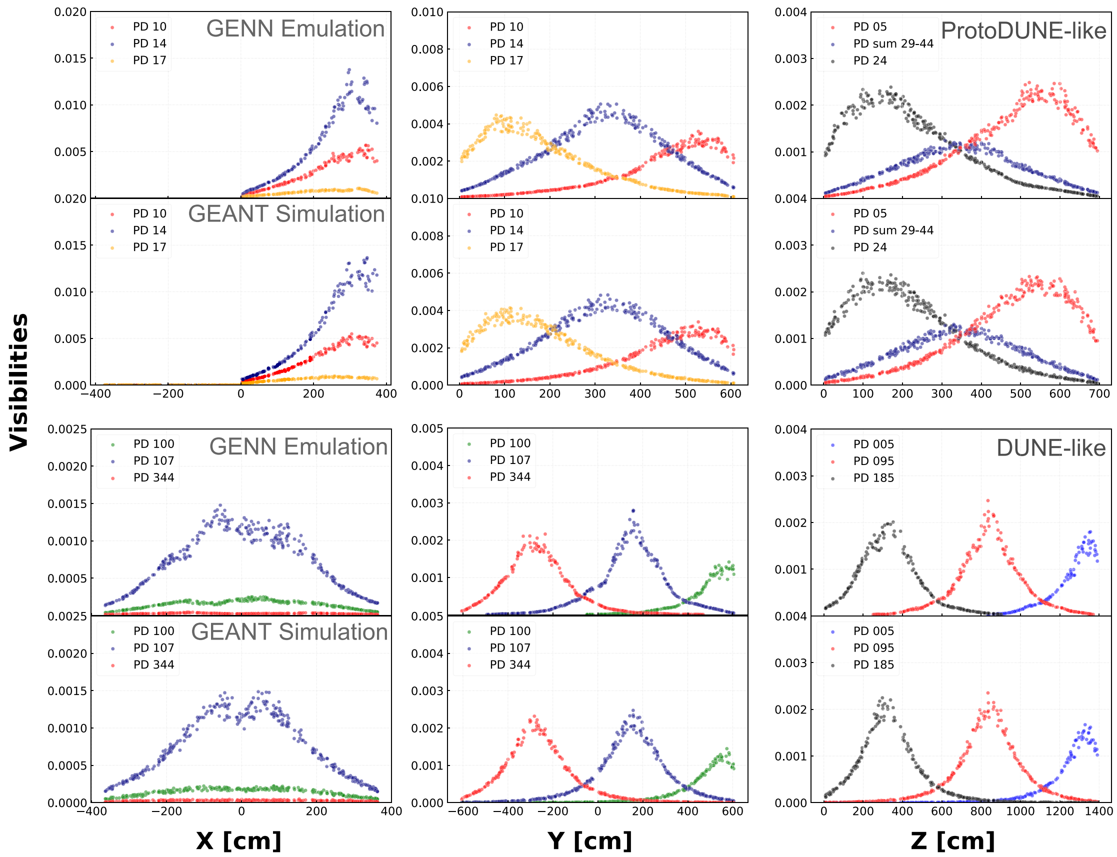

Figure 3. Feature-learning capability. In each figure, the horizontal axis is the position of the light sources along specific directions: x, y, and z direction, while the vertical axis shows the photon detection probabilities (visibilities), on particular photon detectors (PDs). The top two rows show the visibilities in ProtoDUNE-like geometry while the bottom two are for the DUNE-like geometry. The visibilities from GENN prediction are plotted on the first and third rows and those from Geant4 simulation are on the second and forth rows. Note: positions of particular PDs and light sources are showed in table 1.

ProtoDUNE-like geometry and supernova neutrinos uniformly distributed in the DUNE-like geometry. Along the tracks, there are thousands of steps where the particle deposits energy, which we consider as scintillation vertices. We construct light sources at each scintillation vertex, where the number of photons emitted is calculated based on the deposited energy. Finally, we use GENN models to predict and Geant4 simulation to produce photon detection probabilities with respect to individual light sources. We then compare the performance of GENN prediction to Geant4 simulation at the same level of detail, which means the GENN prediction does not lose any stepping information along the tracks.

The number of total detected photons, used for event energy reconstruction, is the critical information we want to extract from the photon detection system. We obtain the number of total detected photons from a scintillation vertex by summing up the number of detected photons on each photon detector, and calculate the relative difference between the two approaches for each vertex. The distribution of these relative differences is shown in figure 4, along with a Gaussian fit to the peak. The discrepancy between the Gaussian fit and the histogram is mainly due to the fact that the latter is the standard deviation of the full distribution, which has non-Gaussian tails.

It shows that the GENN model for the ProtoDUNE-like geometry reproduces the total number of photons detected from each scintillation vertex with a resolution of 3.3%, and a mean 0.4% less than the Geant4 simulation, while the model for DUNE-like geometry has a resolution of 3.6% and on average produces 1.6% fewer photons. Considering the high statistics, the uncertainties on the resolution and mean values are negligible. We estimate the fraction of scintillation vertices whose deviation is larger than 10% is 8.95% in ProtoDUNE-like geometry and 7.69% in DUNE-like geometry. It indicates the 1-neuron vis_{int}-layer is able to accurately normalize the number of total detected photons.

Besides the number of total detected photons, we need the model to predict the photon distribution on photon detectors, which is valuable for position reconstruction of the scintillation vertex. Figure 5 shows the number of photons detected by each photon detector for both the Geant4 simulation and the GENN prediction. Giving similar number of total detected photons, both methods also give similar photon distributions for both photon detection systems. For the DUNE-like geometry, where identical photon

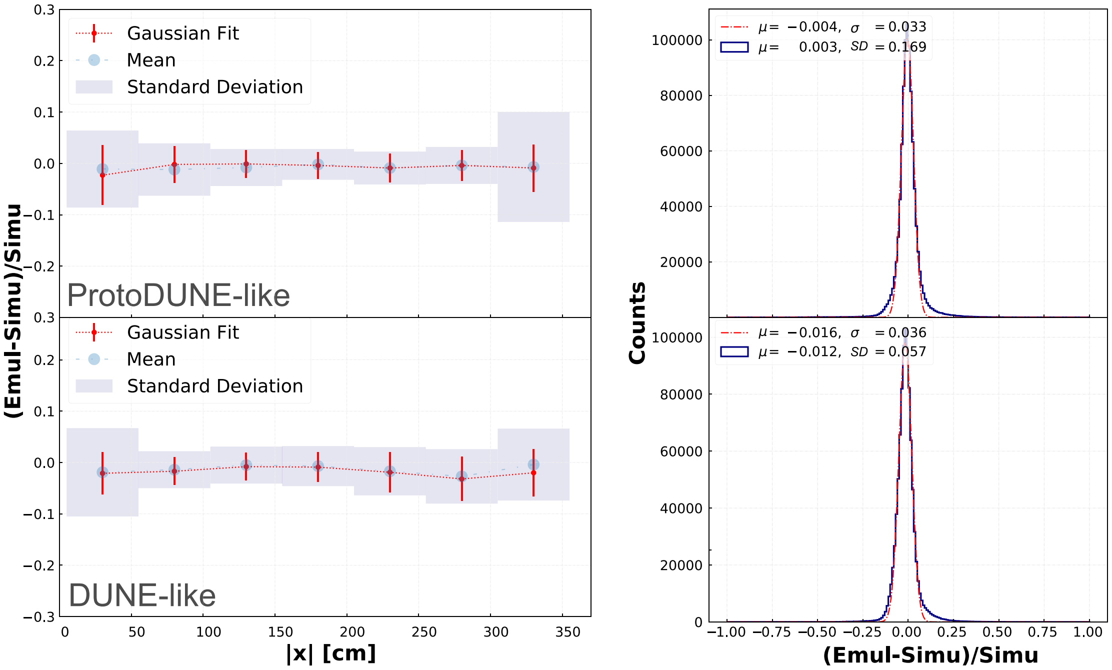

Figure 4. Discrepancy of total detected photons per scintillation vertex. Left: discrepancies variation along the *x*-direction, which is the distance to the photon detector array. Right: overall discrepancies, where the red dashed line is a Gaussian fit. Comparison for ProtoDUNE-like geometry is on the top and DUNE-like geometry on the bottom. Note, due to the large sample sizes, the uncertainties on the mean values and resolutions are negligible.

sensors are symmetrically distributed in the photon detector array, the GENN gives an accurate prediction. Although the photon detection system deployed in ProtoDUNE-like geometry contains two different types of photon detectors, the GENN model can still learn key features from training samples, and gives a reasonable prediction. It suggests OuterProduct-layer can be well trained with the 1D data structure, even in a complex case.

Using the 1D data structure also allows us to calculate some explicit statistical metrics, such as Chi-square distance, Euclidean distance, JS divergence, and MAE, for similarity measures. It also allows us to quantitatively compare the performance between different emulation or simulation approaches, but this is beyond the scope of this paper.

3.3. Inference performance

One of the goals for the GENN model is to predict the photon detection probabilities at high speed using a CPUs. To measure the computing performance of GENN models, we use the method mentioned in section. 3.2 to build light sources along the track of ionizing particles: 200-MeV monoenergetic muons, 2-GeV monoenergetic electrons, and supernova neutrinos. We load those light sources in LArSoft to run GENN prediction and Geant4 simulation for benchmarking. In table 2, we present the CPU time per event, per image, and per photon to compare the computing performance between the two approaches. Thanks to the OuterProduct-layer and the Multiply-layer, the GENN model introduced in this paper is lightweight, and the GENN prediction speed is 20 to 50 times faster than Geant4 simulation while keeping same level of detail on particle tracks, such as number of energy depositions, and precision.

The important information we extract from the test is: GENN prediction run time only depends on the complexity of the models, while Geant4 simulation run time increases significantly with detector volume. The total CPU time for an event will be determined by the number of photons and detector volume for Geant4 simulation, and only by the number of energy deposition vertices along the particle's track for GENN prediction. Since the granularity of the energy depositions along the track can be adjusted, the GENN prediction can be accelerated even further at the expense of less detail on the particle tracks. For instance, by combining some neighboring vertices, the total prediction time by GENN prediction will be much less, since less 'images' are produced, while the total simulation time by Geant4 will be the same since same amount of photons transport in the detector. As the step length is tiny comparing to the detector granularity, it is possible to get faster inference by combining few neighboring vertices without impact the detector spatial resolution.

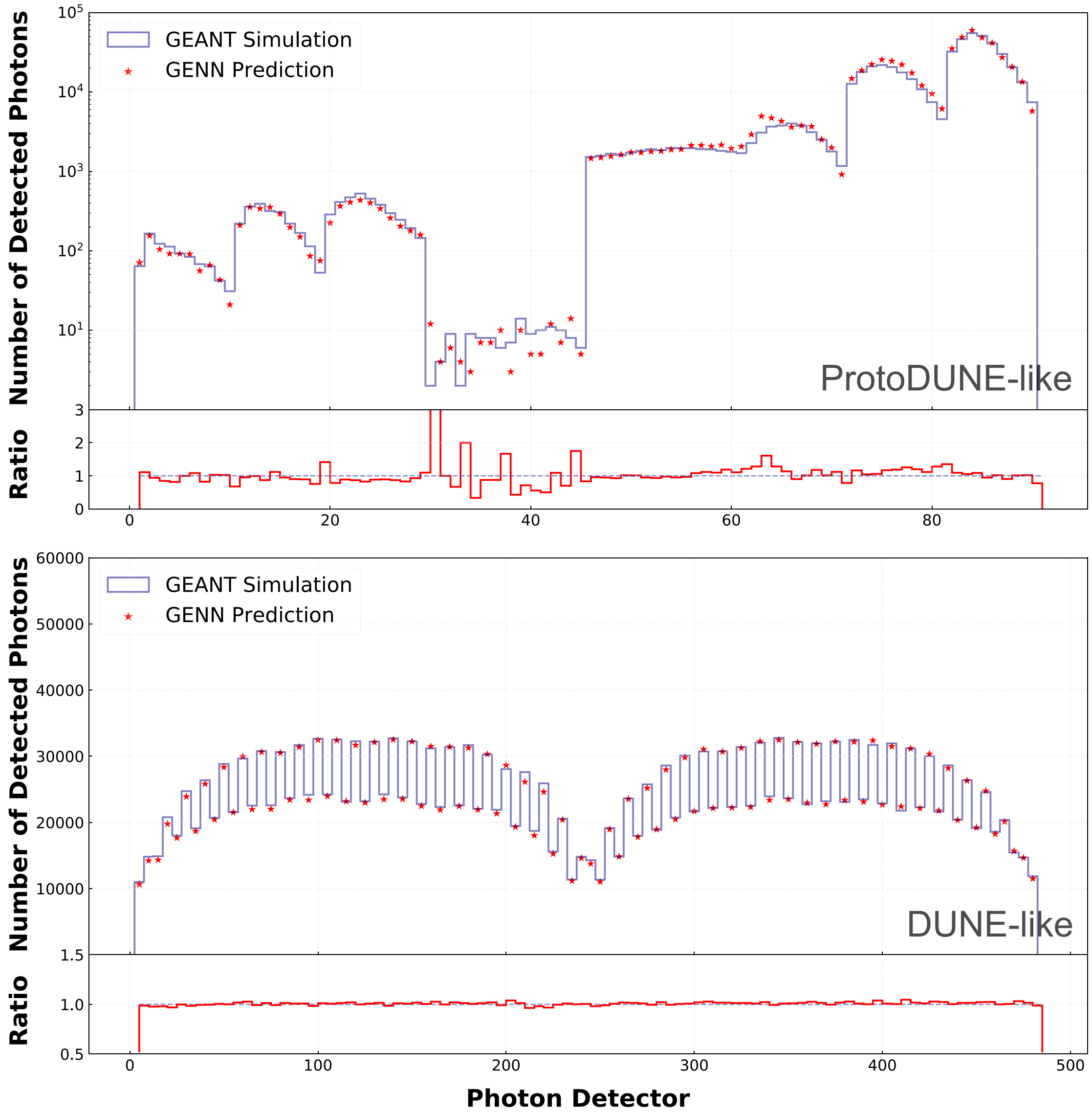

Figure 5. Distribution of the detected photons on photon detection system. Top: the sum of detected photons on photon detectors from 200 MeV monoenergetic muon events starting from one specific position in the ProtoDUNE-like geometry. The response is much lower and scattered around photon detector #40 is caused by the fact that those photon detectors are the segmented ones which are much smaller than the bar-like light collectors. Bottom: the sum of detected photons on photon detectors from supernova neutrinos uniformly distributed in the DUNE-like geometry.

Table 2. CPU time for Geant4 simulation and GENN prediction. Events used for the test: 200 MeV monoenergetic μ, 2 GeV monoenergetic e^-, and Supernova ν_e. All benchmarks for CPU time are performed on DUNEGPVM at Fermilab where 2.4 GHz 4-core Intel® Core™ Processor (Broadwell) are deployed.

		Geant4 Simulation			Genn Prediction		
		(s/event)	(ms/image)	(μs/photon)	(s/event)	(ms/image)	(μs/photon)
ProtoDUNE-like	μ	3.3 ± 0.5	3.71 ± 0.51	34.32 ± 4.72	0.14 ± 0.04	0.15 ± 0.04	1.43 ± 0.40
	e^-	61.1 ± 0.5	3.24 ± 0.03	44.54 ± 0.39	2.83 ± 0.10	0.15 ± 0.01	2.06 ± 0.08
DUNE-like	ν_e	127.4 ± 2.9	10.12 ± 0.23	85.05 ± 1.92	2.65 ± 0.15	0.21 ± 0.01	1.77 ± 0.10
	e^-	103.6 ± 2.9	6.76 ± 0.19	75.62 ± 2.13	2.74 ± 0.09	0.18 ± 0.01	2.00 ± 0.07

Another key advantage of the GENN model is that the model inference requires relatively little memory. The samples for ProtoDUNE-like and DUNE-like geometries show the required memory for the model inference is around 15% of the Geant4 simulation. Further, this memory use is not directly correlated to the volume of the detectors, unlike using lookup libraries where the available memory on the machine limits the potential granularity (and hence precision).

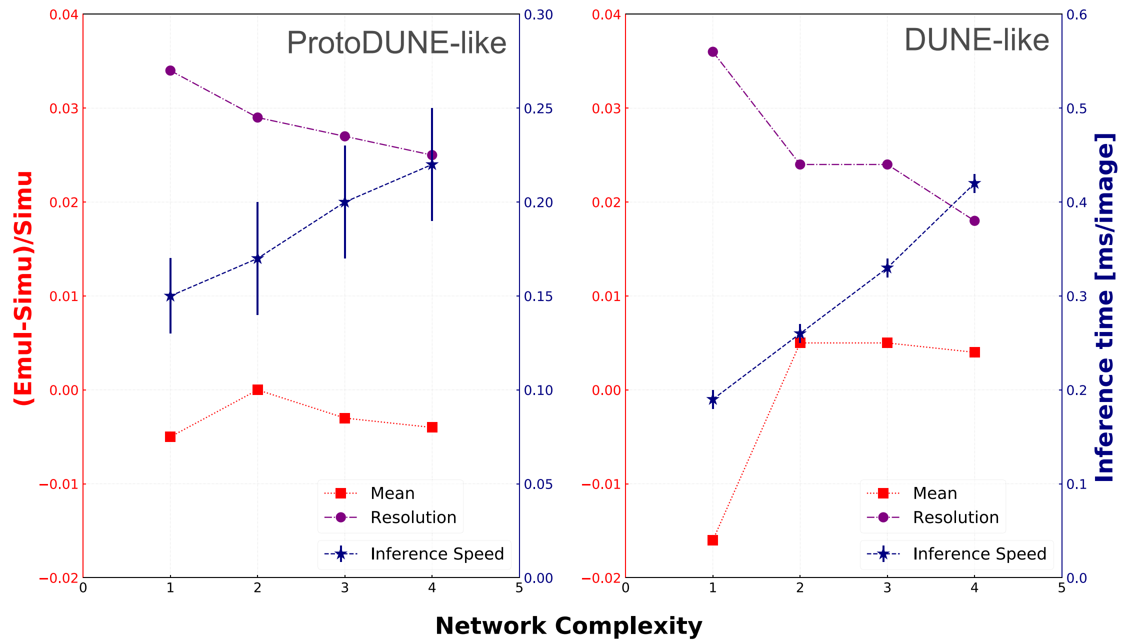

Figure 6. Precision (y axis on the left), inference time (y axis on the right), and neural network complexity.

Table 3. Inference time (CPU) for difference photon detection systems.

Geometry	ProtoDUNE-like		DUNE-like				
Num. of PDs	60	90	120	240	480	960	1920
Time (ms/image)	0.11 ± 0.02	0.15 ± 0.03	0.14 ± 0.01	0.17 ± 0.01	0.21 ± 0.01	0.41 ± 0.01	0.87 ± 0.01

3.4. Scalability

While the GENN model already looks promising, it is possible to add more 'optional' layers for higher precision, with a trade-off that the inference time increases. We conduct an empirical study to quantify both inference time and precision at different levels of network complexity. In particular, we construct GENN models at four levels of complexity by introducing 1 to 4 additional dense layers. The comparison of the precision and inference performance between the four GENN models is showed in figure 6. More advanced network layers may give better performance improvement per loss of speed, but the fundamental relationship remains the same. Our observation indicates a balance between the inference speed and precision must be found when choosing the depth of the network.

As the GENN model prediction run time depends on the network complexity, we benchmark the inference performance for photon detection systems with different number of photon detectors. The metrics in table 3 reveal the inference time increases linearly with the number of photon detectors. The GENN model still outperforms Geant4 simulation, at least, by a factor of 10 even when 1920 photon detectors are deployed in DUNE-like geometry.

4. Conclusion and outlook

In this paper we present a novel generative model for photon detection probability prediction. This GENN model uses an OuterProduct-layer to predict the photon distribution on photon detectors and a single-neuron layer to normalize the photon intensity. This architecture realizes the 'deconvolution' from the scintillation vertex to the photon detection probability and makes the model prediction precise and fast. The proposed loss-function D_{vKL} using a variational KL-divergence function gives good results in training the generative model, it is possible to be generalized for common 1D generative models training.

The model built for ProtoDUNE-like photon detection system demonstrates that shallow neural networks are able to learn features from training samples represented by 1D data structures, even for complex photon detection systems. The sample for DUNE-like photon detection system indicates the GENN model gives fast and precise prediction of photon detection probability using limited memory, showing it can be a powerful new tool to bypass the full Geant4 simulation in a production environment, especially for

LAr detectors with huge volumes, such as the DUNE far detector. The successful application to the ProtoDUNE-like and DUNE-like geometries shows this general GENN architecture is stable and easy to generalize, at least, for different photon detection systems in liquid argon detectors.

Our future attention will focus on incorporating the most recent cutting-edge neural network architecture to improve the prediction precision and the inference speed. We will also develop new photon simulation strategies when using GENN models, guided by the physics goals for dark matter experiments or neutrino experiments.

While our primary effort will be to keep improving this model application for photon detection probability prediction in large scale LAr detector, this GENN model and the D_{vKL}-loss-function are quite general. We will continue studying the the stability and generalizability of the model and the loss-function in other contexts where the training data can be represented as a 1D vector and the GAN framework might not work efficiently [29].

Data availability statement

The data generated and/or analysed during the current study are not publicly available for legal/ethical reasons but are available from the corresponding author on reasonable request.

Acknowledgments

The authors thank S Alonso Monsalve, E Church, T Junk, A M Szelc, L H Whitehead, and T Yang for helpful comments and discussion. This work is supported by the U.S. Department of Energy through the Early Career Award Program. This manuscript has been authored by Fermi Research Alliance, LLC under Contract No. DE-AC02-07CH11359 with the U.S. DOE, Office of Science, Office of High Energy Physics and prepared using the resources of the Fermi National Accelerator Laboratory.

ORCID iDs

Wei Mu https://orcid.org/0000-0002-5849-5689
Alexander I Himmel https://orcid.org/0000-0003-1703-7486

References

[1] Calvo J *et al* 2017 Commissioning of the ArDM experiment at the Canfranc underground laboratory: first steps towards a tonne-scale liquid argon time projection chamber for Dark Matter searches *J. Cosmol. Astropart. Phys.* **2017** 003
[2] Aalseth C E *et al* 2018 DarkSide-20k: a 20 tonne two-phase LAr TPC for direct dark matter detection at LNGS *Eur. Phys. J. Plus* **133** 131
[3] Benetti P *et al* 2008 First results from a dark matter search with liquid Argon at 87 K in the Gran Sasso underground laboratory *Astropart. Phys.* **28** 495
[4] Abi B *et al* 2020 First results on ProtoDUNE-SP liquid argon time projection chamber performance from a beam test at the CERN Neutrino Platform *J. Instrum.* **15** 12004
[5] Bettini A *et al* 1991 The ICARUS liquid argon TPC: a complete imaging device for particle physics *Nucl. Instrum. Methods Phys. Res.* A **315** 223
[6] Acciarri R *et al* 2017 Design and construction of the MicroBooNE detector *J. Instrum.* **12** P02017
[7] Agostinelli S *et al* 2003 GEANT4—a simulation toolkit *Nucl. Instrum. Methods Phys. Res.* A **506** 250
[8] Alonso-Monsalve S and Whitehead L H 2020 Image-based model parameter optimization using model-assisted generative adversarial networks *IEEE Trans. Neural Netw.* **31** 5645
[9] Gatys L A, Ecker A S and Bethge M 2016 Image style transfer using convolutional neural networks *Proc. of the Conf. on Computer Vision and Pattern Recognition* pp 2414–23 (available at: https://openaccess.thecvf.com/content_cvpr_2016/papers/Gatys_Image_Style_Transfer_CVPR_2016_paper.pdf)
[10] Kimura N and Rekimoto J 2018 ExtVision: augmentation of visual experiences with generation of context images for a peripheral vision using deep neural network *Proc. of the 2018 Conf. on Human Factors in Computing Systems* pp 1–10
[11] Buehler M J 2020 Liquified protein vibrations, classification and cross-paradigm de novo image generation using deep neural networks *Nano Futures* **4** 035004
[12] Abi B *et al* 2017 The single-phase ProtoDUNE technical design report (arXiv:1706.07081)
[13] D. collaboration *et al* 2021 Design, construction and operation of the ProtoDUNE-SP liquid argon TPC (arXiv:2108.01902)
[14] Abi B, Acciarri R, Acero M A, Adamov G, Adams D, Adinolfi M, Ahmad Z, Ahmed J, Alion T, Monsalve S A *et al* 2020 Deep Underground Neutrino Experiment (DUNE), far detector technical design report, volume II DUNE physics (arXiv:2002.03005)
[15] Machado A and Segreto E 2016 ARAPUCA a new device for liquid argon scintillation light detection *J. Instrum.* **11** C02004
[16] Keras: (Available at: https://keras.io/)
[17] TensorFlow: (Available at: www.tensorflow.org/)
[18] Code: (Available at: https://github.com/Healthborn/genn)
[19] Goodfellow I J, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, and Bengio Y 2014 Generative adversarial networks (arXiv:1406.2661)
[20] Mirza M and Osindero S2014 Conditional generative adversarial nets (arXiv:1411.1784)

[21] Daskalakis C, Goldberg P W and Papadimitriou C H 2009 The complexity of computing a Nash equilibrium *SIAM J. Comput.* **39** 195

[22] Santini S and Jain R 1999 Similarity measures *IEEE Trans. Pattern Anal. Mach. Intell.* **21** 871

[23] Goldberger J *et al* 2003 An efficient image similarity measure based on approximations of KL-Divergence between two Gaussian mixtures *ICCV* vol 3 pp 487–93 (available at: http://lear.inrialpes.fr/people/triggs/events/iccv03/cdrom/iccv03/0487_goldberger.pdf)

[24] Gretton A, Borgwardt K M, Rasch M J, Schölkopf B and Smola A 2012 A kernel two-sample test *J. Mach. Learn. Res.* **13** 723 (available at: www.jmlr.org/papers/volume13/gretton12a/gretton12a.pdf?ref=https://githubhelp.com)

[25] Huszár F 2015 How (not) to train your generative model: scheduled sampling, likelihood, adversary? (arXiv:1511.05101)

[26] Snider E and Petrillo G 2017 LArSoft: toolkit for simulation, reconstruction and analysis of liquid argon TPC neutrino detectors *J. Phys.: Conf. Ser.* **898** 042057

[27] Kingma D P and Ba J 2014 Adam: a method for stochastic optimization (arXiv:1412.6980)

[28] Wilson Cluster (available at: https://computing.fnal.gov/wilsoncluster/)

[29] Zaheer M, Li C-L, Póczos B and Salakhutdinov R 2018 GAN connoisseur: can GANs learn simple 1D parametric distributions (available at: https://chunliangli.github.io/docs/dltp17gan.pdf)