Generative models for simulation of KamLAND-Zen

Eur. Phys. J. C (2024) 84:651
https://doi.org/10.1140/epjc/s10052-024-12980-7

THE EUROPEAN PHYSICAL JOURNAL C

Regular Article - Experimental Physics

Generative models for simulation of KamLAND-Zen

Zhenghao Fu[1,a], **Christopher Grant**[2], **Dominika M. Krawiec**[3], **Aobo Li**[4,5,b], **Lindley A. Winslow**[1]

[1] Laboratory of Nuclear Science, Massachusetts Institute of Technology, 77 Massachusetts Ave, Cambridge, MA 02139, USA
[2] Department of Physics, Boston University, 590 Commonwealth Ave, Boston, MA 02215, USA
[3] Department of Physics, University of Warwick, Coventry CV4 7AL, UK
[4] Halıcıoğlu Data Science Institute, University of California San Diego, 9500 Gilman Dr, La Jolla, CA 92093, USA
[5] Department of Physics, University of California San Diego, 9500 Gilman Dr, La Jolla, CA 92093, USA

Received: 2 January 2024 / Accepted: 3 June 2024

Abstract The next generation of searches for neutrinoless double beta decay ($0\nu\beta\beta$) are poised to answer deep questions on the nature of neutrinos and the source of the Universe's matter–antimatter asymmetry. They will be looking for event rates of less than one event per ton of instrumented isotope per year. To claim discovery, accurate and efficient simulations of detector events that mimic $0\nu\beta\beta$ is critical. Traditional Monte Carlo (MC) simulations can be supplemented by machine-learning-based generative models. This work describes the performance of generative models that we designed for monolithic liquid scintillator detectors like KamLAND to produce accurate simulation data without a predefined physics model. We present their current ability to recover low-level features and perform interpolation. In the future, the results of these generative models can be used to improve event classification and background rejection by providing high-quality abundant generated data.

1 Introduction

Event simulation is critical to modern particle and nuclear physics and is used in all experimental stages from detector design to the extraction of the final result with the corresponding statistical significance. Traditionally, the simulation of particle detectors starts by modelling the microphysics of the particle depositing energy in the detector, and using Monte Carlo, techniques propagates the signal through the detector geometry. However, the stochasticity and complexity of these processes makes it difficult to reproduce the detector response while simultaneously being computationally expensive to produce datasets of sufficient size.

Rare event searches are a class of experiments that use highly specialized detectors to search for new processes that would indicate new physics at an energy-scale beyond the reach of any modern particle accelerators. Monolithic kiloton-scale liquid scintillator detectors, like KamLAND-Zen, are an excellent detector technology for rare event searches as they provide economical scaling to large volumes. For this reason, they have been the work horse of neutrino physics for many decades [1–9]. KamLAND is a spherical detector, which is composed of 1 kiloton of liquid scintillator (LS) contained in a 13-m-diameter balloon. The LS-filled balloon is surrounded by mineral oil (acting as a buffer volume) and is viewed by 1879 photomultiplier tubes (PMTs). A smaller secondary balloon is currently deployed at the center of the main balloon and contains LS doped with 742 kg of ^{136}Xe (XeLS) to search for $0\nu\beta\beta$[10]. An observation of this rare process (current limits greater than $\sim 10^{26}$ years) would prove that the neutrino is its own antiparticle, also known as a Majorana particle. This is a key ingredient for Leptogenesis [11], which describes the observed matter–antimatter asymmetry in our universe.

In our previous work, we used deep learning methods to classify critical backgrounds [12,13], however the power of deep learning is not limited to background suppression. In this work, we leverage deep learning to tackle event simulation in spherical liquid scintillator detectors with the goal of producing simulations that more accurately produce the detector response while simultaneously reducing the computational burden. While the aspiration is to generate a large number of events from a limited number of training samples, achieving so-called few-shot learning, it is crucial to validate the accuracy of the simulation with extensive data or by initially training with a large dataset and then fine-tuning with a smaller set.

[a] e-mail: fuzh@mit.edu (corresponding author)
[b] e-mail: liaobo77@ucsd.edu (corresponding author)

Published online: 27 June 2024

This work benchmarks two models using simulation data: the main variational auto encoder (VAE) model and an alternative generative adversarial network (GAN) model. Thanks to the data-driven nature of deep learning algorithms, the generalization to real detector data should be very straightforward. This paper is structured as follows. We first introduce the simulation of the liquid scintillator detector data used in this study in Sect. 2. The robustness of the generative model is demonstrated by learning from these different datasets. Section 3 explains the structure of the VAE model and the GAN model. The backbone network PointNet is highlighted in this section, which elucidates the underlying mechanism of information extraction in our generative model. Section 4 outlines the application of the generative models and demonstrates the training results. Finally, Sect. 5 compare the generated events to real detector events using KamLAND-Zen reconstruction algorithm. We evaluate the model by comparing the statistical properties of the original and generated datasets.

2 Detector data

In this work, we train and evaluate our generative model on both simulations and real detector data. The simulation dataset is generated by KLG4sim, a detailed KamLAND simulation based on the GEANT4 toolkit. This dataset is the standard KamLAND-Zen 800 detector MC simulation, referred to as Sim-KLZ800. Sim-KLZ800 consists of events coming from ^{136}Xe $0\nu\beta\beta$ with $Q = 2.458$ MeV and ^{214}Bi (β^- decay). All events are uniformly distributed within a 1.98-meter-radius mini-balloon, contained in a 13-meter-diameter balloon filled with liquid scintillator, and surrounded by a 2.5-meter-thick mineral oil buffer volume. Photons generated in each event will propagate through all these layers, reach the outer boundary of the buffer and trigger simulated KamLAND-Zen PMT. Sim-KLZ800 has been carefully tuned to replicate the response of the real detector.

Besides Sim-KLZ800, we have also prepared a physics validation dataset of real KamLAND-Zen events. To procure this dataset, we chose to use the neutron capture because of its high statistics and isotropy in the spherical detector. Our neutron capture events are from anti-neutrino inverse beta decay, determined by the delayed-coincidence tagging method. In this method, we use the criteria that delayed energy $E_d \geq 1.5$ MeV, space correlation $\Delta R < 300$ cm, time correlation $\Delta T < 2.5$ ms. This criteria is also applied to data selection. The energy spectrum of neutron capture simulation is demonstrated in Fig. 1. Visible energy distribution from neutron capture has its peak at around 2.225 MeV. The theoretical energy spectrum of neutron capture is also shown in the plot for a direct comparison.

Each simulated event results in a collection of triggered PMTs. When a PMT fires, its position, arrival time (hit time t) and registered photoelectron charge (hit charge q) are recorded as a *point* in the 5D space defined by the vector $[x, y, z, t, q]$. In machine learning language, a collection of points is called a *point cloud*. Point cloud data has two main characteristics:

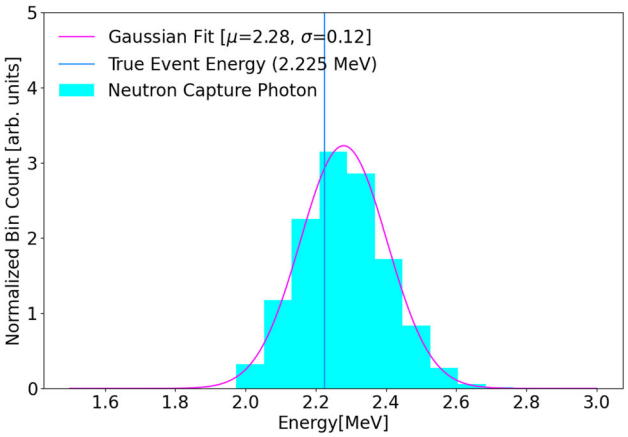

Fig. 1 Visible energy distribution of a subset of neutron capture events. The red line indicates the best-fit curve. And the blue line indicates the theoretical energy line at 2.225 MeV

- Disorder: point-cloud is insensitive to the order of points within.
- Invariance: point-cloud data is invariant to spatial transformations in Poincaré group.

Therefore, applying translations and rotations to the point cloud will not affect the training result. To generate the point clouds for training purpose, some additional corrections are needed. For the time dimension, two corrections are applied to calculate the proper hit time t from this raw hit time. T_{raw} is the raw hit time when the optical photon arrives at the PMT surface. To produce the proper hit time, the following correction is executed upon T_{raw}:

$$t = T_{raw} - TOF - T_0, \tag{1}$$

where TOF is the photon time-of-flight from the event vertex to PMT position and T_0 is the proper start time of the event. By subtracting TOF from T_{raw}, we effectively move the vertex of each event to the center of the detector. By subtracting T_0, we correct for intra-event distortion of the scintillation time profile by the vertex position. The calculation of T_0 is a fractional charge weighted sum of the differences between T_{raw} and TOF over all the PMTs. This is calculated as follows:

$$T_0 = \frac{\sum_i (T^i_{raw} - TOF^i) \times q_i}{\sum_i q_i} \tag{2}$$

where q_i is the hit charge on the i-th PMT. We use the criteria that restrict the hit time to the event within ± 30 ns window,

and any hit times recorded outside this interval will not be considered.

Hit charge is pivotal for reproducing the energy deposition in liquid scintillator detector. It is obtained by first integrating the area under the PMT pulse, which is proportional to the number of optical photons registered at the PMT. The raw integrated value is then normalized by the the so-called 1 photoelectron (p.e.) peak integration. The 1 p.e. refers to the pulse profile where exactly one photoelectron is produced within the PMT. The calculation is displayed below:

$$q = \frac{\int f_{\text{event}}(\tau)d\tau}{\int f_{\text{1p.e.}}(\tau)d\tau}, \tag{3}$$

where f_{event} and $f_{\text{1p.e.}}$ are the PMT pulse and 1 p.e. pulse, respectively. This normalized value is thus the proper hit charge that reflects the number of p.e. While a low energy event creates only a single p.e., a high energy event can creates more than 1000 p.e. PMTs with charge smaller than 0.3 p.e. are recognized as baseline fluctuations due to noise and are dropped from the point cloud.

Up to now, we have obtained the corrected 5D point cloud dataset. However, each event (point cloud) in this dataset contains a varying number of PMT hits (points). The number of PMT hits, or the NHIT, is a essential attribute of LS detector event since it is proportional to energy. However, neural networks can only produce a fixed number of points and, therefore, cannot handle the event-wise variation of NHIT. An additional trigger-dimension is introduced to resolve this problem. In KamLAND-Zen, there are 1879 PMTs spherically covering the detector surface, thus the maximal possible NHIT is 1879. Therefore, the generative model additionally generates 1879 floating point numbers corresponding to 1879 PMTs in KamLAND-Zen. These floating point numbers are then fed to a sigmoid function to constrain their values between 0 and 1, and subsequently converted to 1879 binaries using the Bernoulli distribution. If the value corresponding to a given PMT reads 1, the PMTs will be considered as triggered; otherwise that PMTs will be considered as untriggered.

The trigger-dimension is then used to transform the point clouds to a concentric double-sphere. If a PMT is marked as triggered by this dimension, its 5D values $[x, y, z, t, q]$ are kept at the original position, while the untriggered PMTs are shifted to the origin in 5D point cloud space, as shown in Fig. 2. As we will discuss in Sect. 3.2, the concentric double-sphere efficiently ignores the untriggered PMTs while allowing the generative model to produce fix-sized outputs.

Lastly, the variations of charge and time distributions are scaled to a comparable size before training. Ideally, a robust model should not be confused by scaling. It should regard time and charge values equally, however the difference in sizes of charge and time can mislead the machine. It may only focus on the dimension with a larger scale and ignore the dimension with a smaller scale. This rescaling method improves the accuracy by giving a default discrepancy in spatial information from different categories. It also speeds up the convergence of the model and prevents the gradient exploding problem (excessively large gradients that leads to unstable convergence) during the training.

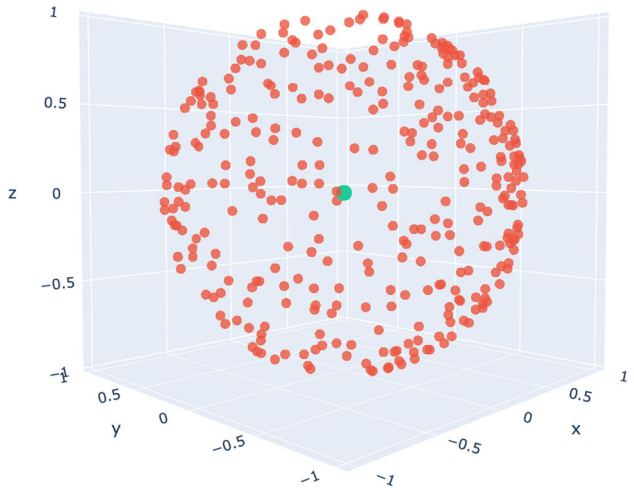

Fig. 2 The inner sphere is for the untriggered PMTs (blue), and the outer sphere is for the triggered PMTs (red). The concentric double-sphere efficiently ignores the untriggered PMTs while allowing the generative model to produce a fix-sized outputs

3 Network design

The power of a generative model emerges from its ability to probe the underlying low-level physics of the KamLAND-Zen events. If we define X is the (simulated) detector events and Y is the type of events. A generative model aims to describe what an observation X should be when Y is given. This process requires a likelihood function $P(X|Y)$ and a probability distribution $P(Y)$, where $P(Y)$ is obtained as prior knowledge, and $P(X|Y)$ is learned by the training of generative model. Two most popular generative models — variational auto-encoder (VAE) and generative adversarial network (GAN) — are used and compared in this study. VAE has an explicit latent space that the inference of distribution $P(X|Y)$ is enforced, while GAN has an implicit latent space and will not solve inference queries [14].

3.1 PointNet

Typical image data such as photos or portraits can be projected on a regular pixel grid with uniform data density. However, the spatial and temporal distribution of liquid scintillator detector data is irregular and uneven and thus cannot

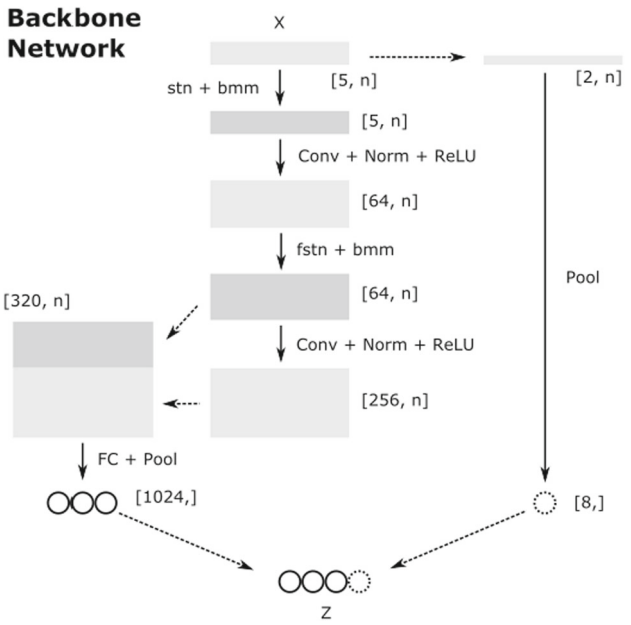

Fig. 3 Major components in both generative models are shown based on the PointNet design. The *stn* and *fstn* are the spatial transform networks [15] used to study global information, and feature information, respectively. Conv stands for Convolutional Layers, FC stands for fully connected layers, and norm stands for batch normalization. ReLU is the Rectified Linear Unit activation function

be efficiently projected onto a 2D pixel grid. Therefore, we treat PMT hits as point clouds and adopt the PointNet model [16] as the backbone network for the generative models, see Fig. 3. PointNet has the ability to extract both global and local features at various scales with a multi-layer structure. In KamLAND-Zen, the coverage of the PMT array is constrained by the number of PMTs. In this case, each point in the point cloud does not represent an exact photon location on the PMT photocathode; rather, it marks the PMT photocathode area where a hit could occur.

3.2 Variational auto-encoder (VAE)

An autoencoder (or self-encoder) is an unsupervised learning model. A conventional autoencoder encodes data into a low-dimensional latent space representation (or vector). This model contains a encoder network and a decoder network. The encoder network encodes the input data into a low-dimensional representation containing key information of the data. The representation is then fed into the decoder part to create an output data with the same dimensionality. During training, a reconstruction loss function is defined to minimize the differences between the output and the input data. This guarantees that important features from data can be encoded into the latent space, which can be used to reconstruct the input. Conventional autoencoders do not place any constraint upon the latent space. This model can reconstruct known events from encoded latent space vector, but lacks the ability to generate new events.

Conventional autoencoders provide a foundational framework, which variational autoencoders (VAEs) build upon to introduce event generation capabilities. VAEs regularize the latent space representation to follow a multivariate normal distribution. They enhance the basic autoencoder model by including additional terms in the autoencoder loss function:

$$L = L_0(x, \hat{x}) + \beta D_{KL}(q(z|x) \,||\, p(z)),$$

where $L_0(x, \hat{x})$ is the reconstruction loss in the conventional auto-encoder. D_{KL} denotes the Kullback–Leibler divergence [17] computed between the returned distribution $q(z|x)$ of the latent vector $\mathbf{z}$ and the desired distribution $p(z)$ [18]. Instead of directly extracting $\mathbf{z}$ from the encoder network, the VAE produces two vectors with the same dimension as z, namely $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$. The $\mathbf{z}$ of the VAE is then produced using the following equation:

$$z = \boldsymbol{\sigma} \cdot \boldsymbol{\omega} + \boldsymbol{\mu}$$

where $\boldsymbol{\omega}$ is sampled from a multivariate normal distribution with the same dimension as μ and σ. This method is referred to as a "reparameterization trick". It allows gradient to flow through the network. After training, we can repetitively sample from the multivariate normal distribution to simulate new events. Lastly, β is the hyperparameter that controls the strength of the latent space regularization.

The structure of the customized VAE for KamLAND-Zen data is displayed in Fig. 4. The encoding part is the PointNet model we introduced in previous section, and the decoding part is a fully-connected neural network. In this work, the reconstruction loss L_0 contains two parts. The first part is the Chamfer distance [20] calculated between the input and output data. Chamfer distance is defined as the sum of the minimal distance between each pair of points separately from two point clouds. As discussed in Sect. 2, the untriggered PMTs will be at the origin after the concentric double-sphere transformation. When calculating the Chamfer distance, contributions from the untriggered PMTs will be 0 thus does not contribute to the network training. Therefore, we are effectively training on triggered PMTs when minimizing the Chamfer distance. Furthermore, we use the Binary Cross-Entropy (BCE) loss to regularize the trigger-dimension. This loss effectively limits the total number of triggered PMTs in the output data, compensating for the Chamfer loss's neglect of the inconsistent number of hits in the output and input.

3.3 Generative adversarial network (GAN)

The GAN takes a different approach to generate events. The GAN consists of two networks competing against each other: the discriminator $D(x)$ is designed to determine the authenticity of data x, and the generator G converts the randomly-

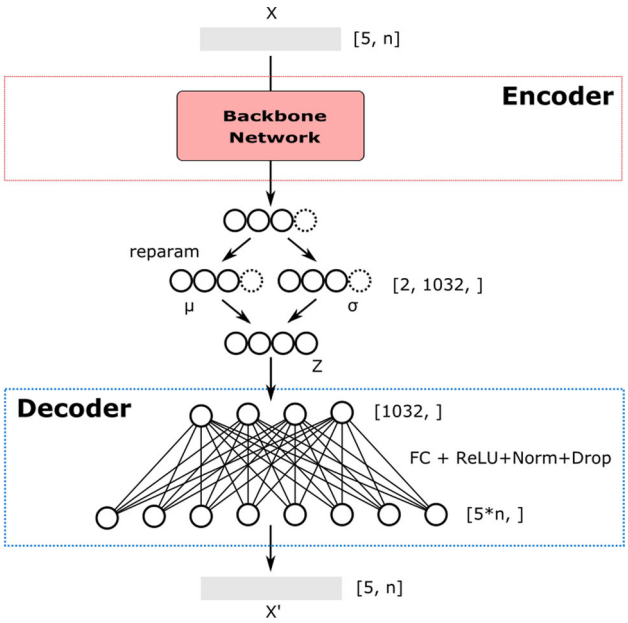

Fig. 4 Schematic diagram of our generative model based on the VAE architecture. The encoder transforms the original input X to a latent vector Z, and the decoder recovers the input X' from this latent vector. FC stands for fully connected layers, norm stands for batch normalization, and Drop stands for Dropout Layers [19]. ReLU is the Rectified Linear Unit activation function

sampled noise vector z to a synthetic event $G(z)$. In a well-trained GAN model, the discriminator and generator will reach a Nash equilibrium thereby leading to efficient event generation from random noise z.

The Wasserstein-GAN model [21] is adopted to avoid the divergence in losses of generators and discriminators. Wasserstein-distance measures the earth-moving distance between the distributions of real and fake events, which provides a valid gradient for the GAN model to train. A gradient penalty [22] is also implemented to facilitate the training, which gives a bound on the Lipschitz-norm of the gradient of the discriminator function $D(x)$ and limits the discriminator from making dramatic changes when the input sample only varies slightly.

In this work, we developed the GAN to fulfill the same event generation task of the VAE. However, the reconstruction power of GAN turns out to be worse than the VAE model in Sect. 3.2, as shown in Table 1. The design of GAN is shown in the Fig. 5.

4 Training results

We used several datasets to train and validate the generative models. The performance of conventional autoencoder, VAE and GAN models are shown for Sim-KLZ800 datasets. We also demonstrate VAE's capability to conduct few-shot learning with merely 50 training samples.

4.1 Standard simulation of $0\nu\beta\beta$ decay

To test the performance of generative models on sim-Full, we train VAE and GAN to simultaneously generate the trigger-dimension, time and charge for every PMT. The generation performance is evaluated with Intersection over Union (IoU) metric, also named as *Jaccard index*, between the original (P_0) and the generative distributions (P_g). The IoU ratio is calculated using the binned distributions of the input and output data as:

$$J(P_0, P_g) = \frac{\sum_{k=1}^{n} min(P_0[k], P_g[k])}{\sum_{k=1}^{n} max(P_0[k], P_g[k])}, \tag{4}$$

where k is the index of histogram bins of both the original and generated data. The ratio J ranges from 0 to 1. Two identical datasets will have $J = 1$, and any difference in distribution will result in a decrease in IoU. Meanwhile, we also included normalized Wasserstein distance to evaluate the generation accuracy, expressed as:

$$\mathscr{R}(P_0, P_g) = \frac{W(P_0, P_g)}{\max(D_0) - \min(D_0)}, \tag{5}$$

where D_0 is the input data. Two identical datasets will also have $\mathscr{R} = 0$, and any difference in distributions will increase this value.

The training result is illustrated in Fig. 6, where the distributions of hit times and hit charges of 1000 events are compared. The hit time has $J_t = 89.0 \pm 0.1\%$, and the charge has $J_q = 91.9 \pm 0.1\%$. This suggests a strong correlation between the real and generated events, with the observed discrepancies likely stemming from the subtle yet intrinsic variability

Table 1 Performance of the VAE and the GAN model. The comparison is performed with three metrics: intersection over Union (IoU), normalized Wasserstein distance (WD), and p-value from KS test. The GAN is trained with 10,000 epochs

Metrics	VAE IoU [%]	VAE WD [a.u.]	p-value [a.u.]	GAN IoU [%]	GAN WD [a.u.]
Time dist.	89.0	9.32×10^{-2}	0.997	85.5	0.241
Charge dist.	91.9	0.254	0.861	80.0	0.461
Nhit dist.	81.2	1.96	0.963	76.2	2.98
Tot charge dist.	76.0	5.57	0.861	–	–

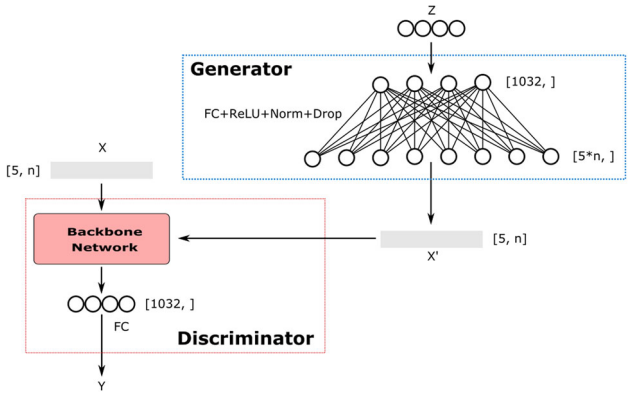

Fig. 5 Schematic diagram of our generative model based on the GAN architecture. Original input X and generated input X' from random noise Z are compared and classified. FC stands for fully connected layers, norm stands for batch normalization, and Drop stands for Dropout Layers [19]. ReLU is the Rectified Linear Unit activation function

and noise in our data that was not fully captured by our model. The NHIT and total charge of original events and generated events are compared in Fig. 7. The NHIT distribution has $J_N = 89.2 \pm 2.0\%$, and the total charge distribution has $J_Q = 76.0 \pm 2.0\%$. We conducted a Kolmogorov-Smirnov (KS) test on the four previously mentioned distributions to assess their goodness of fit. In every instance, the p-value was found to be greater than 85%, indicating good agreements. We also tested the GAN models on the same dataset, and the training result is shown in Table 1. The GAN model gives decent performance, but it underperforms when compared to the VAE on the four distributions under both evaluation metrics.

4.2 Few-shot learning

One essential advantage of generative model is its few-shot learning capability, in that it learn crucial features with an extremely small number of training events, denoted as *few-shot dataset*. To perform few-shot learning, we first pre-train our generative model using a large ^{214}Bi dataset from sim-KLZ800. Pre-training allows the generative model to

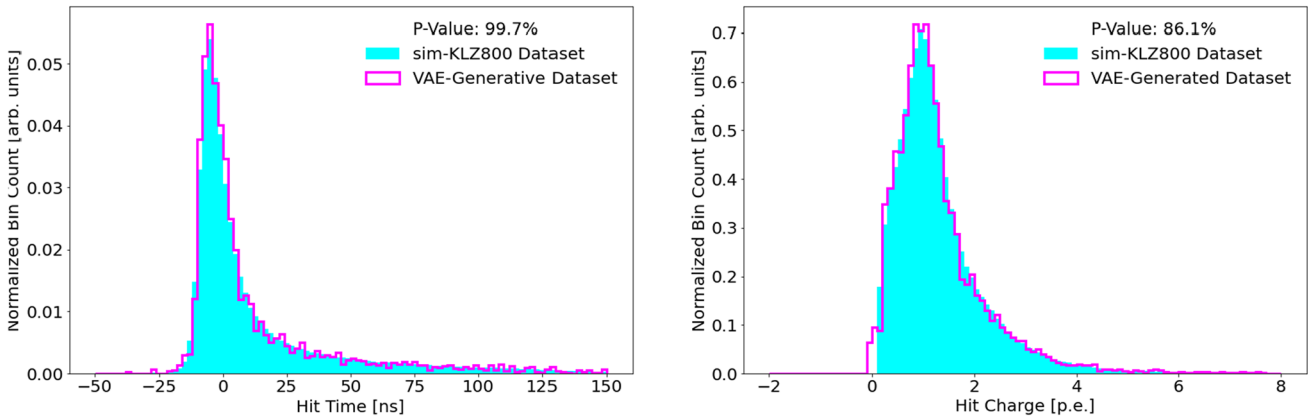

Fig. 6 The averaging time distribution of all triggered hit points is shown on the left, and the averaging charge distribution of all triggered hit points is shown on the right

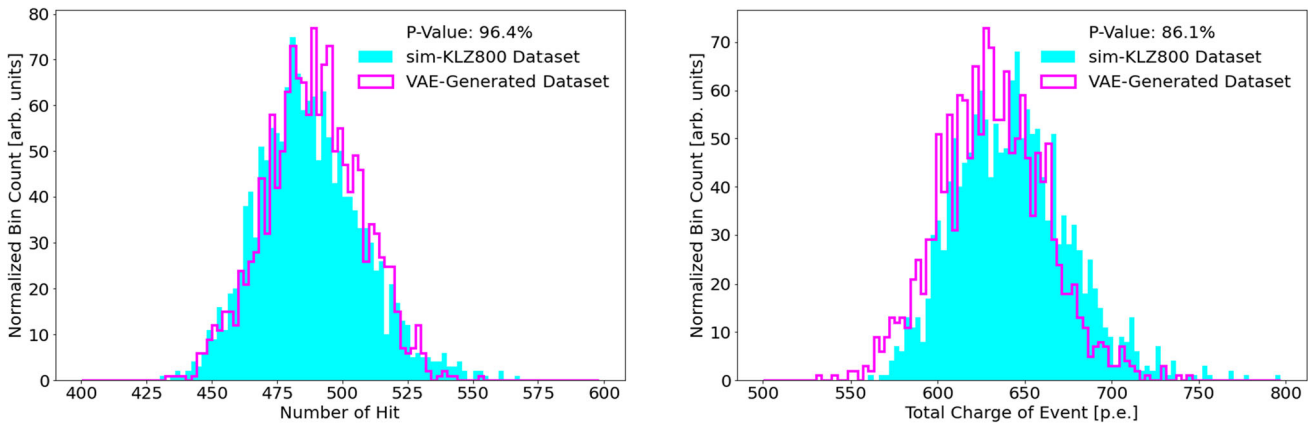

Fig. 7 The NHIT distribution for 1000 ^{136}Xe $0\nu\beta\beta$ events is shown on the left and the total charge distribution for the same events is shown on the right. All events are generated by the VAE model

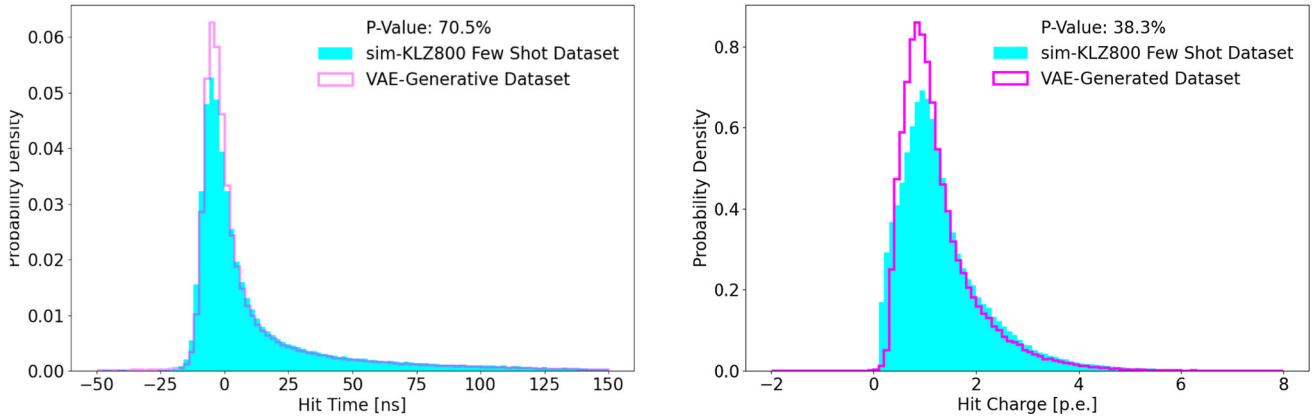

Fig. 8 Few-shot training result for the time and charge distributions

learn basic features of liquid scintillator events and provide a good starting point for training. We selected ^{214}Bi for pre-training because a pure set of ^{214}Bi can be easily selected by delayed coincidence tagging in KamLAND-Zen. Next, the pre-trained model is trained with a relatively small collection of 50 ^{136}Xe $0\nu\beta\beta$ events forming the few-shot dataset. The trained model is used to generate 1000 ^{136}Xe $0\nu\beta\beta$ events and compared to 1000 real ^{136}Xe $0\nu\beta\beta$ events.

The result of the few-shot training is shown in Fig. 8. With an extremely small number of few-shot samples, few-shot learning reproduces the statistical distribution of individual events where the associated uncertainties for IoU values are obtained through bootstrapping. the IoU values for time distribution $J_t = 83.59 \pm 0.03\%$ and charge distribution $J_q = 83.09 \pm 0.03\%$ are approximately equal to normal learning, while the network without few-shot training gives gives $J_t = 2.65 \pm 0.05\%$ and $J_q = 5.04 \pm 0.06\%$. We further assessed our results using the same KS test for goodness-of-fit as detailed in Sect. 4.1, with the p-values presented in Fig. 8. When comparing few-shot learning to high-statistics training, the p-value for the hit-time distribution decreased from 99.7 to 70.5%, and for the hit-charge distribution, it dropped from 86.1 to 38.3%. These findings highlight the reduced efficiency of few-shot learning, especially in simulating hit charges which need to be improved in future studies.

This outcome suggests that methods combining pre-training and fine-tuning could serve as a possible way for improving model performance in scenarios with limited training data, although the ability to simulate PMT hit charge needs to be improved. In KamLAND-Zen 800, we found an increase in the background rate at the inner balloon bottom, possibly due to the settling of dust particles. However, an unambiguous identification of the source is impossible due to limited statistics [10]. In this case, the few-shot data are the unknown background events with limited statistics. Leveraging few-shot learning, we will be able to boost the population of few-shot data using the following steps: we first use tagged ^{214}Bi data to pre-train the VAE, then use the collected few-shot data to train the network. "After training, the generative model will be able to generate as many events as needed. Those generated events can provide hints of the origin of low-statistics background events in real data. Note that the training procedure is based on detector data, therefore the generated events will accurately include effects coming from the real detector response.

5 Physics validation

To enhance our understanding of generative models' capabilities, we further conducted a physics validation study by reconstructing key physics parameters in real detector datasets. Within the context of liquid scintillator detectors, the reconstructed energy and position stand out as two key quantities of each physics event. In the KamLAND-Zen experiment, these quantities are derived using a three-stage hierarchical reconstruction algorithm, as detailed in [23]. The first stage involves the LT Vertex fitter, providing a fast yet somewhat less precise estimation of event position. Subsequently, the output from the LT fitter feeds into the second stage, the V2 Vertex fitter, which yields a more accurate estimation of the event position. Finally, the output of the V2 vertex fitter acts as input for the energy fitter to reconstruct event energy.

For this physics validation study, we leveraged the neutron capture (NC) dataset as described in Sect. 2. Throughout the KamLAND-Zen 800, a total of 3760 neutron capture events were tagged and collected. The three aforementioned reconstruction algorithm was used to reconstruct their energy E and positions $\{x, y, z\}$. We first independently implemented the reconstruction algorithm of LT position fitter, but implementing the V2 and Energy fitter requires access to both the KamLAND-Zen source code and its calibration data, which are not available within the scope of this work. To

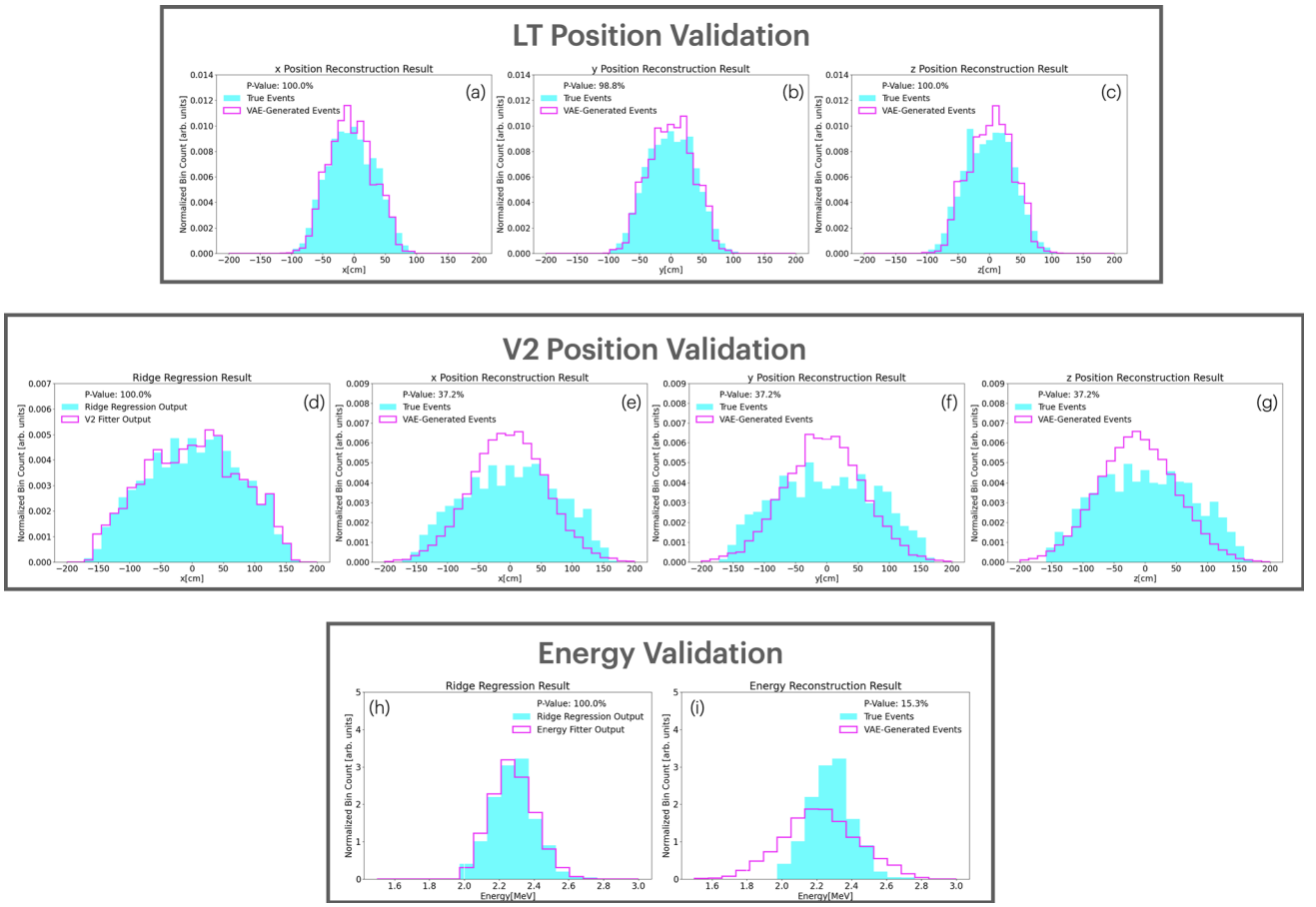

Fig. 9 Summary of the physics validation study results. The first row of histograms (**a**–**c**) show the LT Vertex fitter output between true neutron capture events and VAE-Generated events. The second row of histograms (**d**–**g**) show the V2 Vertex fitter output between true neutron capture events and VAE-Generated events. The response of V2 Vertex fitter is reproduced by a Ridge Regression algorithm as shown in (**d**). The third row, containing histograms (**h**,**i**) show the Energy fitter output between true neutron capture events and VAE-Generated events. The response of Energy fitter is reproduced by a Ridge Regression algorithm as shown in (**h**)

mimic the effect of V2 Position and Energy fitters we subsequently divided the NC dataset into two subsets at random: NC Subset A, comprising 1000 events, and NC Subset B, comprising 2760 events. We employed a Ridge regression algorithm trained on NC Subset A to emulate the V2 Position and Energy fitter algorithms, aiming to replicate its effect in reconstructing energy and positions. The efficiency of the Ridge Regression algorithm in mimicking the KamLAND-Zen Reconstruction Algorithm is depicted in Fig. 9h, demonstrating decent agreement between the Energy fitter and Ridge Regression. A similar level of agreement is observed for V2 Position reconstruction, as illustrated in Fig. 9d.

Subsequently, we applied the Ridge Regression algorithm to both NC Subset B and 10,000 neutron capture events generated by the PointNet-VAE algorithm, aiming to compare the distribution of reconstructed energy and the reconstructed x, y, and z positions between these two datasets. The result of these comparison is illustrated in Fig. 9 and Table 2. In terms of energy reconstruction, the energy distribution of actual events from NC Subset B was found to be 2.279 ± 0.123 MeV, while the VAE-generated neutron capture events exhibited an energy distribution of 2.234 ± 0.335 MeV. This comparative analysis of the energy distributions, between real events and those generated by the VAE, is visually presented in Fig. 9i. The data generated by the VAE are closer to the true neutron capture energy at 2.225 MeV. However, the VAE-reconstructed events exhibits a worse standard deviation, indicating an area for potential improvement in generative model's performance.

In terms of position reconstruction, we assessed the x, y, and z coordinates using both the LT Position and V2 Position fitters. The outcomes of both fitters were documented in Table 2, along with the goodness of fit determined via the KS test. As the detected neutron capture events are anticipated to be uniformly distributed within the designated region of

Table 2 Physics validation result of three KamLAND-Zen reconstruction algorithms: LT Vertex, V2 Vertex, and Energy fitters. Each fitter is applied over two datasets: the NC Subset B containing tagged 2.2 MeV neutron capture events, and the VAE-Generated dataset. All reconstructed quantities are reported with their mean value ± their standard deviations over the entire dataset. A goodness of fit metric is calculated by KS test

Reconstruction algorithm	x [cm]	p-value [%]	y [cm]	p-value [%]	z [cm]	p-value [%]	E [MeV]	p-value [%]
LT [NC subset B]	−3.4±36.8	100	0.8±36.9	98.8	0.3±37.1	100	–	–
LT [VAE-generated]	−4.4±34.1		0.5±34.6		2.7±34.1		–	
V2 [NC subset B]	0.7±74.6	37.2	0.3±74.6	37.2	2.7±75.5	37.2	–	–
V2 [VAE-generated]	−0.4±56.1		−4.3±62.1		−13.0±63.4		–	
Energy [NC subset B]	–	–	–	–	–	–	2.279±0.123	15.3
Energy [VAE-generated]	–		–		–		2.234±0.335	

interest ($r < 157$ cm), the reported x, y, and z position distributions in Table 2 has a large standard deviations. These comparisons are also visually depicted in Fig. 9e–g, wherein it becomes evident that the VAE-generated data tends towards a more Gaussian-like distribution. This tendency likely stems from the intrinsic nature of the VAE, whereby event generation relies on random sampling from a multivariate Gaussian distribution. An excellent level of agreement is achieved with the LT Position fitter results, while a comparatively poorer agreement is observed with the V2 Position fitter results, suggesting an area for potential enhancement. Nevertheless, considering that the vertex resolution of KamLAND-Zen is approximately 17 cm, the discrepancies between the NC Subset B and VAE-generated events for both LT and V2 fitters fall within this vertex resolution. This implies that the VAE has the capability to produce data that closely resembles KamLAND-Zen events within the current detector resolution.

6 Conclusion and outlook

Traditional simulation is based on inferences from first principles, but it depends on the accuracy of the input parameters and often fails to exactly reproduce the detector microphysics. In this work, we developed two generative models to simulate liquid scintillator detector data. The generative models improve the efficiency of generating data with reasonable reconstruction accuracy. With a standard detector configuration similar to the current KamLAND-Zen detector, the variational autoencoder model can accurately simulate data with $J \gtrsim 90\%$ and $\mathcal{R} \lesssim 5\%$. Furthermore, we examine the possibility of few-shot learning using the given generative model. With fewer than 50 training events and an easy-to-collect pre-training dataset, our generative model is one possible way to significantly boost event population when obtaining high-statistics data is impossible. Lastly, we performed a physics validation study to demonstrate that the events generated by our model aligns with the real detector events when processed through the KamLAND-Zen reconstruction algorithm.

This work's focus is the optimization of the algorithms to study the statistical properties of raw data and generate like-real detector events. However, during the course of our physics validation study, we have identified certain disparities between the reconstructed position and energy of our generated events and those observed in real KamLAND-Zen events. Notably, the energy resolution of generated neutron capture events appears to be compromised compared to real events. To address these issues and further enhance the fidelity of our model, we plan to incorporate additional machine learning technique techniques, potentially incorporating the Ridge Regression algorithm into the training process to "guide" the event generation. This approach could steer our generative model in the direction of producing events with more accurately reconstructed energies and positions. Generative modeling techniques, as cited in LHC [24], LArTPC Neutrino Experiments [25], as well as other $0\nu\beta\beta$ experiments [26]. These models will benefit particle and nuclear physics by offering a faster and data-driven method for simulation development, and our findings highlight the potential of these techniques in $0\nu\beta\beta$ experiments.

Acknowledgements This material is based upon work supported by the National Science Foundation under Grant Numbers 2110720, 2012964. This work is done in support of the KamLAND-Zen experiment and we thank our collaborators for their input. We thank Alexander Leder and Daniel Mayer for their review of the manuscript. The KamLAND-Zen experiment is supported by JSPS KAKENHI Grant Numbers 21000001, 26104002, and 19H05803; the Dutch Research Council (NWO); and under the U.S. Department of Energy (DOE) Grant No. DE-AC02-05CH11231, as well as other DOE and NSF grants to individual institutions. This research was performed, in part, using the Boston University Shared Computing Cluster.

Data Availability Statement This manuscript has no associated data or the data will not be deposited. [Authors' comment: The data can be made available upon reasonable email request.]

Code Availability Statement This manuscript has no associated code/software. [Authors' comment: The code/software can be made available upon reasonable email request.]

Funded by SCOAP[3].

References

1. A. Gando et al., Phys. Rev. Lett. **117**(8), 082503 (2016). https://doi.org/10.1103/PhysRevLett.117.082503. [Addendum: Phys. Rev. Lett. 117, 109903 (2016)]
2. K. Eguchi et al., Phys. Rev. Lett. **90**, 021802 (2003). https://doi.org/10.1103/PhysRevLett.90.021802
3. M. Agostini et al., Nature **587**, 577 (2020). https://doi.org/10.1038/s41586-020-2934-0
4. C. Arpesella, H.O. Back, M. Balata, G. Bellini, J. Benziger, S. Bonetti, A. Brigatti, B. Caccianiga, L. Cadonati, F. Calaprice, C. Carraro, G. Cecchet, A. Chavarria, M. Chen, F. Dalnoki-Veress, D. D'Angelo, A. de Bari, A. de Bellefon, H. de Kerret, A. Derbin, M. Deutsch, A. di Credico, G. di Pietro, R. Eisenstein, F. Elisei, A. Etenko, R. Fernholz, K. Fomenko, R. Ford, D. Franco, B. Freudiger, C. Galbiati, F. Gatti, S. Gazzana, M. Giammarchi, D. Giugni, M. Goeger-Neff, T. Goldbrunner, A. Goretti, C. Grieb, C. Hagner, W. Hampel, E. Harding, S. Hardy, F.X. Hartman, T. Hertrich, G. Heusser, A. Ianni, A. Ianni, M. Joyce, J. Kiko, T. Kirsten, V. Kobychev, G. Korga, G. Korschinek, D. Kryn, V. Lagomarsino, P. Lamarche, M. Laubenstein, C. Lendvai, M. Leung, T. Lewke, E. Litvinovich, B. Loer, P. Lombardi, L. Ludhova, I. Machulin, S. Malvezzi, S. Manecki, J. Maneira, W. Maneschg, I. Manno, D. Manuzio, G. Manuzio, A. Martemianov, F. Masetti, U. Mazzucato, K. McCarty, D. McKinsey, Q. Meindl, E. Meroni, L. Miramonti, M. Misiaszek, D. Montanari, M.E. Monzani, V. Muratova, P. Musico, H. Neder, A. Nelson, L. Niedermeier, L. Oberauer, M. Obolensky, M. Orsini, F. Ortica, M. Pallavicini, L. Papp, S. Parmeggiano, L. Perasso, A. Pocar, R.S. Raghavan, G. Ranucci, W. Rau, A. Razeto, E. Resconi, P. Risso, A. Romani, D. Rountree, A. Sabelnikov, R. Saldanha, C. Salvo, D. Schimizzi, S. Schönert, T. Shutt, H. Simgen, M. Skorokhvatov, O. Smirnov, A. Sonnenschein, A. Sotnikov, S. Sukhotin, Y. Suvorov, R. Tartaglia, G. Testera, D. Vignaud, S. Vitale, R.B. Vogelaar, F. von Feilitzsch, R. von Hentig, T. von Hentig, M. Wojcik, M. Wurm, O. Zaimidoroga, S. Zavatarelli, G. Zuzel, Phys. Rev. Lett. **101**, 091302 (2008). https://doi.org/10.1103/PhysRevLett.101.091302
5. G. Bellini et al., Phys. Rev. Lett. **107**, 141302 (2011). https://doi.org/10.1103/PhysRevLett.107.141302
6. M. Anderson et al., Phys. Rev. D **99**(3), 032008 (2019). https://doi.org/10.1103/PhysRevD.99.032008
7. M. Anderson et al., Phys. Rev. D **99**(1), 012012 (2019). https://doi.org/10.1103/PhysRevD.99.012012
8. F.P. An et al., Phys. Rev. Lett. **115**(11), 111802 (2015). https://doi.org/10.1103/PhysRevLett.115.111802
9. F. An et al., J. Phys. G **43**(3), 030401 (2016). https://doi.org/10.1088/0954-3899/43/3/030401
10. S. Abe, et al. First Search for the Majorana Nature of Neutrinos in the Inverted Mass Ordering Region with KamLAND-Zen (2022)
11. M. Fukugita, T. Yanagida, Phys. Lett. B **174**(1), 45 (1986)
12. A. Li, Z. Fu, L. Winslow, C. Grant, H. Song, H. Ozaki, I. Shimizu, A. Takeuchi (2022)
13. A. Li, A. Elagin, S. Fraker, C. Grant, L. Winslow, Nucl. Instrum. Meth. A **947**, 162604 (2019). https://doi.org/10.1016/j.nima.2019.162604
14. S. Mohamed, B. Lakshminarayanan, arXiv e-prints arXiv:1610.03483 (2016)
15. M. Jaderberg, K. Simonyan, A. Zisserman, K. Kavukcuoglu, arXiv e-prints arXiv:1506.02025 (2015)
16. C.R. Qi, H. Su, K. Mo, L.J. Guibas, arXiv e-prints arXiv:1612.00593 (2016)
17. S. Kullback, R.A. Leibler, Ann. Math. Statist. **22**(1), 79 (1951)
18. D.P. Kingma, M. Welling, arXiv e-prints arXiv:1312.6114 (2013)
19. N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, J. Mach. Learn. Res. **15**(1), 1929 (2014)
20. T. Wu, L. Pan, J. Zhang, T. Wang, Z. Liu, D. Lin, arXiv e-prints arXiv:2111.12702 (2021)
21. M. Arjovsky, S. Chintala, L. Bottou, arXiv e-prints arXiv:1701.07875 (2017)
22. I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, A. Courville, arXiv e-prints arXiv:1704.00028 (2017)
23. A. Li, The Tao and Zen of neutrinos: Neutrinoless double beta decay in KamLAND-Zen 800. Ph.D. thesis, Boston U. (2020)
24. M. Paganini, L. de Oliveira, B. Nachman, Phys. Rev. D **97**(1), 014021 (2018). https://doi.org/10.1103/PhysRevD.97.014021
25. P. Lutkus, T. Wongjirad, S. Aeron, In *9th International Conference on Learning Representations* (2022)
26. S. Li et al., JINST **18**(06), P06005 (2023). https://doi.org/10.1088/1748-0221/18/06/P06005