Course Project: Application of Variational Autoencoder in Particle Physics Events Generation

Rui XUE
Department of Physics and Astronomy
University of Pittsburgh
Pittsburgh, PA 15213
rux23@pitt.edu
rxue@andrew.cmu.edu

Abstract

Events generation in particle physics research is a highly time-consuming task. Researchers use event generators, based on Quantum Field Theory (QFT) and Monte-Carlo simulation, to simulate how elementary particles interact with each other. The angle distribution results of those particles can be seen as a high dimensional probability distribution. Variational Autoencoders (VAE) are widely used nowadays to learn probability distributions of datasets. This project aims at exploring the potential use of the VAE method in particle physics events generations.

1 Introduction

The Standard Model (SM), based on Quantum Field Theory (QFT), is the most important theoretical frame work in particle physics. The finding of Higgs Bosons in 2012 by LHC is a great triumph of the Standard Model, and it marks that all the predicted particles are found by human beings[1]. However, the story hasn't come to an end. There are still lots of problems that cannot be solved by the SM, such as the origin of dark matters, the imbalance of matters and anti-matters in the universe, attempts to unify strong force with electro-magnetic force and weak force, attempts to unify QFT with General Relativity. To investigate those problems, we must find particle-interaction signals that cannot explained by SM, and hopefully, those signals could serve as evidence of some Beyond Standard Model (BSM) theories.

The Effective Field Theory (EFT) proposes the real world Lagrangian is in the form of

$$\mathcal{L}_{\text{eff}} = \mathcal{L}_{\text{SM}}^{(4)} + \frac{1}{\Lambda}\sum_i C_i^{(5)}\mathcal{O}_i^{(5)} + \frac{1}{\Lambda^2}\sum_i C_i^{(6)}\mathcal{O}_i^{(6)} + \mathcal{O}\left(\frac{1}{\Lambda^3}\right)$$

And the first term $\mathcal{L}_{\text{SM}}^{(4)}$ is the SM Lagrangian. The rest terms are the BSM terms we are searching for. $C_i^{(5)}$ and $C_i^{(6)}$ are Wilson Coefficients that determines the intensity of BSM interactions. Top quarks, as the heaviest quarks, are expected to show BSM signals with higher probabilities. Using ten angles, we can fully describe a top-antitop decaying system as shown by Fig.1.

The $t\bar{t}$ decaying system can be fully described with $\rho(\theta^+, \phi^+, \theta^-, \phi^-, \theta^{*+}, \phi^{*+}, \theta^{*-}, \phi^{*-}, \theta, \eta)$. This is a 10 dimensional probability distribution, determined by the Lagrangian $\mathcal{L}$. Nowadays, the main bottleneck of physics analysis comes from the simulation side. The complicated theoretical computation takes weeks or even months to generate enough full-simulation datasets. So a fast simulation method is urgently needed for early-stage studies. This project is trying to explore the potential of Variational Encoders (VAE) in particle physics simulation, using the VAE model to learn 10 angle distribution $\rho(\vec{\theta})$, thus doing quick sampling to generate simulation data.

10617 Course Project (2024).

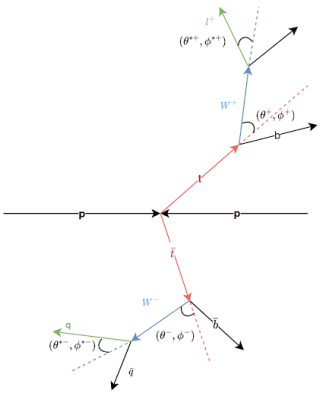


Figure 1: Top-Antitop decay system. 8 angles are marked on the graph, the rest 2 angles describes the whole system orientation.

The generator-level simulation is the first step of full-simulation, and the process is purely based on theoretical calculation and Monte-Carlo sampling. There are two main generators used by physicists — SMEFTSim and DIM6. The training data we used comes from DIM6 simulation, and the total number of events used for training and testing is 5,5000. Just completing the generator simulation costs hours, but 55K events are still too little for real analysis, not to mention millions of events. And the more complicated full-simulation process can take months in the worst case.

Considering the time consumption problems, fast-simulation methods are necessary, and there are some works in this area. However, most of them are trying to speed up the full-simulation process, and there is little attention on speeding up the generator simulation. Besides, the early works on fast-simulation are mainly based on Generative Adversarial Networks (GAN)[3]. GAN is hard to train and tricky in tuning parameters. Since it's more reasonable to learn the particle-angle distribution using probability models, VAE models, based on strict math and easier for tuning and training as well, are better candidates for fast-simulation tasks [4][8]. In this project, we use VAE models for fast-generator simulation, so the topic and the method are brand new in particle physics simulation.

2 Data Preprosessing

As mentioned in the introduction section, each $t\bar{t}$ collision event can be fully denoted with ten-angle variables, meaning we can recover the full physics process using only these ten variables. Using DIM6 simulation, we get

$$cos(\theta^+), \phi^+, cos(\theta^-), \phi^-, cos(\theta^{*+}), \phi^{*+}, cos(\theta^{*-}), \phi^{*-}, \theta, \eta / Y_{rapidity}$$

The variables have different ranges, thus we normalize all these variables to the range of $[0, 1]$ using

$$\frac{x - min(x)}{max(x) - min(x)}$$

Since this is a high-dimension distribution, to get an overview of the data, we plot each variable's marginal distribution, shown by Fig. 2. The marginal distributions are very different, $cos(\theta^{*+}), cos(\theta^{*-}), Y_{rapid}$ are concave distributions and $cos(\theta)$ is a convex distribution. And $cos(\theta^+), \phi^+, cos(\theta^-), \phi^-, \phi^{*+}, \phi^{*-}$ are almost uniform distributions, this is because the $t\bar{t}$ events are symmetric in spin polarization, so on average the angles have no preferences in specific directions. Even though most of the distributions are uniform, the angles are correlated, and we can draw the covariance heatmap between these angles, shown by Fig. 3a.

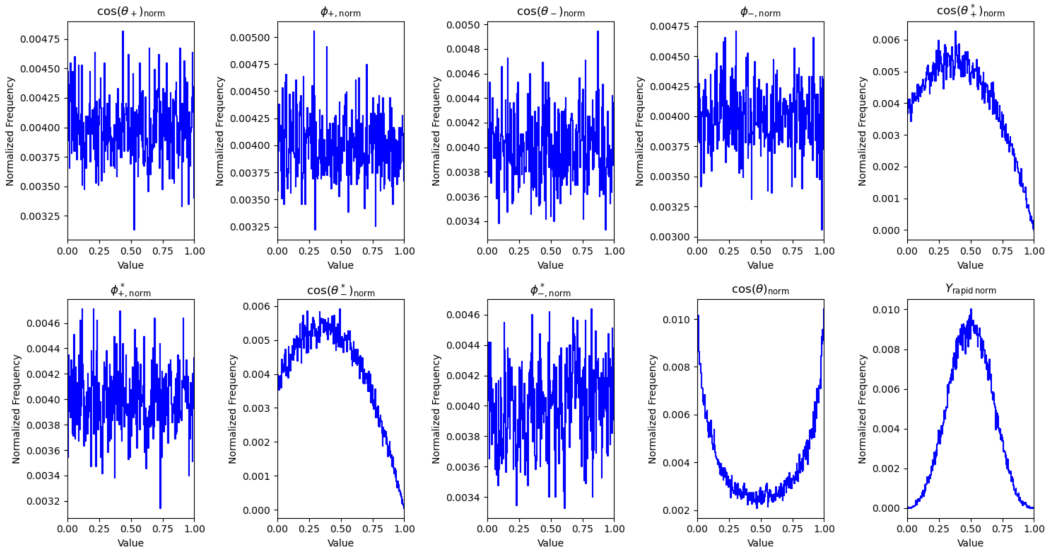
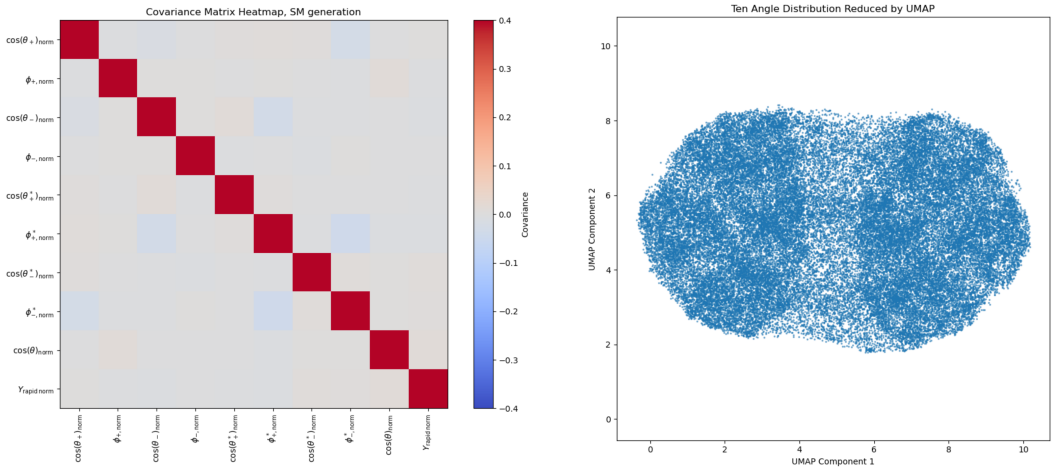


Figure 2: Normalized 10-angle marginal distributions.



(a) (b)

Figure 3: (a) Covariance heatmap of 10 angles; (b) UMAP representation of 10 angles.

The angles are weakly correlated, but we can still find some angle pairs, such as $\phi^{*+}cos(\theta^-)$, $\phi^{*-}cos(\theta^+)$, $\phi^{*+}\phi^{*-}$, are negatively correlated. These relations can serve as benchmarks for our simulated data. Also to gain an rough view of the high dimensional distribution, we use UMAP method to compress the 10d manifold into 2d, shown by Fig. 3b. The shape has two dense areas, and this is reasonable because the physics system contains two decaying systems: top quark and anti-top quark. They are correlated with each other and that's why the two dense areas are still connected by data, instead of clustering.

3 Naive Variational Autoencoder (VAE)

The Variational Autoencoder (VAE) model is a generative model that can learn from the input data's probability distribution, and generates simulation data that approximates the target distribution. With the encoder-decoder architecture, the VAE model aims to learn

1. **An encoder** $q_\phi(z|x)$**:** An approximate posterior distribution mapping data x to latent space z.

2. **A decoder $p_\theta(x|z)$:** A generative model reconstructing x from latent space z.

During training process, the naive VAE model maximizes the Evidence Lower Bound (ELBO):

$$\mathcal{L}(\theta, \phi; x) = \mathbb{E}_{q_\phi(z|x)}\big[\log p_\theta(x|z)\big] - \mathrm{KL}\big(q_\phi(z|x)\|p(z)\big)$$

where the first term is related to minimizing the reconstruction loss between the input data x and the output data x', while the second term is called KL-divergence, which can be seen as a regularization term. The KL divergence is always non-negative and it is minimized when $q_\phi(z|x) = p(z)$. Since we are assuming the prior $p(z) \sim \mathcal{N}(0, I)$, the KL term is trying to confine the posterior to a Gaussian distribution. Fig. 4 is the sketch of a VAE model.

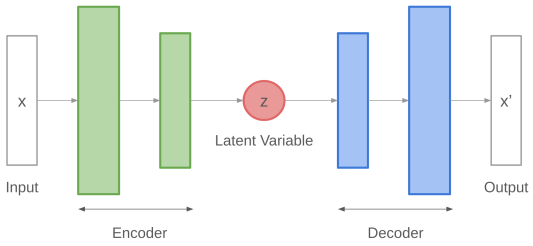


Figure 4: Structure of a naive VAE model.

3.1 Preliminary Study: Two Dimensional Eight-Gauss Peaks

To test the learning ability of the VAE model, we conducted a preliminary study before we really train the model using ten-angle datasets. We started with the two dimensional eight-Gauss peak distribution. The VAE model to generate 8-Gauss peaks is structured with 3 fully connected linear layers as an encoder, with dimensions $(2, 70), (70, 70), (70, K)$; 3 fully connected linear layers as a decoder, with dimensions $(K, 70), (70, 70), (70, 2)$, $K = 25$. The 2d 8-Gauss peak has distribution of $\rho(x, y)$, so the input and output data are in dimension of two. The latent layer dimension is in 25 to fully catch the features of the discrete distributions. The loss function $L = L2 + L_{KL}$, contributed by L2 loss and KL-divergence. The 8-Gauss toy model shows that VAE model can learn complex distributions. Fig. 5 shows the actual performance of the VAE model and the training and tesing loss with respect to epochs.

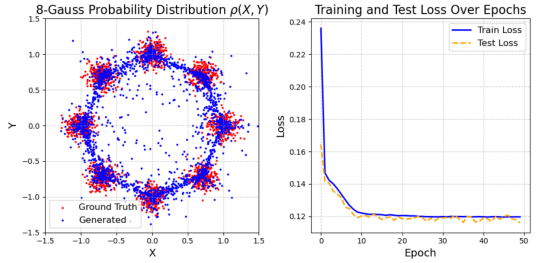


Figure 5: Left: 8-Gauss distribution; Right: Train Loss vs Test Loss

3.2 Ten Angle Distribution

In order to train for ten-angle distribution, we need to construct a more complicated VAE model. The encoder is made of four fully connected layers with dimension of $(10, 2000)$, $(2000, 2000)$, $(2000, 2000)$, $(2000, 200)$, and the decoder is also made of four fully connected layers with dimension of $(200, 2000)$, $(2000, 2000)$, $(2000, 2000)$, $(2000, 10)$. Each layer is batch normalized and then activated by ReLU. The dropout probability is set to 0.5 to reduced overfitting. The table for model hyperparameters is listed:

Table 1: Structure of the VAE Model for Ten-Angle Distribution

Component	Layers and Dimensions	Additional Features
Encoder	Fully connected: $(10, 2000) \rightarrow (2000, 2000)$ $(2000, 2000) \rightarrow (2000, 200)$	Batch Normalization, ReLU
Decoder	Fully connected: $(200, 2000) \rightarrow (2000, 2000)$ $(2000, 2000) \rightarrow (2000, 10)$	Batch Normalization, ReLU
Dropout	Dropout probability: 0.5	

The main difficulty of this task is that the model has to learn from both variable correlations and the variable marginal distributions. And this requires implementing a new loss function[6], which should include:

1. **Reconstruction Loss:** L_2 distance between real data and reconstructed data.
2. **KL Divergence:** Regularize the posterior distribution and prior distribution of the latent variables.
3. **Marginal Loss:** Wasserstein distance between real and reconstructed data.
4. **Sampled Loss:** L_2 distance between real data and simulated data, generated by random sampling from the latent space.

We also use the annealing KL divergence and assign weights to the reconstruction and marginal loss in order to make the model focus more on the correlation and marginal distributions. So our loss function formula is:

$$Loss = (L2 + W_d) * weight_{recon} + KLD * weight_{KL} + Loss_{sample}$$

Setting batch size as 2048 and epoch as 120, learning rate as $1e^{-4}$ and using Adam optimizer, we get the trained VAE model. By random sampling on the latent space, we get the reconstructed data, and the comparison of the real data and reconstructed data is shown by Fig. 6. We can find that our model has problems in recovering the marginal distributions.

From Fig. 7a, the VAE model successfully recovers the negative correlation relation between angle pairs $\phi^{*+}cos(\theta^-)$, $\phi^{*-}cos(\theta^+)$, $\phi^{*+}\phi^{*-}$. Other angle correlateions, as shown by Fig. 3a, are too weak and can hardly be extracted by the VAE model. To improve the performance of the naive VAE model, the dataset size must be increased, and the overall performance of the model needs enhancement.

Fig. 7b reveals differences between the compressed manifold of the real data and the reconstructed data, indicating limitations in the latent space representation. Additionally, Fig. 7c shows that after epoch 40, the testing loss plateaus, remaining nearly constant. This suggests that the naive VAE model has reached its performance limit, and further training will not result in significant improvement. To address these shortcomings, a more complex model beyond the VAE is required.

4 Flow VAE Model

The primary limitation of the VAE model is its assumption that the latent variable follows a standard Gaussian distribution, which restricts the model's expressiveness. To address this limitation and enhance the model's capability, we introduce flow models[5][7] to transform the standard Gaussian latent space into more expressive distributions.

Suppose the beginning distribution is $q(\boldsymbol{z_0}) \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, and the latent variable $\boldsymbol{z_i} = f_i(\boldsymbol{z_j})$, thus we can transform $q(\boldsymbol{z_0})$ into any other distributions $q(\boldsymbol{z_K})$through a chain of K transformations:

$$\mathbf{z}_K = f_K \circ \cdots \circ f_2 \circ f_1(\mathbf{z}_0)$$

$$\ln q_K(\mathbf{z}_K) = \ln q_0(\mathbf{z}_0) - \sum_{k=1}^{K} \ln \left| \det \frac{\partial f_k}{\partial \mathbf{z}_{k-1}} \right|$$

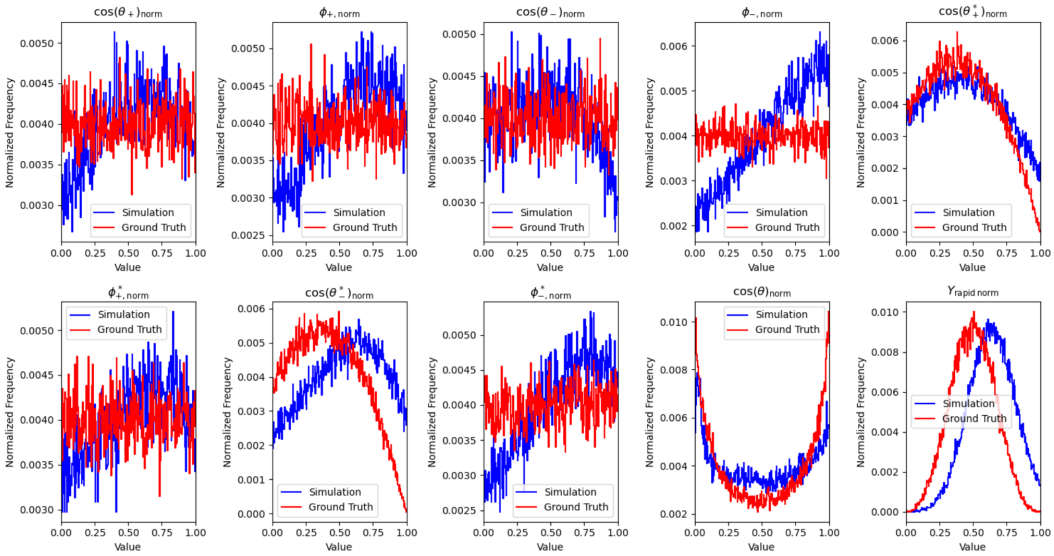
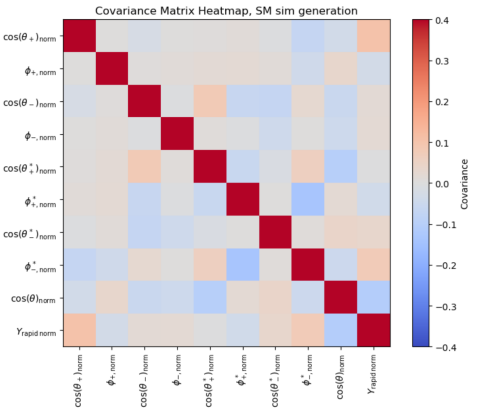
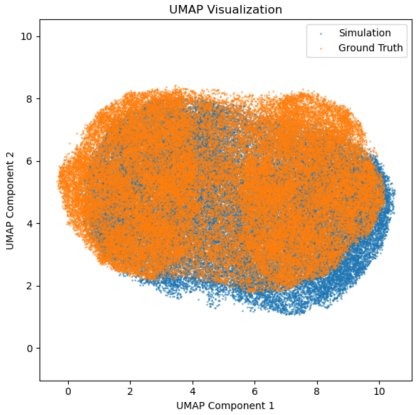


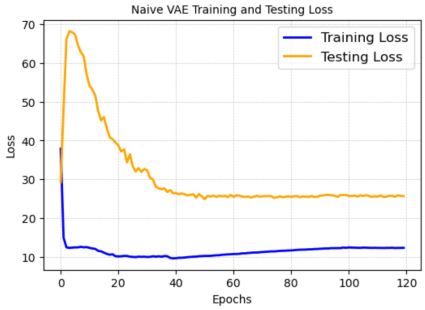
Figure 6: Reconstructed marginal distribution from Naive VAE model: Blue(reconstructed data), Red(real data). The simulation data deviates from the ground truth.



(a) Naive VAE covariance result.



(b) Naive VAE UMAP result.



(c) Naive VAE training loss and testing loss.

Figure 7

In this model, we choose to use radial-flow transformations:

$$z' = z + \beta h(\alpha, r)(z - z_0)$$

where $r = \|z - z_0\|$, and α, β, z_0 are learnable parameters. Thus, keeping the base naive VAE model structure, we add three more radial flow layers to transform the latent space z_0. Fig. 8 shows the structure of the flow VAE model.

Table 2: Structure of the Flow VAE Model for Ten-Angle Distribution

Component	**Layers and Dimensions**	**Additional Features**
Encoder	Fully connected: $(10, 2000) \rightarrow (2000, 2000)$ $(2000, 2000) \rightarrow (2000, 200)$	Batch Normalization, ReLU
Radial Flow Layers	Fully connected: $(70, 70) \times 3$	
Decoder	Fully connected: $(200, 2000) \rightarrow (2000, 2000)$ $(2000, 2000) \rightarrow (2000, 10)$	Batch Normalization, ReLU
Dropout	Dropout probability: 0.5	

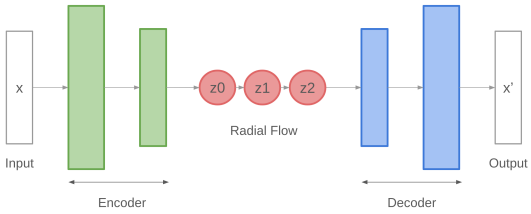


Figure 8: Structure of Flow VAE model

And the loss function is updated to be:

$$Loss = (L2 + W_d) * weight_{recon} + KLD * weight_{KL} + Loss_{sample} - Log(Det(Jacobian))$$

Except for the addition of flow layers, all other settings remain unchanged in training the flow VAE model. By random sampling from the latent space, we obtain the reconstructed data. As shown in Fig. 9, , the marginal distribution of the simulation data aligns well with the ground truth. The uniform, concave, and convex distributions are successfully recovered.

Fig. 10a shows the flow VAE model recovers the same negative correlation relation between $\phi^{*+}cos(\theta^{-})$, $\phi^{*-}cos(\theta^{+})$, $\phi^{*+}\phi^{*-}$ as observed in the naive VAE model. Fig. 10b demonstrates that while the compressed reconstructed manifold still shows some mismatch with the real data, it is more concentrated around the ground truth compared to the naive VAE model. (A more detailed quantitative assessment of the high-dimensional manifold's performance will be included in my future thesis research. The main idea involves expanding this manifold into an angular-series representation and calculating the corresponding expansion coefficients and likelihood.)

Fig. 10c presents a clearer view of the training and testing loss behavior. Initially, the training loss decreases sharply while the testing loss increases significantly — due to model initialization, which temporarily overfits the data. However, as training progresses, both losses decrease smoothly, with the testing loss converging closer to the training loss. This behavior indicates that the model effectively learns the ten-angle distribution, and its performance improves with further training.

5 Summary

In this project, we implemented both the naive VAE model and the flow VAE model to learn the ten-angle distribution. We observed that the flow VAE model outperforms the naive VAE in generating simulation data, particularly in reproducing the marginal distributions. This improvement can be attributed to the normalizing flow, which further reshapes the encoded latent space, making the latent variables more expressive for generation tasks. Additionally, introducing different metrics to the VAE models proved to be significant during training.

However, we also observed that the compressed high-dimensional manifold deviates from the ground truth. This limitation could be intrinsic to the VAE model itself — prior research has reported that VAEs often struggle to generate high-quality data[2]. Alternative models, such as GANs and diffusion models, can produce data of higher quality but are generally more challenging to train. As a result, researchers must find a balance between training complexity and data quality.

Looking ahead, in future research, we plan to explore conditional VAE models[8] to generate beyond Standard Model (BSM) data under varying initial conditions. By leveraging the latent space, we aim to perform interpolation to generate simulation data for any given set of initial conditions without retraining the entire model. However, this is a complex and ambitious task, so we consider it a long-term objective.

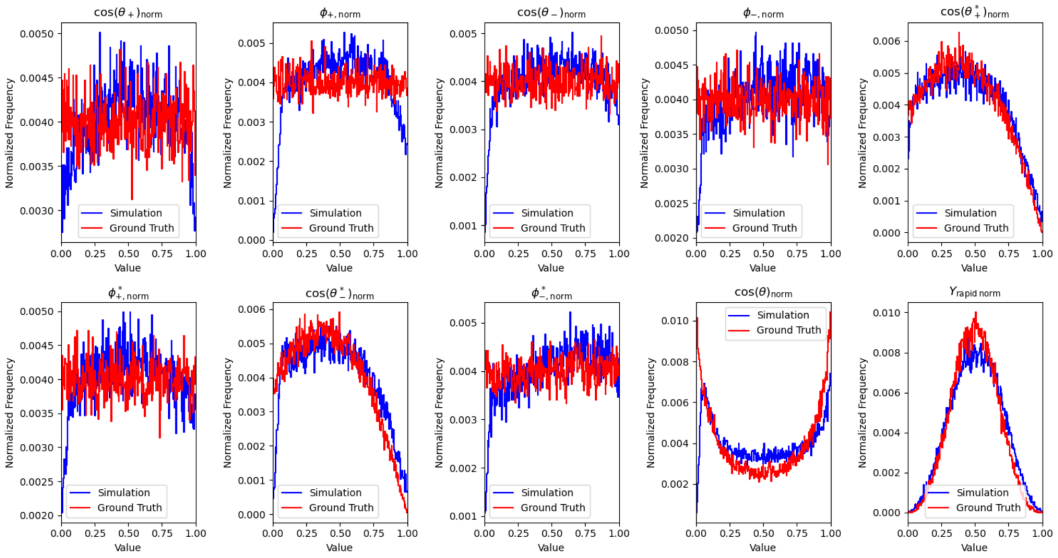
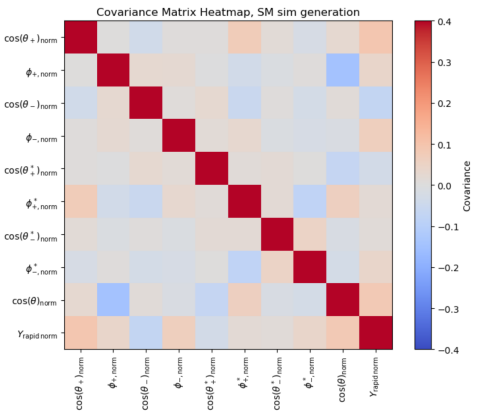
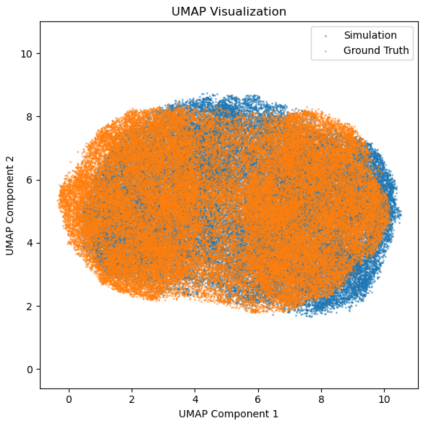


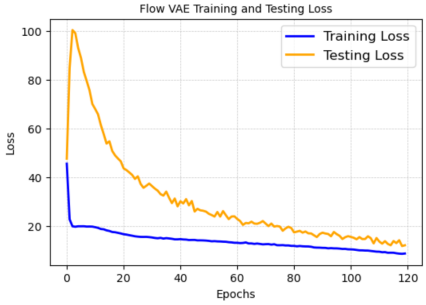
Figure 9: Reconstructed marginal distribution from Flow VAE model: Blue(reconstructed data), Red(real data). The model performance has been significantly improved.



(a) Flow VAE covariance result.



(b) Flow VAE UMAP result.



(c) Flow VAE training loss and testing loss.

Figure 10

References

[1] ATLAS Collaboration. Observation of a new particle in the search for the standard model higgs boson with the atlas detector at the lhc. *Physics Letters B*, 716(1):1–29, September 2012.

[2] Imant Daunhawer, Thomas M. Sutter, Kieran Chin-Cheong, Emanuele Palumbo, and Julia E. Vogt. On the limitations of multimodal vaes, 2022.

[3] Michele Faucci Giannelli. FastCaloGAN: a fast simulation for the ATLAS calorimeter system using GANs. 2020.

[4] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2022.

[5] George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference, 2021.

[6] David K. Park, Yihui Ren, Ozgur O. Kilic, Tatiana Korchuganova, Sairam Sri Vatsavai, Joseph Boudreau, Tasnuva Chowdhury, Shengyu Feng, Raees Khan, Jaehyung Kim, Scott Klasky, Tadashi Maeno, Paul Nilsson, Verena Ingrid Martinez Outschoorn, Norbert Podhorszki, Frederic Suter, Wei Yang, Yiming Yang, Shinjae Yoo, Alexei Klimentov, and Adolfy Hoisie. Ai surrogate model for distributed computing workloads, 2024.

[7] Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows, 2016.

[8] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.