Controlling Quality for a Physics-Driven Generative Models and Auxiliary Regression Approach

Alexander Rogachev[1,*] and *Fedor* Ratnikov[1,2,**]

[1]HSE University
[2]Yandex School of Data Analysis

Abstract. High energy physics experiments heavily rely on the results of MC simulation of data used to extract physics results. However, the detailed simulation often requires tremendous amount of computation resources.
Using Generative Adversarial Networks and other deep learning generative techniques can drastically speed up the computationally heavy simulations like a simulation of the calorimeter response. To be useful, such models are required to satisfy quality metrics which are driven by a specific physics properties of generated objects rather than by a regular ML image-like quality metrics.
The auxiliary regression extension to the GAN-based fast simulation demonstrated improvements of the physics quality for generated objects. This approach introduces physics metrics to a Discriminator path of the model thus allows direct discriminating of objects with poorly reproduced properties.
In this paper we discuss the auxiliary regression GAN approach to physics-based fast simulation and concentrate on requirements to the quality of the auxiliary regressor to provide a necessary precision of the generative models built on top of this regressor.

1 Introduction

High energy physics experiments heavily rely on MC simulations, however, these simulations require significant computational resources. To address this, the use of Generative Adversarial Networks (GANs) and other deep learning generative techniques have shown promising potential to accelerate computationally heavy simulations, such as the simulation of an electromagnetic calorimeter (ECAL) response. However, it is crucial for these models to meet quality metrics that are driven by the specific physics properties of generated objects rather than general image-like quality metrics used in computer vision.

The use of GANs for simulation in high energy physics was introduced by [1] and further developed in [2, 3, 4, 5, 6, 7]. To improve the quality of produced distributions, especially in terms of particular properties used for evaluation, we proposed an extension of the Discriminator called the Auxiliary Regressor, that evaluates specific metrics that we want to reproduce and shares weights with the regular Discriminator [8].

Now we further develop this approach for physics-based fast simulation and focus on the requirements for the precise regressors and its effect on the quality of the generative model built on top of it.

[*] e-mail: airogachev@hse.ru
[**] e-mail: fedor.ratnikov@cern.ch

2 Generative Models for Detector’s Response Simulation

2.1 Dataset

The dataset used in this study contains information on electron interactions within the electromagnetic calorimeter (ECAL). The ECAL employs the "shashlik" technology, which consists of alternating layers of lead and scintillation plates [9]. The readout cells within different modules have varying sizes, ranging from 4×4 to 12×12 cm^2. These cells can be aggregated to obtain a response for different granularities. All events in the dataset correspond to electrons with specific momentum and direction entering the calorimeter at a given location, resulting in the generation of an electromagnetic shower in the ECAL. The sum of energies deposited in the scintillator layers of a single cell produces a matrix of energies corresponding to the ECAL response for the impacting electron. The dataset consists of ECAL response matrices of 2×2 cm^2 cells, with a size of 30×30 cells approximately centered on the energy clusterś location. The matrices are generated using the GEANT4 package.

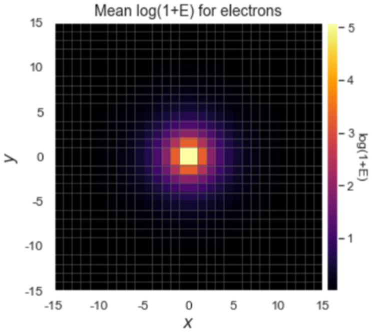

Figure 1: Visualization of the ECAL response dataset.

2.2 Quality evaluation

To evaluate the performance of different models, we use precision and recall for distributions (PRD) [10]. PRD allows us to measure the quality of generated samples by disentangling precision (quality of generated samples) from recall (proportion of target distribution covered by the generated distribution).

In addition to evaluating general quality, we also want to assess the generated responses based on physics metrics. To achieve this, we use the minimum of two PRD-AUC scores, evaluated over raw images and a set of physics statistics. These physics statistics include:

- shower asymmetry along and across the direction of inclination;
- shower width;
- number of cells with energies above a certain threshold (sparsity level).

Since PRD requires discrete distributions as input, we combine the objects from real and generated distributions and cluster them using MiniBatchKMeans. The PRD is then evaluated over the pair of histograms built after the clustering procedure, with a total of 400 clusters.

3 Auxiliary Regressor for GANs

In order to improve the quality of reproduced distributions, especially of those statistics that we use during quality evaluation, we proposed Auxiliary Regressor [8]. Its goal is to evaluate some particular metrics that we want to reproduce on object level. It shares the first layers with the regular Discriminator, thus we have just a lightweight adjustment to the initial number of trainable parameters, as it is shown at Fig. 2. We try to provide the Discriminator network with the information about the desired metrics, expecting it to learn it as now discriminators can detect generated objects with badly reproduces metrics values and show that it allows to boost the quality. Conceptually, it even becomes possible to optimize a quality metric that was not differentiable before, as now we can use backpropagation to train a network that approximates it. The training procedure can follow two approaches:

- **Multi-task.** Both networks are trained simultaneously with two losses, adversarial (Hinge) and regression (MSE) and ones. Objective of Disriminator looks as follows:

$$\begin{aligned}\mathcal{L}_{adv}(\theta) = & -\mathbb{E}_{(x,y)\sim p_{data}}[\min(0, -1 + D(x,y))] \\ & -\mathbb{E}_{z\sim p_z, y\sim p_{data}}[\min(0, -1 - D(G(z),y))],\end{aligned} \tag{1}$$

$$\mathcal{L}_{reg}(\theta) = \sum_{k=1}^{K} \frac{\alpha_k}{N} \sum_{i=1}^{N} (o_i - {}_i)^2, \tag{2}$$

$$\mathcal{L}(\theta) = \mathcal{L}_{adv}(\theta) + \mathcal{L}_{reg}(\theta), \tag{3}$$

  where $\mathcal{L}_{adv}(\theta)$ is the adversarial part of the Discriminator's loss, $\mathcal{L}_{reg}(\theta)$ is the regression one, K is the number of the object's properties, that we evaluate via regressors, α_k is the weight of k-th regression loss, N is the number of objects, o_{real} is the real property value and ${}_i$ is the predicted value.

- **Two-step.** Regression parts are trained for the regression task only using (2). All obtained weights of the regressor are frozen and plugged it into the Discriminator. Finally, the model is trained in an adversarial setting with hinge loss (1) only.

By introducing an additional task into the training procedure we let our model catch some general information that can be useful for both objectives.

As we mentioned in 2.2, asymmetry is used in order to evaluate the performance of the model. Through our experiments in [7, 8], we noticed that this property reproduced the worse, so we wanted asymmetry of generated objects improved and become closer to original one. To achieve it, we added an auxiliary regressor to evaluate asymmetry of a given energy sample and use the output of the regressor as a condition inside Discriminator.

At this point we faced the following question: do we really need a strong regressor? Should we add more parameters to the regression part of the Discriminator? We performed a comparison of different architectures and training settings to study the relationship between the quality of the regression task and generative one.

4 Experiments and results

4.1 Comparing Regressors

To study the relationship between regression and generation qualities we compare different architectures for the regression part:

- CNN-based model from scratch (CNN)

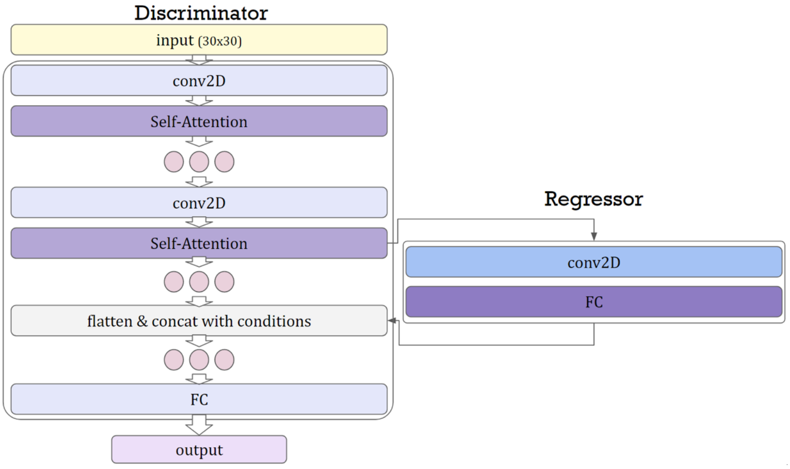

Figure 2: An example of AUX-Regressor extension architecture

Table 1: Generation (PRD) and regression (MSE) quality of compared models.

Model Name	Raw PRD	Phys PRD	MSE
Baseline	0.669	0.877	-
CNN	0.943	0.959	0.297
CNN SA	0.947	0.963	0.279
DiscArc	0.939	0.952	0.300
DiscArch HLR	0.951	0.966	0.235
DiscArch LLR	**0.952**	**0.967**	**0.223**

- CNN-based model with Self-Attentions from scratch (CNN SA)
- Discriminator-like architecture from scratch (DiscArch)
- Pretrained Discriminator-like architecture tuned with high learning rate (DiscArch HLR)
- Pretrained Discriminator-like architecture tuned with low learning rate (DiscArch LLR)

As it is complicated to directly control the quality of regression part of the model in multi-task setting, we perform comparison in two-step setting, allowing us to easily achieve different qualities of regression models. CNN, CNN SA and DiscArch model were trained from scratch using random initialization; DiscArch HLR and LLR have the same architecrute as DiscArch, but we start from pretrained weights of the regression part that were achieved throug [8] multi-task experiments, and then tune this models with different learning rates. After we fit all these models, we freeze all the weighs and plug these models into the discriminator.

Plotting generation quality versus MSE of regression part from Table 1, we may notice from Fig. 3 that the better regressor we have, the better quality of generative part we may achieve. Even the order of models, sorted by the PRD in terms of physical properties and raw images, is the same.

However, even the worst regressor highly improves the quality of generated objects in terms of objective metric. From Fig. 4 we may notice that the asymmetry distibution of

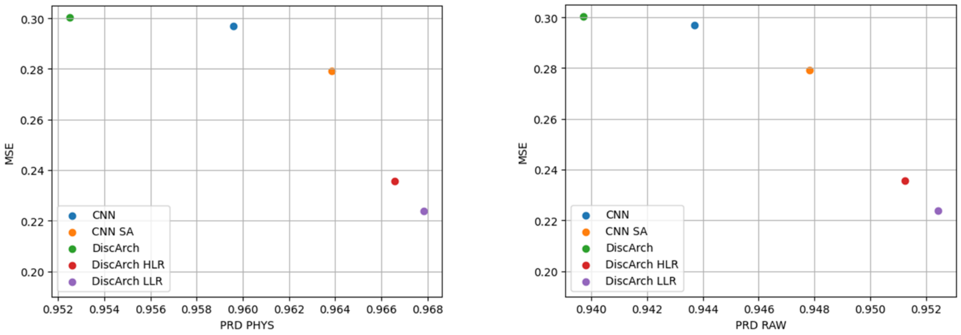

Figure 3: Physical (left) and RAW (right) PRD of compared models versus their regression quality.

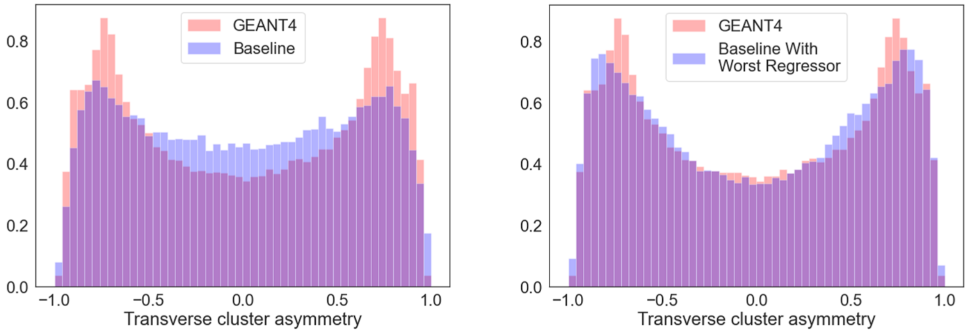

Figure 4: Asymmetry distributions of the baseline (left) and CNN model (right).

generated objects became closer to the real one as we even added the regressor with the highest MSE among the compared models.

4.2 Training settings

Introducing auxiliary regression approach in [8], we fitted GAN in multi-task setting. During our experiments in this paper we used two-step approach. Analyzing the quality of the regression part, we noticed that model with the lowest MSE was pretrained in multi-task manner and regressor with the highest MSE was trained only using regression loss. Both these models have the same architecture, but different training settings. Multi-task approach provided significantly better regressor quality than two-step approach.

Both models predict values that are close to -1 and 1, but regressor, fitted in multi-task setting predicts mid-values significantly better (Fig 5).

5 Conclusion

We demonstrated that incorporating extra surrogate regressor does improve quality of the generative model for ECAL case. We fitted multiple models with different regression capabilities and showed that the better quality the regressors provide, better quality for the chosen

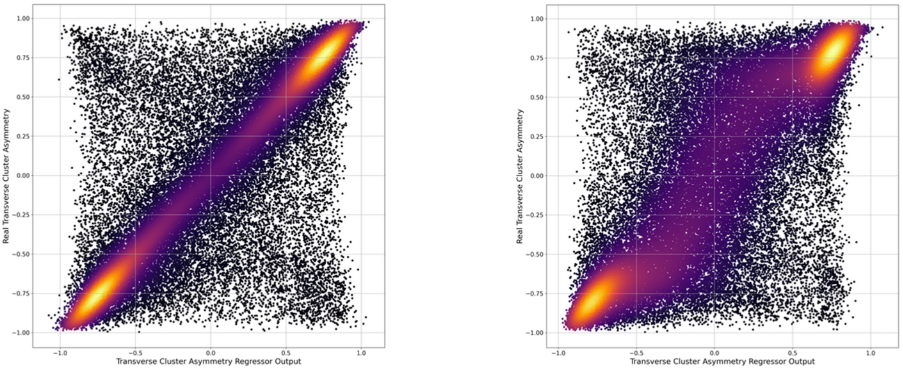

Figure 5: Real and predicted asymmetry of models fitted in multi-tastk(left) and two-step (right) settings.

metrics of the generative model can be achieved in two-step fitting procedure. However, even a regressor with poor quality improves the quality of generated objects significantly. Comparing training settings with two objectives, the Multi-Task approach provided us with better quality.

References

[1] Michela Paganini, Luke de Oliveira, and Benjamin Nachman. "CaloGAN: Simulating 3D high energy particle showers in multilayer electromagnetic calorimeters with generative adversarial networks". In: *Physical Review D* 97.1 (Jan. 2018). ISSN: 2470-0029. DOI: 10.1103/physrevd.97.014021.

[2] Viktoria Chekalina et al. "Generative Models for Fast Calorimeter Simulation: the LHCb case". In: *EPJ Web Conf.* 214 (2019), p. 02034. DOI: 10.1051/epjconf/201921402034.

[3] Martin Erdmann et al. *Generating and refining particle detector simulations using the Wasserstein distance in adversarial networks*. 2018. arXiv: 1802.03325 [astro-ph.IM].

[4] Erik Buhmann et al. "Getting High: High Fidelity Simulation of High Granularity Calorimeters with High Speed". In: *Computing and Software for Big Science* 5.1 (May 2021). DOI: 10.1007/s41781-021-00056-0. URL: https://doi.org/10.1007%2Fs41781-021-00056-0.

[5] Martin Erdmann, Jonas Glombitza, and Thorben Quast. "Precise Simulation of Electromagnetic Calorimeter Showers Using a Wasserstein Generative Adversarial Network". In: *Computing and Software for Big Science* 3.1 (Jan. 2019). DOI: 10.1007/s41781-018-0019-7. URL: https://doi.org/10.1007%2Fs41781-018-0019-7.

[6] Fedor Sergeev et al. "Fast simulation of the LHCb electromagnetic calorimeter response using VAEs and GANs". In: *Journal of Physics: Conference Series* 1740.1 (Jan. 2021), p. 012028. DOI: 10.1088/1742-6596/1740/1/012028.

[7] Alexander Rogachev and Fedor Ratnikov. “Fast simulation of the electromagnetic calorimeter response using Self-Attention Generative Adversarial Networks”. In: *EPJ Web Conf.* 251 (2021), p. 03043. doi: 10.1051/epjconf/202125103043.

[8] Alexander Rogachev and Fedor Ratnikov. *GAN with an Auxiliary Regressor for the Fast Simulation of the Electromagnetic Calorimeter Response*. 2022. doi: 10.48550/ARXIV.2207.06329. url: https://arxiv.org/abs/2207.06329.

[9] S. Amato et al. *LHCb calorimeters: Technical Design Report*. Tech. rep. CERN-LHCC-2000-036. Geneva: CERN, 2000.

[10] Mehdi S M Sajjadi et al. *Assessing Generative Models via Precision and Recall*. 2018. arXiv: 1806.00035 [stat.ML].