

A Survey of Machine Learning based Physics Event Generation

Yasir Alanazi¹, N. Sato², Pawel Ambrozewicz², Astrid N. Hiller Blin², W. Melnitchouk²,
Marco Battaglieri², Tianbo Liu³, Yaohang Li¹

¹Department of Computer Science, Old Dominion University, Norfolk, Virginia 23529, USA

²Jefferson Lab, Newport News, Virginia 23606, USA

³Key Laboratory of Particle Physics and Particle Irradiation (MOE), Institute of Frontier and Interdisciplinary Science, Shandong University, Qingdao, Shandong 266237, China
yalan001@odu.edu, {nsato, pawel, ahblin, wmelnitc, battagli}@jlab.org, liutb@sdu.edu.cn, yaohang@cs.odu.edu

Abstract

Event generators in particle physics play an important role in facilitating the studies of high-energy particle reactions. In this paper, we survey the state of the art of machine learning (ML) efforts at building physics event generators. We start from reviewing the ML generative models used in ML-based event generators (MLEGs) and their specific challenges. We then discuss various approaches of incorporating physics into the ML model designs to overcome these challenges. Finally, we explore several open questions related to super-resolution, fidelity, and extrapolation for physics event generation based on ML technologies.

1 Introduction

In high-energy nuclear and particle physics, accelerators are built to gain insight into elementary particles and composite hadrons and nuclei, by accelerating these to large kinetic energy and colliding them with each other or with nuclear targets. The reactions transform the incoming particles into a set of outgoing particles, referred to as physics “events.” Existing high-energy accelerators include the LHC at CERN for proton-proton collisions, CEBAF at Jefferson Lab for polarized electron-hadron scattering, RHIC at Brookhaven National Lab (BNL) for proton (nucleus)–proton (nucleus) collisions, as well as the future Electron-Ion Collider (EIC). Detecting and analyzing the outgoing particles brings insight into the femto-scale physics involved in these reactions.

Physics event generators, which randomly generate simulated events mimicking those produced in particle accelerators, play a vital role in facilitating the study of matter. The event generators have a broad spectrum of physics applications, including estimating the distribution of expected events for the study of interesting physics scenarios, planning and designing new detectors, optimizing the detector performance under experimental constraints, devising strategies to analyze raw data from experiments, and interpreting observed physics phenomena with fundamental theory [Lehti and Karimaki, 2010].

Since the early 1970s, the simulation of physics events has mainly been implemented by Monte Carlo (MC) meth-

ods, which transform random numbers into simulated physics events. MC event generators (MCEGs) [Mangano and Stelzer, 2005] are constructed by a combination of high-precision data from previous experiments and theoretical inputs. Commonly used MCEGs include Pythia [Sjostrand *et al.*, 2008], Herwig [Bahr *et al.*, 2008], and Sherpa [Gleisberg *et al.*, 2009] for hadronic events; MadGraph [Alwall *et al.*, 2011] and Whizard [Kilian *et al.*, 2011] for parton events; GiBUU [Weil *et al.*, 2012] and HIJING [Bíró *et al.*, 2019] for heavy-ion events; GENIE [Andreopoulos *et al.*, 2010] and NuWro [J., 2009] for neutrino events, as well as specialized event generators such as AcerMC [Kerševan and Richter-Was, 2013], ALPGEN [Mangano *et al.*, 2003], and others.

Recently, along with advances in ML, and particularly deep learning, ML-based generative models for physics event generation have received considerable attention. ML-based event generators (MLEGs) have become an alternative approach to MC simulations of physical processes. Instead of simulating physics events from first principles as in MCEGs, MLEGs employ a data-driven approach to learn from event samples. ML generative models, including Generative Adversarial Networks (GAN) [Goodfellow *et al.*, 2014], Variational Autoencoder (VAE) [Kingma and Welling, 2014], and Normalizing Flows (NF) [Kobyzev *et al.*, 2020], have been adopted to implement MLEGs.

Compared to MCEGs, MLEGs yield several attractive advantages. First, MLEGs can be significantly faster than MCEGs. MC simulations of the complete pipeline of particle experiments, including detector effect simulations, often take minutes to generate a single event, even with the support from modern supercomputers [Buckley, 2020]. In contrast, MLEGs, after proper training, can produce millions of events per second. Fast MLEGs can serve as compactified data storage utilities, eliminating the need for maintaining MCEG event repositories [Chekanov, 2015]. Second, MCEGs rely on theoretical assumptions such as factorization and statistical models, which limit their ability to capture the full range of possible correlations existing in nature between particles momenta and spins. On the other hand, MLEGs trained directly on experimental event data are agnostic of theoretical assumptions, and are able to explore the true underlying probability distributions governing the spectrum of particles produced in a reaction.

In this paper, we provide a comprehensive survey of ML-based methods for physics event generation. Specifically, we review the generative models adopted in state of the art MLEGs and their detector effect simulations. In addition to the well-known issues in training a generative model, MLEGs come along with significant physics-related challenges. We then present the approaches of encoding physics in the ML models to address these challenges. Finally, we discuss the following three open questions with respect to the outlook of MLEG applications:

- Can MLEGs go beyond the statistical precision of the training event samples?
- Can MLEGs faithfully reproduce physics?
- Can MLEGs provide new physics insights?

2 Machine Learning-based Event Generation

In this section, we first review the generative models used in MLEGs. Then we describe the simulation of detector effects, after which we discuss the physics-related challenges in MLEGs and the methods of incorporating physics into ML models to address these.

2.1 Generative Models in MLEGs

Generative Adversarial Networks

GANs are the most popularly used generative models in MLEGs. The regular GAN event generator is composed of two neural networks: a generator G and a discriminator D . The former is trained to generate fake event samples, and the latter is a binary classifier to distinguish the events of the generated distribution P_G from the true events with distribution P_T . G and D are trained under the value function $V(D, G)$

$$\min_G \max_D V(D, G) = \langle \log D(x) \rangle_{x \sim P_T} + \langle \log(1 - D(\tilde{x})) \rangle_{\tilde{x} \sim P_G}. \quad (1)$$

As shown in the original GAN paper [Goodfellow *et al.*, 2014], given an optimal discriminator $D^* = P_T(x)/(P_T(x) + P_G(x))$, training G becomes identical to minimizing the Jensen-Shannon divergence (JSD)

$$\min_G V(D^*, G) = -2 \log 2 + \text{JSD}(P_G || P_T). \quad (2)$$

If JSD becomes 0, then $P_G = P_T$.

Although GANs have demonstrated success in many applications, training a successful GAN model is known to be notoriously difficult [Salimans *et al.*, 2016]. Many GAN models suffer from major problems including mode collapse, non-convergence, model parameter oscillation, instability, vanishing gradient, and overfitting due to unbalanced generator/discriminator combinations. Studies [Ottens *et al.*, 2019; Hashemi *et al.*, 2019] also reported a less satisfactory performance when a regular GAN is used for event generation.

Several improved GAN architectures have been employed in MLEGs to enhance GAN training:

- **Least Squares GAN (LS-GAN):** LS-GAN [Mao *et al.*, 2017] replaces the cross entropy loss function in the dis-

criminator of a regular GAN with a least square term

$$\begin{aligned} \min_D V(D) &= \frac{1}{2} \langle (D(x) - b)^2 \rangle_{x \sim P_T} \\ &+ \frac{1}{2} \langle (D(G(\tilde{x})) - a)^2 \rangle_{\tilde{x} \sim P_G}, \quad (3) \\ \min_G V(G) &= \frac{1}{2} \langle (D(G(x)) - c)^2 \rangle_{x \sim P_G}. \end{aligned}$$

As a result, by setting $b - a = 2$ and $b - c = 1$, minimizing the loss function of LS-GAN yields minimizing the Pearson χ^2 divergence. The main advantage of LS-GAN is that, by penalizing the samples far away from the decision boundary, the generator is pushed to generate samples closer to the manifold of the true samples.

- **Wasserstein GAN (WGAN):** WGAN [Arjovsky *et al.*, 2017] used Wasserstein or Earth-Mover's distance [Villani, 2016] to replace JSD in the regular GAN. Under Kantorovich-Rubinstein duality, the Wasserstein distance is defined as

$$W(P_G, P_T) = \max_{f \in \mathcal{F}} \langle f_w(x) \rangle_{x \sim P_T} - \langle f_w(G(\tilde{x})) \rangle_{\tilde{x} \sim P_G}, \quad (4)$$

where f is a family of K -Lipschitz continuous functions, f_w , parameterized by w in parameter space W . Instead of directly telling fake events from the true ones, the discriminator in WGAN is trained to learn a K -Lipschitz continuous function to minimize the Wasserstein distance. Compared to JSD, Wasserstein distance provides a meaningful and continuous measure of the distance between the event distribution from the generator and the true event distribution, even when they have no overlaps, which helps guide the training of the generator toward the true event distribution and reduce the likelihood of mode collapse.

- **Wasserstein GAN Gradient Penalty (WGAN-GP):** A problem in WGAN is to use weight clipping to maintain K -Lipschitz continuity of f_w during training, which still results in unstable training, slow convergence, and gradient vanishing. WGAN-GP [Gulrajani *et al.*, 2017] replaces weight clipping with gradient penalty to ensure K -Lipschitz continuity and thus further improve WGAN stability. The gradient penalty is calculated as

$$\lambda \langle (\|\nabla_{\hat{x}} f_w(\hat{x})\|_2 - 1)^2 \rangle_{\hat{x} \in p_{\hat{x}}}, \quad (5)$$

where parameter λ balances the Wasserstein distance and gradient penalty, and $p_{\hat{x}}$ is uniformly sampled along lines between event pairs from P_T and P_G .

- **Maximum Mean Discrepancy GAN (MMD-GAN):** MLEGs are particularly concerned about the precise matching between the generated and the true event distributions, where MMD-GAN [Li *et al.*, 2017] can be used to enhance the matching precision. MMD-GAN incorporates an MMD term to the generator loss function:

$$\begin{aligned} \text{MMD}^2(P_G, P_T) &= \langle k(x, x') \rangle_{x, x' \sim P_G} \\ &+ \langle k(y, y') \rangle_{y, y' \sim P_T} - 2 \langle k(x, y) \rangle_{x \sim P_G, y \sim P_T}, \quad (6) \end{aligned}$$

MLEGs	Data Source	Detector Effect	Reaction/Experiment	ML Model
[Hashemi <i>et al.</i> , 2019]	Pythia8	DELPHES + pile-up effects	$Z \rightarrow \mu^+ \mu^-$	regular GAN
[Ottens <i>et al.</i> , 2019]	MadGraph5 aMC@NLO	DELPHES3	$e^+ e^- \rightarrow Z \rightarrow l^+ l^-$, $pp \rightarrow t\bar{t}$	VAE
[Butter <i>et al.</i> , 2019]	MadGraph5 aMC@NLO		$pp \rightarrow t\bar{t} \rightarrow (b\bar{q}q')(b\bar{q}q')$	MMD-GAN
[Di Sipio <i>et al.</i> , 2019]	MadGraph5, Pythia8	DELPHES + FAST-JET	$2 \rightarrow 2$ parton scattering	GAN+CNN
[Ahdida <i>et al.</i> , 2019]	Pythia8 + GEANT4		Search for Hidden Particles (SHiP) experiment	regular GAN
[Alanazi <i>et al.</i> , 2020b] [Velasco <i>et al.</i> , 2020]	Pythia8		electron-proton scattering	MMD-WGAN-GP, cGAN
[Martínez <i>et al.</i> , 2020]	Pythia8	DELPHES particle-flow	proton collision	GAN, cGAN
[Gao <i>et al.</i> , 2020]	Sherpa		$pp \rightarrow W/Z + n$ jets	NF
[Howard <i>et al.</i> , 2021]	MadGraph5 + Pythia8	DELPHES	$Z \rightarrow e^+ e^-$	SWAE
[Choi and Lim, 2021]	MadGraph5 + Pythia8	DELPHES	$pp \rightarrow b\bar{b}\gamma\gamma$	WGAN-GP

Table 1: List of existing MLEGs.

where $k(\cdot)$ is a kernel function. MMD typically works well in low dimension; however, the power of MMD degrades with dimension polynomially [Ramdas *et al.*, 2015].

A GAN can also be extended to a conditional GAN (cGAN) [Mirza and Osindero, 2014] to be involved in MLEGs to generate events based on an initial reaction condition. For example, [Velasco *et al.*, 2020] generated electron-proton scattering events conditioned on beam energy. [Martínez *et al.*, 2020] simulated LHC parasitic collisions conditioned on missing transverse energy. The condition is injected as an input to the generator along with noise and is then propagated to the discriminator to differentiate fake and true events under the same condition. The cGAN allows MLEGs to explore events under unseen conditions, either interpolatively or extrapolatively.

Variational Autoencoder

A VAE, composed of an encoder network Ψ and a decoder network Φ , is an alternative generative model employed in MLEGs. In MLEGs using a VAE, Ψ projects the events onto latent variables z , and Φ reconstructs the events from z , while z is forced to follow a standard normal distribution. Then, the loss function of the VAE is motivated by variational inference [Blei *et al.*, 2017] via minimizing the Kullback–Leibler divergence (KLD) between the posterior $p(z|x)$ and the encoded prior distribution $q(z) = \mathcal{N}(0, 1)$:

$$\mathcal{L}_{\text{VAE}} = \|x - \Phi(\Psi(x))\|^2 + \eta \text{KLD}(q(z)||p(z|x)), \quad (7)$$

where the first term is the reconstruction error, the second term computes KLD, and η is the harmonic parameter to balance the two.

The VAE can be further improved as a Wasserstein Autoencoder (WAE) [Tolstikhin *et al.*, 2018] by replacing the KLD term with Wasserstein distance in the loss function, for a similar reason as for the GAN:

$$\mathcal{L}_{\text{WAE}} = \|x - \Phi(\Psi(x))\|^2 + \eta' W(q(z), p(z|x)), \quad (8)$$

where η' is the harmonic parameter. [Howard *et al.*, 2021] adopted a Sliced Wasserstein Autoencoder (SWAE) [Kolouri *et al.*, 2018] for their MLEG, using a sliced Wasserstein distance to approximate $W(q(z), p(z|x))$.

Normalizing Flow

Without adopting adversarial learning, the NF is another generative model that has been used in MLEGs. The fundamental idea underlying NFs is the change of variables in probability functions. Under some mild conditions, the transformation can lead to complex probability distributions of the transformed variables. NFs use an invertible mapping (bijection) function f , often implemented as a neural network, to transform a distribution of $x \in \mathbb{R}^D$ into $y \in \mathbb{R}^D$. The transformed probability density function $q(y)$ becomes

$$q(y) = p(x) \left| \det \frac{\partial f}{\partial x} \right|^{-1}. \quad (9)$$

With a series of mappings $f_1 \dots f_k$, an NF is obtained:

$$x_k = f_k \circ \dots \circ f_1(x_0), x_0 \sim q_0(x_0). \quad (10)$$

The NF is able to transform a simple distribution into a complex multi-modal distribution, and has demonstrated success in collider physics simulations [Gao *et al.*, 2020].

Existing MLEGs

Table 1 lists the existing MLEGs. In the literature, GANs, VAEs, NFs, and their various improved architectures, have been used to simulate physics events from different reactions and training datasets. Both [Ottens *et al.*, 2019] and [Butter and Plehn, 2020] reported that the LS-GAN yields better performance than other generative models, not only in terms of better precision, but also that in the explored scenarios they were faster. However, at this point, it is too early to rule out the optimal generative model architecture for general MLEGs, which requires not only computational verification, but also rigorous theoretical justifications.

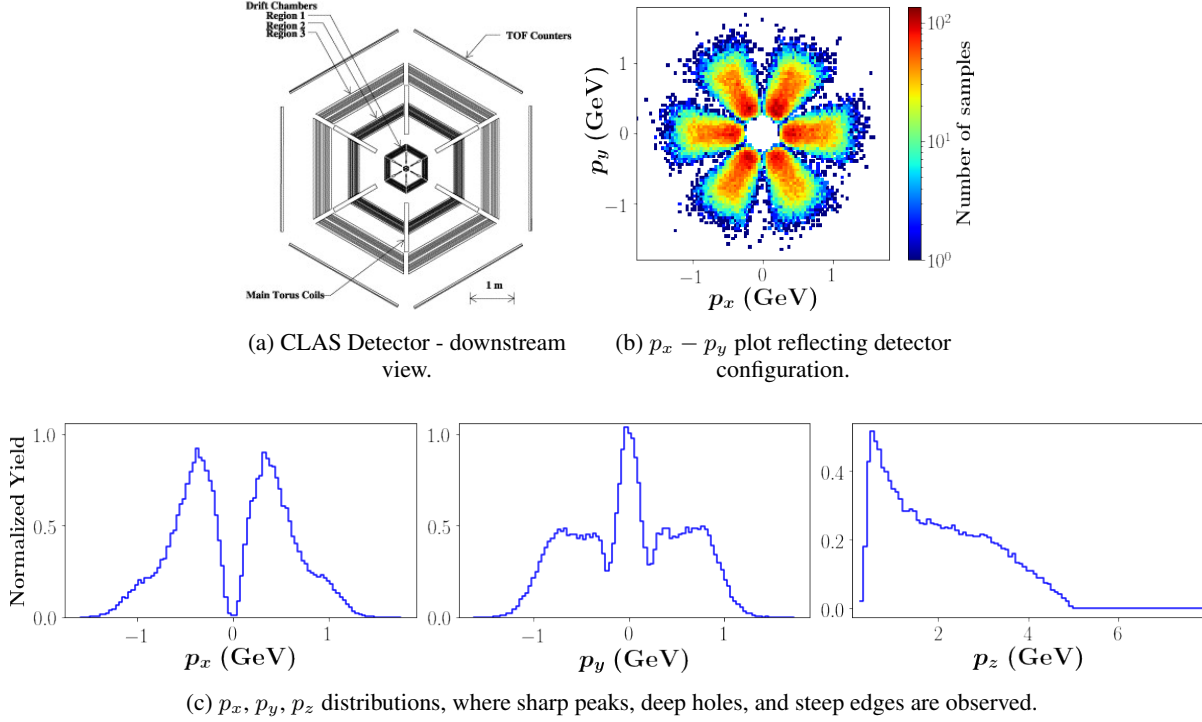


Figure 1: Momentum components distributions of experimental data from an electron scattering experiment with the CLAS detector (a) at the Jefferson National Accelerator Facility. (b) and (c) are generated using 75k samples.

2.2 Detector Effects

Any event generator, whether an MCEG or MLEG, that attempts to faithfully reproduce a specific reaction channel, as seen by a specific experimental apparatus, needs to take into account not only the primary interaction at its vertex but also the interactions of the emergent particles with materials, along their trajectories, and with the devices detecting them. The former should take into account energy losses, decay, new particle production, as well as multiple scattering effects, while the latter should carefully model detector responses to particles being detected.

The experimental setup introduces a detection volume, or acceptance, which is usually quite complicated. The acceptance covers only a portion of the phase space of the reaction, and has to be modeled employing MC packages, such as GEANT4 [Agostinelli *et al.*, 2003], DELPHES [Ovyn *et al.*, 2009], FLUKA [Böhlen *et al.*, 2014], or similar.

We can classify the detector effects into three categories: acceptance-, smearing-, and misidentification-related. All these effects are mitigated in the MLEGs by using well designed procedures that allow either to remove or to introduce these effects into the synthetic data. The former is known as “unfolding”, and the latter as “folding.” These procedures usually involve training GANs with additional information introduced by modifying loss functions to improve stability and convergence [Musella and Pandolfi, 2018], using fully conditional GAN (FCGAN), where the conditioning is done on the detector response [Bellagente *et al.*, 2020], or employing Wasserstein distance based loss function in a conditional

GAN framework (WGAN) [Erdmann *et al.*, 2018].

2.3 Additional Physics-related Challenges

Compared to many applications employing machine learning generative models to produce images, music, and arts, using MLEGs to simulate events from particle reactions poses new additional physics-related challenges for machine learning:

- i Events generated by MLEGs should not violate physics laws, such as energy and momentum conservation;
- ii MLEGs for generating particle physics events are required to model the distributions of event features and their correlations sufficiently precisely for the nature of particle reactions to be correctly replicated;
- iii The distributions of events exhibit natural, physics driven patterns, such as discrete attributes, prominent and narrow peaks, or symmetric behavior of certain physical quantities. On top of that they also exhibit artificial, detector-related, patterns, such as acceptance induced holes and gaps, and efficiency based regions of lower particle occupancy, which complicate MLEGs; and
- iv The outgoing particles, with increasing incident energy, will yield increased dimensionality of the emergent products.

It is important to note that all existing MLEGs listed in Table 1 are trained using simulated data generated from MCEGs, such as Pythia, MadGraph, and Sherpa. When

learning from real experimental data, it adds an additional level of complications to the MLEGs. Figure 1 shows the momenta plots of experimental data from an electron scattering experiment with the CEBAF Large Acceptance Spectrometer (CLAS) detector at the Jefferson National Accelerator Facility. Due to the configuration of superconducting coils of the torus magnet, spaced by angles of 60° , the detector packages [Adams *et al.*, 2001] shown in Figure 1a are accordingly divided into six sectors so events that fall into the coils are not detected, leaving six gaps in any particle transverse momentum components plot, p_x and p_y , shown in Figure 1b. As a result, the 1D plots of p_x , p_y , and the remaining longitudinal component, p_z , as shown in Figure 1c, yield spikes, deep holes, and sharp edges, which pose difficulty for MLEGs to precisely learn their inherent physics laws as well as the detector patterns.

2.4 Incorporating Physics into ML Model

To address the above physics-related challenges, an important approach is to incorporate physics laws into the generative models. When physics laws are appropriately encoded into the generative models, it can lead to a reduction of the degrees of freedom of the problem and thus improves the performance of MLEGs.

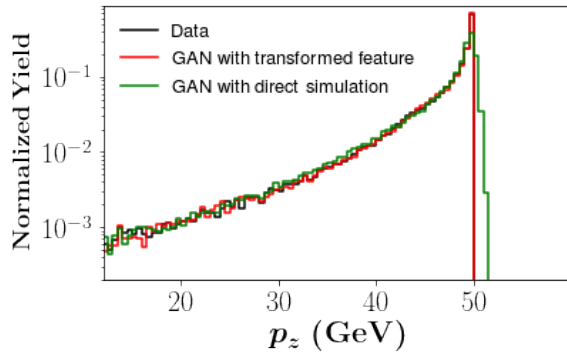


Figure 2: Shows the distribution of p_z generated by GAN with the transformed features (red), and the direct simulation GAN (green), and the true distribution from Pythia (black).

One way to incorporate a physics law into the generative models is via feature engineering. When simulating the inclusive scattering electrons, [Alanazi *et al.*, 2020b] found that direct simulation GAN generates non-physical events violating energy conservation law. As shown in Fig. 2, a sharp edge in the particle energy E distribution arises from energy conservation, which restricts E to be less than the input beam energy, E_b . This sharp edge is very difficult to learn for the inclusive GAN, whose output is the momenta 3-vector (p_x, p_y, p_z) of the electron, as non-physical events can be generated with $E > E_b$, which the discriminator is not sensitive enough to differentiate from the eligible physics events, particularly when $E_b - E$ is small. To address this problem, a transformation $\mathcal{T}(p_z) = \log(E_b - p_z)$ is applied to replace p_z as the output variable of the generator, which avoids the production of non-physical particles. As shown in Fig. 2, the transformation $\mathcal{T}(p_z)$ improves the sensitivity of the discriminator, yielding a significantly better match of p_z distribution

with the data. Another example is, [Hashemi *et al.*, 2019] took into account the symmetries of process $Z \rightarrow \mu^+ \mu^-$ and pre-processed the event samples so that the azimuthal angle of the leading charged lepton is always zero, which resulted in a substantial improvement in terms of agreement with the testing samples.

Another approach to incorporate physics into generative models is to make the latent variables physically meaningful. Typically, the noise fed to GAN or the latent variables in VAE follow certain well-known, easy-to-generate distributions such as Gaussian or uniform without physics meanings. [Howard *et al.*, 2021] expressed the latent distributions in SWAE with quantum field theory and thus sampling the latent space became efficient and was able to infer physics.

3 Open Questions

Compared to MCEGs which have been dedicatedly developed in over 50 years, MLEGs are in their infant stage, bringing a lot of anticipations as well as many challenges and questions without clear answers yet. In this section, we discuss three open questions/arguments with respect to the applications of MLEGs.

3.1 Can MLEGs Display Super-Resolution?

One of the very attractive properties of generative models is super-resolution [Ledig *et al.*, 2017], i.e., generating samples going beyond the resolution of its training samples. Correspondingly, in physics event generation, a question that has been debated in literature is whether the generated events can add statistics beyond that of the training sample or not. That is, can the MLEG generate data that does not only reproduce the examples seen in the training data, but produces additional, diverse, and realistic samples that are more useful for downstream tasks than the original training data?

It has been claimed in [Matchev and Shyamsundar, 2020] that, since the network does not add any physics knowledge, one can only achieve as much statistical precision as the training sample's. The main reason for this statement was that an MLEG does not learn to mimic the true event generator of the training sample, but rather the data of the sample it was trained on. This would imply that one cannot improve the model or parameter discrimination by increasing statistics with the MLEG. In addition, the statistical uncertainty of the training samples would enter the MLEG as systematical uncertainty, thus creating an even more stringent overall uncertainty than in the original data. It is stated, nevertheless, that the MLEG could still offer a better relation between accuracy and computational and storage resources than MCEGs or real experimental data.

However, the findings in [Butter *et al.*, 2020] show empirically that the above claim of [Matchev and Shyamsundar, 2020] is neither well founded nor fulfilled, thus declaring MLEG as a promising venue for the amplification of training statistics. Moreover, that work quantifies to which extent the events can be amplified before being limited by the statistics of the training samples. The argumentation in favor of augmentation feasibility relies on the fact that MLEG are powerful interpolation tools even in high-dimensional spaces. Despite of being model-agnostic interpolators, the interpolation

functions do fulfill basic properties such as smoothness, and therefore can add to the discrete data sets in a reliable fashion, by enabling denser binning, i.e., higher resolution. In fact, it was shown that an MLEG can achieve the same precision as the minimal precision of a fit with a known functional form, saturating at when a number of generated events is reached that is up to orders of magnitude larger than the size of the original training set. While this means that the MLEG cannot outperform the precision of a functional fit, at the same time it reassures us that in those cases where the functional form of the fitting curve is unknown – which represents most of the realistic physics scenarios –, the MLEG becomes a powerful tool for precision augmentation.

3.2 Can MLEGs Faithfully Reproduce Physics?

Whether MLEGs can fully represent the underlying physics of a reaction and then faithfully reproduce physical events or not is critical to many applications where MLEGs are proposed. In the existing MLEGs listed in Table 1, the agreement between the events generated by MLEGs and the sample events is mostly measured in 1D using χ^2 or other statistical metrics. In practice, physicists are often interested in the correlation among event feature distributions. Another issue is related to the rare events occurring in the reaction, which are often precisely those of significant interest to physics. However, these rare events pose a difficult challenge to MLEGs, which often bias to more frequent events during their training process.

The methods of incorporating physics into MLEGs, as described in Section 2.4, certainly help improve the precision of MLEGs. Augmenting the generated features of MLEGs to other important physics properties of interest, as described in [Alanazi *et al.*, 2020a], also increases the sensitivity of the discriminator in the GAN event generator and thus enhances the quality of the generated events. Nevertheless, at this point, there is lack of a comprehensive evaluation framework to thoroughly evaluate the quality of MLEG events in comparison with those from MCEGs or from experiments, particularly in quantifying the correlation among event features with physics meanings as well as measuring the quality of the rare events.

3.3 Can MLEGs Provide New Physics Insights?

Can an MLEG go beyond the manifold of its training event data and bring physical insight into regions without any training (experimental) data? We start the discussion of this question from the extrapolation capability of neural networks. A major drawback of a neural network is its difficulty in extrapolation. The theoretical explanation is rooted from the “universal approximation property” of a feedforward neural network [Scarselli and Tsoi, 1998], i.e., a neural network can approximate any continuous function and many discontinuous functions by adjusting its parameters with respect to its training samples. Therefore, for the space outside of the range of the training samples, the output of a neural network is not reliable. MLEGs trained using GAN, VAE, or NF are fundamentally neural networks, which thus inherit the extrapolation disadvantage. [Velasco *et al.*, 2020] showed that the

MLEG based on cGAN yields good agreement for interpolating events between training beam energy levels but less good agreement for extrapolating events beyond training beam energy levels, particularly for events related to beam energies further away from the training beam energies. This is consistent with the above analysis.

There are two potential ways to extend MLEGs to generate correct events in the unknown regions. One is to apply regularization, which forces the generator to adopt a simple model to limit the degrees of freedom of its neural network and avoid overfitting. More specifically, incorporating known physics laws into the regularizers helps generalize the neural networks and reduce their variation to explore the unknown physics laws and theories. The other way is to generate artificial data samples within the unknown region by either physics theory or simulation to correct the behavior of MLEGs. Both approaches can also be combined to allow MLEGs, at least at some extent, to extrapolate.

4 Conclusion and Outlook

Along with the advance of ML methods, MLEGs emerge as an alternative approach to MCEGs to generate simulated physical events mimicking those generated in high energy physics accelerators. Compared to MCEGs, MLEGs demonstrate attractive advantages including the fast event generation and the quality of being agnostic of theoretical assumptions.

The development of MLEGs is still in its early stage. In this survey paper, we review the ML generative models used in existing MLEGs, including GAN, VAE, NF, and their enhanced architectures. Some studies reported that LS-GAN yields better event quality over other generative architectures, but they lack theoretical justifications. MLEGs, as practical tools to simulate physical events, pose additional challenges related to physics. We also review the methods of incorporating physics into MLEGs to address these challenges. We further explore the open questions on the capability of MLEGs on super-resolution, faithful reproduction of physics, and extrapolation. The answers to these open questions will have significant impact on the applications of MLEGs.

It is important to note that MLEGs are not likely to replace MCEGs, which are used to verify the underlying theory when compared with experimental data. MLEGs, on the other hand, can serve the purpose of remedying MCEGs’ statistical weakness, particularly if the MLEGs’ super-resolution capability is confirmed. If MLEGs are extended with extrapolation capability, they may interest broad applications to bring in new physics insights. When MLEGs’ faithfulness of reproducing physics is well-justified, MLEGs can also be used as a compactified data storage utility to efficiently store and regenerate physical events.

Unlike many generative model applications, such as producing sharp looking images or fancy objects, where the distribution agreement between the generated samples and the truths is often not strictly enforced, the general requirements underlying MLEGs are to precisely reproduce a specific target distribution. The generative models developed in MLEGs, which incorporate domain knowledge into the machine learning algorithms to faithfully generate samples mimicking com-

plex target distributions, have the potential to be applied to broader applications, such as bioinformatics [Liu *et al.*, 2019] and cosmology [Li *et al.*, 2021].

Acknowledgements

We thank the CLAS Collaboration, particularly Florian Hauenstein for his assistance on extracting the electron scattering data from the Jefferson National Accelerator Facility. We would also like to thank Jianwei Qiu for helpful discussions. This work was supported by the LDRD project No. LDRD19-13, No. LDRD20-18, and No. LDRD21-22.

References

- [Adams *et al.*, 2001] G. Adams, V. Burkert, R. Carl, T. Carstens, V. Frolov, L. Houghtlin, G. Jacobs, M. Kossov, M. Klusman, B. Kross, M. Onuk, J. Napolitano, J.W. Price, C. Riggs, Y. Sharabian, A. Stavinsky, L.C. Smith, W.A. Stephens, P. Stoler, W. Tuzel, K. Ullrich, A. Vlassov, A. Weisenberger, M. Witkowski, B. Wojtkowski, P.F. Yergin, and C. Zorn. The clas cherenkov detector. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 465(2):414–427, 2001.
- [Agostinelli *et al.*, 2003] S. Agostinelli *et al.* Geant4-a simulation toolkit. *Nuclear Instruments and Methods in Physics Research Section A* 506(3), pp 250, 2003.
- [Ahdida *et al.*, 2019] C. Ahdida, R. Albanese, A. Alexandrov, A. Anokhina, S. Aoki, G. Arduini, E. Atkin, N. Azorskiy, J.J. Back, A. Bagulya, and *et al.* Fast simulation of muons produced at the ship experiment using generative adversarial networks. *arXiv*, 1909.04451, 2019.
- [Alanazi *et al.*, 2020a] Y. Alanazi, P. Ambrozewicz, M. P. Kuchera, Y. Li, T. Liu, R. E. McClellan, W. Melnitchouk, E. Pritchard, M. Robertson, N. Sato, R. Strauss, and L. Velasco. Ai-based monte carlo event generator for electron-proton scattering. *arXiv*, 2008.03151, 2020.
- [Alanazi *et al.*, 2020b] Y. Alanazi, N. Sato, T. Liu, W. Melnitchouk, M. P. Kuchera, E. Pritchard, M. Robertson, R. Strauss, L. Velasco, and Y. Li. Simulation of electron-proton scattering events by a feature-augmented and transformed generative adversarial network (fat-gan). *arXiv*, 2001.11103, 2020.
- [Alwall *et al.*, 2011] J. Alwall, M. Herquet, F. Maltoni, O. Mattelaer, and T. Stelzer. Madgraph 5: going beyond. *Journal of High Energy Physics*, 128(2011), 2011.
- [Andreopoulos *et al.*, 2010] C. Andreopoulos *et al.* The GENIE Neutrino Monte Carlo Generator. *Nucl. Instrum. Meth. A*, 614:87–104, 2010.
- [Arjovsky *et al.*, 2017] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein gan. *arXiv*, 1701.07875, 2017.
- [Bahr *et al.*, 2008] M. Bahr *et al.* Herwig++ Physics and Manual. *Eur. Phys. J.*, C58:639–707, 2008.
- [Bellagente *et al.*, 2020] M. Bellagente, A. Butter, G. Kasieczka, T. Plehn, and R. Winterhalder. How to gan away detector effects. *arXiv*, 1912.00477, 2020.
- [Blei *et al.*, 2017] D. Blei, A. Kucukelbir, and J. McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.
- [Buckley, 2020] A. Buckley. Computational challenges for MC event generation. *J. Phys. Conf. Ser.*, 1525(1):012023, 2020.
- [Butter and Plehn, 2020] A. Butter and T. Plehn. Generative networks for lhc events. *arXiv*, 2008.08558, 2020.
- [Butter *et al.*, 2019] A. Butter, T. Plehn, and R. Winterhalder. How to gan lhc events. *arXiv*, 1907.03764, 2019.
- [Butter *et al.*, 2020] A. Butter, S. Diefenbacher, G. Kasieczka, B. Nachman, and T. Plehn. Ganplify-ing event samples. *arXiv*, 2008.06545, 2020.
- [Bíró *et al.*, 2019] G. Bíró, G. G. Barnaföldi, G. Papp, M. Gyulassy, P. Lévai, X. Wang, and B. Zhang. Introducing hijing++: the heavy ion monte carlo generator for the high-luminosity lhc era. *arXiv*, 1901.04220, 2019.
- [Böhlen *et al.*, 2014] T. T. Böhlen, F. Cerutti, M. P. W. Chin, A. Fassò, A. Ferrari, P. G. Ortega, A. Mairani, P. R. Sala, G. Smirnov, and V. Vlachoudis. The FLUKA Code: Developments and Challenges for High Energy and Medical Applications. *Nucl. Data Sheets*, 120:211–214, 2014.
- [Chekanov, 2015] S. V. Chekanov. Hepsim: A repository with predictions for high-energy physics experiments. *Advances in High Energy Physics*, 2015:1–7, 2015.
- [Choi and Lim, 2021] S. Choi and J. Lim. A data-driven event generator for hadron colliders using wasserstein generative adversarial network. *Journal of the Korean Physical Society*, Feb 2021.
- [Di Sipio *et al.*, 2019] R. Di Sipio, M. Fauci Giannelli, S. Ketabchi Haghighat, and S. Palazzo. DijetGAN: A Generative-Adversarial Network Approach for the Simulation of QCD Dijet Events at the LHC. *JHEP*, 08:110, 2019.
- [Erdmann *et al.*, 2018] M. Erdmann, L. Geiger, J. Glombitza, and D. Schmidt. Generating and refining particle detector simulations using the wasserstein distance in adversarial networks. *arXiv*, 1802.03325, 2018.
- [Gao *et al.*, 2020] C. Gao, S. Höche, J. Isaacson, C. Krause, and H. Schulz. Event Generation with Normalizing Flows. *Phys. Rev. D*, 101(7):076002, 2020.
- [Gleisberg *et al.*, 2009] T. Gleisberg, S. Hoeche, F. Krauss, M. Schonherr, S. Schumann, F. Siegert, and J. Winter. Event generation with SHERPA 1.1. *JHEP*, 02:007, 2009.
- [Goodfellow *et al.*, 2014] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial networks. *Proceedings of the 27th International Conference on Neural Information Processing Systems, (NIPS 2014)*, 2014.
- [Gulrajani *et al.*, 2017] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville. Improved training of wasserstein gans. *arXiv*, 1704.00028, 2017.

- [Hashemi *et al.*, 2019] B. Hashemi, N. Amin, K. Datta, D. Olivito, and M. Pierini. Lhc analysis-specific datasets with generative adversarial networks. *arXiv*, 1901.05282, 2019.
- [Howard *et al.*, 2021] J. N. Howard, S. Mandt, D. Whiteson, and Y. Yang. Foundations of a fast, data-driven, machine-learned simulator. *arXiv*, 2101.08944, 2021.
- [J., 2009] Cezary J. Running nuwro. *arXiv*, 0909.1492, 2009.
- [Kerševan and Richter-Was, 2013] B. P. Kerševan and E. Richter-Was. The monte carlo event generator acermc versions 2.0 to 3.8 with interfaces to pythia 6.4, herwig 6.5 and ariadne 4.1. *Computer Physics Communications*, 184(3):919–985, 2013.
- [Kilian *et al.*, 2011] W. Kilian, T. Ohl, and J. Reuter. Whizard—simulating multi-particle processes at lhc and ilc. *The European Physical Journal C*, 71(9), 2011.
- [Kingma and Welling, 2014] D. Kingma and M. Welling. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations (ICLR 2014)*, 2014.
- [Kobyzev *et al.*, 2020] I. Kobyzev, S. Prince, and M. Brubaker. Normalizing flows: An introduction and review of current methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, page 1–1, 2020.
- [Kolouri *et al.*, 2018] S. Kolouri, P. E. Pope, C. E. Martin, and G. K. Rohde. Sliced-wasserstein autoencoder: An embarrassingly simple generative model. *arXiv*, 1804.01947, 2018.
- [Ledig *et al.*, 2017] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi. Photo-realistic single image super-resolution using a generative adversarial network. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017)*, 2017.
- [Lehti and Karimaki, 2010] S. Lehti and V. Karimaki. Computing methods in high energy physics. *Lecture Notes of Helsinki Institute of Physics*, 2010.
- [Li *et al.*, 2017] C. Li, W. Chang, Y. Cheng, Y. Yang, and B. Póczos. Mmd gan: Towards deeper understanding of moment matching network. *arXiv*, 1705.08584, 2017.
- [Li *et al.*, 2021] Y. Li, Y. Ni, R. A. C. Croft, T. Di Matteo, S. Bird, and Y. Feng. Ai-assisted superresolution cosmological simulations. *Proceedings of the National Academy of Sciences*, 118(19):e2022038118, 2021.
- [Liu *et al.*, 2019] Q. Liu, H. Lv, and R. Jiang. hicGAN infers super resolution Hi-C data with generative adversarial networks. *Bioinformatics*, 35(14):i99–i107, 2019.
- [Mangano and Stelzer, 2005] M. L. Mangano and T. J. Stelzer. Tools for the simulation of hard hadronic collisions. *Annual Review of Nuclear and Particle Science*, (1):555–588, 2005.
- [Mangano *et al.*, 2003] M. L. Mangano, F. Piccinini, A. D. Polosa, M. Moretti, and R. Pittau. Alpgen, a generator for hard multiparton processes in hadronic collisions. *Journal of High Energy Physics*, 2003(07):001–001, Jul 2003.
- [Mao *et al.*, 2017] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley. Least squares generative adversarial networks. *arXiv*, 1611.04076, 2017.
- [Martínez *et al.*, 2020] J. A. Martínez, T. Q. Nguyen, M. Pierini, M. Spiropulu, and V. Jean-Roch. Particle generative adversarial networks for full-event simulation at the lhc and their application to pileup description. *Journal of Physics: Conference Series*, 1525:012081, 2020.
- [Matchev and Shyamsundar, 2020] K. T. Matchev and P. Shyamsundar. Uncertainties associated with gan-generated datasets in high energy physics. *arXiv*, 2002.06307v2, 2020.
- [Mirza and Osindero, 2014] M. Mirza and S. Osindero. Conditional generative adversarial nets. *arXiv*, 1411.1784, 2014.
- [Musella and Pandolfi, 2018] P. Musella and F. Pandolfi. Fast and accurate simulation of particle detectors using generative adversarial networks. *arXiv*, 1805.00850, 2018.
- [Ottens *et al.*, 2019] S. Ottens, S. Caron, W. de Swart, M. van Beekveld, L. Hendriks, C. van Leeuwen, D. Podareanu, R. R. de Austri, and R. Verheyen. Event generation and statistical sampling for physics with deep generative models and a density information buffer. *arXiv*, 1901.00875, 2019.
- [Ovyn *et al.*, 2009] S. Ovyn, X. Rouby, and V. Lemaître. Delphes, a framework for fast simulation of a generic collider experiment. *arXiv*, 1307.6346, 2009.
- [Ramdas *et al.*, 2015] A. Ramdas, S. J. Reddi, B. Póczos, A. Singh, and L. Wasserman. On the decreasing power of kernel and distance based nonparametric hypothesis tests in high dimensions. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI 2015)*, AAAI’15, 2015.
- [Salimans *et al.*, 2016] T. Salimans, I. J. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. *arXiv*, 1606.03498, 2016.
- [Scarselli and Tsoi, 1998] F. Scarselli and A. C. Tsoi. Universal approximation using feedforward neural networks: A survey of some existing methods, and some new results. *Neural Networks*, 11(1):15–37, 1998.
- [Sjostrand *et al.*, 2008] T. Sjostrand, S. Mrenna, and P. Z. Skands. A Brief Introduction to PYTHIA 8.1. *Comput. Phys. Commun.*, 178:852–867, 2008.
- [Tolstikhin *et al.*, 2018] I. Tolstikhin, O. Bousquet, S. Gelly, and B. Schoelkopf. Wasserstein auto-encoders. In *International Conference on Learning Representations (ICLR 2018)*, 2018.
- [Velasco *et al.*, 2020] L. Velasco, Y. Alanazi, E. McClellan, P. Ambrozewicz, N. Sato, T. Liu, W. Melnitchouk,

M. P. Kuchera, and Y. Li. cfat-gan: Conditional simulation of electron-proton scattering events with variate beam energies by a feature augmented and transformed generative adversarial network. In *19th IEEE International Conference on Machine Learning and Applications (ICMLA2020)*, 2020.

[Villani, 2016] C. Villani. *Topics in optimal transportation*. American mathematical Society, 2016.

[Weil *et al.*, 2012] J. Weil, H. van Hees, and U. Mosel. Dilepton production in proton-induced reactions at SIS energies with the GiBUU transport model. *Eur. Phys. J. A*, 48:111, 2012.