Machine learning-based event generator for electron-proton scattering

Y. Alanazi[1], P. Ambrozewicz[2], M. Battaglieri[3], A. N. Hiller Blin[4,5] M. P. Kuchera[6], Y. Li[1], T. Liu[7], R. E. McClellan[2], W. Melnitchouk[2], E. Pritchard[6], M. Robertson[8], N. Sato[2], R. Strauss[8], and L. Velasco[9]

[1]*Department of Computer Science, Old Dominion University, Norfolk, Virginia 23529, USA*
[2]*Jefferson Lab, Newport News, Virginia 23606, USA*
[3]*Istituto Nazionale di Fisca Nucleare, 16146 Genova, Italy*
[4]*Institute for Theoretical Physics, Tübingen University,*
Auf der Morgenstelle 14, 72076 Tübingen, Germany
[5]*Institute for Theoretical Physics, Regensburg University,*
D-93040 Regensburg, Germany
[6]*Department of Physics, Davidson College, Davidson, North Carolina 28035, USA*
[7]*Key Laboratory of Particle Physics and Particle Irradiation (MOE), Institute of Frontier and*
Interdisciplinary Science, Shandong University, Qingdao, Shandong 266237, China
[8]*Department of Mathematics and Computer Science, Davidson College,*
Davidson, North Carolina 28035, USA
[9]*Department of Physics, University of Dallas, Irving, Texas 75062, USA*

We present a new machine learning-based Monte Carlo event generator using generative adversarial networks (GANs) that can be trained with calibrated detector simulations to construct a vertex-level event generator free of theoretical assumptions about femtometer scale physics. Our framework includes a GAN-based detector folding as a fast-surrogate model that mimics detector simulators. The framework is tested and validated on simulated inclusive deep-inelastic scattering data along with existing parametrizations for detector simulation, with uncertainty quantification based on a statistical bootstrapping technique. Our results provide for the first time a realistic proof of concept to mitigate theory bias in inferring vertex-level event distributions needed to reconstruct physical observables.

I. INTRODUCTION

Since the early 1970s, Monte Carlo event generators (MCEGs) have played a vital role in facilitating studies of quantum chromodynamics (QCD) in high-energy scattering processes. From the experimental perspective, MCEGs are a crucial part of the procedure used for modeling the detector response folded into measured quantities ("detector-level") to extract the true energies and momenta of final state particles as produced at the interaction point ("vertex-level"). The development of modern MCEGs, such as PYTHIA [1], HERWIG [2], and SHERPA [3], has been driven by a combination of high-precision experimental data and theoretical inputs. The latter have involved a mix of perturbative QCD methods, describing the dynamics of quarks and gluons at short distances, and phenomenological models that map the transition from quarks and gluons

to observable hadrons, as well as nonperturbative inputs such as parton distribution functions for applications involving hadrons in the initial state [4–9].

While the theoretical assumptions are usually well justified, an approach that mixes data with a model for the underlying physical law which we wish to infer can potentially lead to biased results. Moreover, the need to correct for detector effects typically becomes increasingly difficult in higher dimensions and prevents a faithful reconstruction of vertex-level events in a model independent way. In this work we present a novel approach to build an event-level interpolation tool based on machine learning (ML) that avoids theoretical assumptions about the femtometer-scale physics, and discuss a strategy to correct for detector effects at the event level.

An important application where this approach is particularly needed is in the context of spin physics in inclusive, semi-inclusive and exclusive electron–nucleon scattering. Here, various spin configurations among the initial state particles are prepared in order to explore detailed emergent features of quarks and gluons inside hadrons at modern accelerator facilities, such as COMPASS at CERN, Jefferson Lab, and the future Electron-Ion Collider. Unfortunately,
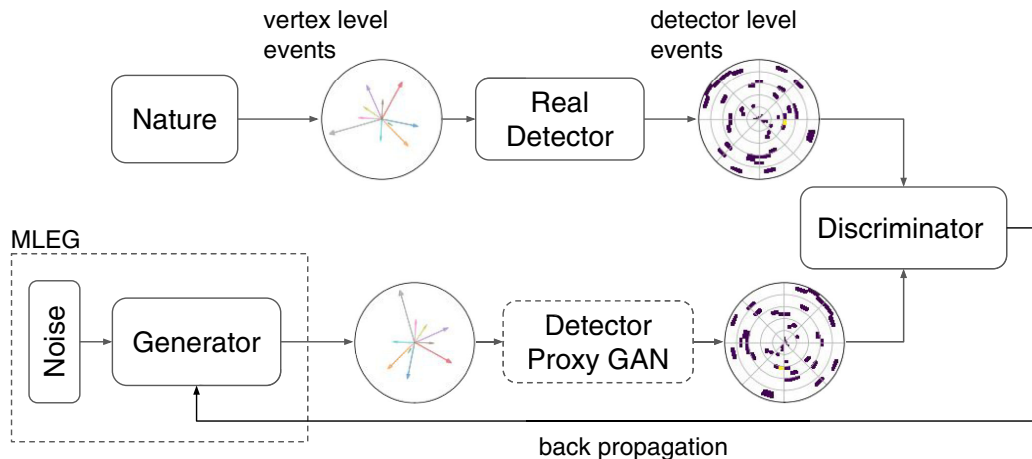
FIG. 1.   Schematic view of the MLEG GAN training framework. The MLEG (dashed box) uses a generator which transforms noise into event-level features. The generator is concatenated with a detector simulator to mimic synthetic detector-level event features. The deep neural network based discriminator compares detector-level event features in order to build gradients to update the generator of the MLEG.

existing theory-based MCEGs are still in their infancy, and at present no MCEG is able to reproduce, even qualitatively, all possible spin asymmetries in these reactions. The reliance on the theory-based MCEGs to extract physics information from these types of measurements inherently introduces biases, which only new developments in the theory behind the MCEGs can mitigate.

In this paper we present an alternative strategy to unfold detector effects by constructing an ML-based event generator (MLEG) using generative adversarial networks (GANs) [10], which have been increasingly utilized recently in high-energy physics applications as a tool for fast Monte Carlo simulations [11–17]. A detailed survey of MLEGs for physics event generation can be found in Ref. [18]. A crucial feature of GANs (and generative models in general) is their ability to generate synthetic data by learning from real samples without explicitly knowing the underlying physical laws of the original system. We present a case study for inclusive electron–nucleon deep-inelastic scattering (DIS) with realistic pseudodata generated from phenomenological models. We first train the MLEG that can faithfully reproduce the phase space of inclusive DIS, along with uncertainty quantification stemming from finite statistics and model architectures. Subsequently, we implement detector effects using an effective parametrization of detectors and train the MLEG and folding algorithms to simulated detector-level DIS events. For the first time a closure test for reconstructing vertex-level DIS events, free of theoretical assumptions, is also performed.

The results provide a new opportunity for experimental data analysis to use the GAN approach to build theory-free event generators which mitigate biases induced in reconstructing physical observables from experimental data. Moreover, the technique provides a new form of data representation that can be easily distributed, in contrast to the traditional data representation via histograms that are limited for processes with high-dimensional phase space.

We begin the discussion in Sec. II with a schematic overview of the MLEG training with our GAN-based event-level interpolator. This is followed in Sec. III by a description of the ML detector surrogate that we use in order to simulate the effects of real particle detectors. The application to inclusive electron-proton DIS is discussed in Sec. IV, where we examine GAN training both with and without detector effects. Finally, in Sec. V we summarize our findings and discuss future extensions and applications.

II. GAN-BASED EVENT-LEVEL INTERPOLATOR

A schematic view of the training workflow of our MLEG GAN is illustrated in Fig. 1, where, as usual, the GAN model is composed of a *generator* and a *discriminator*. The generator converts noise through a deep neural network into event-level features, which is customized by a given reaction. The generated event features are then passed into a detector simulator to convert them as "trial" detector-level events. The discriminator learns through another deep neural network to differentiate the true detector-level event samples from ones produced by the generator and the detector simulator. The GAN training evolves as the generator and discriminator compete adversarially, each updating their parameters during the training process. Eventually, the generator is able to produce synthetic samples that the discriminator can no longer distinguish from the real samples, at which point the training of the MLEG is complete.

Although GANs have demonstrated impressive results in various applications, including generating near-realistic images [19], music [20], and videos [21], training a successful GAN model is known to be notoriously difficult. Many GAN models suffer from major problems, such as
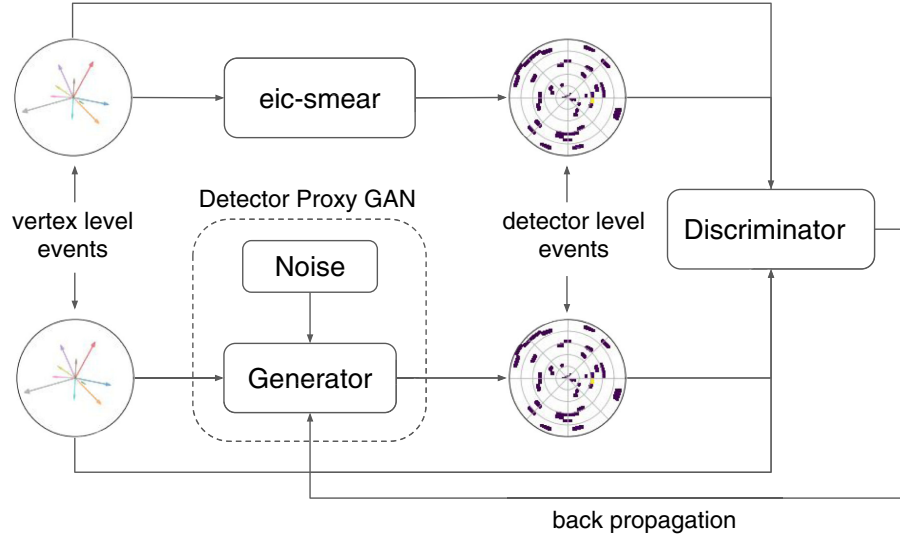
FIG. 2. Schematic view of the ML detector surrogate, where a generator converts input vertex-level event features and noise to detector-level event features. The training samples are obtained from guess vertex-level samples and the corresponding detector-level samples using a detector simulator. The discriminator (right hand side of the figure) is trained simultaneously with vertex-level and detector-level event features in order to minimize the dependence of the generator on the input vertex-level guess samples.

mode collapse, nonconvergence, model parameter oscillation, destabilization, vanishing gradient, and overfitting due to unbalanced training of the generator and discriminator. Approaches and techniques to address these general problems have been proposed and discussed recently in the literature [22–26].

Unlike common GAN applications, such as the generation of realistic high resolution images, the success of our GAN application as nuclear and high-energy physics event generators relies on its ability to faithfully reproduce correlations among the particles' momenta, which are increasingly difficult in higher (greater than one or two) dimensions. At the same time, the corresponding multidimensional momentum distributions or histograms display rapid changes in the phase space that span several orders of magnitude. The challenge is then to design suitable GAN architectures capable of reproducing all of the correlations among the particles, along with a faithful reproduction of the multidimensional histograms across the phase space. In Sec. IV we will discuss in detail about how to customize this for the specific application of inclusive DIS.

III. ML DETECTOR SURROGATE

Experimental data, provided in the form of final state particle momenta, are affected by distortions introduced by experimental detectors. A correction procedure is usually necessary to extract the true information from the measured cross sections and provide the vertex-level distributions used in physics analysis. Such detector effects have multiple causes, including limited acceptance, finite resolution, efficiency distortion, and bin migrations due to radiation and rescattering. Corrections are commonly taken

into account using unfolding procedures that attempt to correct for the detector effects at the histogram level, which require *ad hoc* corrections for each type of observable.

In order to demonstrate that our framework is realizable in a real experimental analysis, such detector effects must be incorporated. For this purpose, we use the open source *eic-smear* software package [27], which was developed at Brookhaven National Laboratory as a fast simulation tool for the future Electron-Ion Collider [28], with smearing capability for quantities such as momentum, energy, polar and azimuthal angles, and provides a simplified parametrization of the response of the detectors. This was used to simulate a simplified version of the H1 and ZEUS detectors with unsegmented 4π acceptance, which made it suitable for our proof of concept problem.

We develop ML-based detector surrogates using a secondary conditional GAN, as illustrated in Fig. 2. The idea is to train a conditional generator simulating the smearing effect of the detector by converting input vertex-level event features and noise into detector-level event features, as dictated by *eic-smear*. To do this we build training samples using trial vertex-level guess event samples and the associated *eic-smear* detector-level samples to train the conditional GAN. Once the conditional GAN is trained, the ML detector surrogate (represented by the dashed box in Fig. 2) can be integrated as the detector simulator in Fig. 1. It is worth noting that for a more realistic description of detector effects, the *eic-smear* parametrization should be replaced by a full GEANT-based detector model [29]. However, its integration within our MLEG models using standard ML libraries is beyond the scope of the present analysis, and will be the subject of future work.

IV. APPLICATION TO INCLUSIVE ELECTRON-PROTON SCATTERING

In this section we describe the application of our MLEG strategy to the inclusive unpolarized DIS of electrons (with four-momentum k) from protons (four-momentum P). Our goal is solely to produce the scattered electron phase space, labeled by the four-momentum k'. As a surrogate for real experimental data, we use pseudodata generated from the Jefferson Lab Angular Momentum (JAM) Collaboration's global QCD analysis framework [30] that has been tuned to describe world data on inclusive DIS and other high-energy scattering processes.

The inclusive electron DIS samples are generated at a center of mass energy of 318.2 GeV, compatible with HERA kinematics, by integrating the 2-dimensional differential cross section $d\sigma/dx\, dQ^2$, computed at next-to-leading order in perturbative QCD using importance sampling, and unweighting events over a very dense binning in (x, Q^2)-space. Each event is transformed into an outgoing electron momentum in the HERA laboratory frame by generating an azimuthal angle relative to the beam axis sampled from a uniform distribution. While our ultimate goal is to apply this approach to real data, this case study provides unique insights of our ML workflow and allows us to identify challenges in formulating a suitable feature space to be learned by the model.

When training the GAN solely using the electron momentum in the laboratory frame as event features, the generator was found to create electron samples that violate momentum conservation near the edge of the phase space, and the model was not sensitive enough to prevent the production of these samples [31]. To alleviate this problem and aid the training, we use a change of variables that enhances the discriminator awareness in these difficult regions. Specifically, we define the scaled variables

$$\nu_1 = \ln((k'_0 - k'_z)/1 \text{ GeV}), \tag{1a}$$

$$\nu_2 = \ln((2E_e - k'_0 - k'_z)/1 \text{ GeV}), \tag{1b}$$

where E_e is the incident electron energy, and k'_0 and k'_z are the scattered electron energy and longitudinal momentum, respectively. In Eqs. (1) the energies and momenta in the arguments of the log are explicitly in units of GeV. These variables can be easily inverted into the original momentum space. In particular, the variable ν_2 changes rapidly as the energy of the outgoing electron approaches its limit, allowing the discriminator to be aware of such region.

In the following, we present details of our chosen ML architecture used for the event-level interpolation and the ML detector surrogate.

(i) MLEG: The input to the generator in Fig. 1 is a 100-dimensional white noise array centered at 0 with unit standard deviation. The generator network consists of 5 hidden dense layers, with 512 neurons per layer, activated by a leaky rectified linear unit (ReLU) function. The number of layers and neurons is optimized to balance execution time and convergence. The last hidden layer is fully connected to a 2-neuron output corresponding to the variables ν_1 and ν_2, activated by a linear function representing the generated features. The corresponding discriminator also consists of 5 hidden dense layers with 512 neurons per layer, optimized as for the generator, and activated by a leaky ReLU function. To avoid overfitting, a 10% dropout rate is applied to each hidden layer. The last hidden layer is fully connected to a single-neuron output, where "1" indicates a true event and "0" a fake event. The discriminator D is trained to give $D(\mathbf{F}) = 1$ for each training sample $\mathbf{F}$, and $D(\tilde{\mathbf{F}}) = 0$ for each sample $\tilde{\mathbf{F}}$ produced by the generator.

(ii) ML detector surrogate: The detector surrogate model is based on a conditional GAN architecture [32]. As shown in Fig. 2, we have a generator that receives vertex-level input in addition to a 100-dimensional white noise centered at 0 with unit standard deviation. The generator learns to fold the inputs and produce detector-level events that mimic the detector response dictated by *eic-smear*. By conditioning the model on vertex-level event features we can enforce learning the correlations between vertex- and detector-level events as opposed to learning a deterministic mapping between inputs and outputs. As for the MLEG, the generator will produce a 2-neuron output corresponding to the detector-level variables ν_1 and ν_2, activated by a linear function representing the generated features, and the discriminator will similarly produce "0" or "1" for training and generated samples, respectively. In both the generator and discriminator architectures of the ML detector surrogate, we use the same number of hidden layers, neurons, dropout rates, and activation functions as in our MLEG. A similar idea of using a GAN for detector effects has been proposed by Bellagente *et al.* [33], where, in contrast to our folding procedure, parton-level data are mapped to detector-level data using a conditional GAN model.

For both of our GAN architectures we adopt the least squares GAN (LSGAN) [34], which replaces the cross entropy loss function in the discriminator of a regular GAN by a least squares term,

$$\min_D V(D) = \frac{1}{2} \langle (D(x) - b)^2 \rangle_{x \sim P_T} + \frac{1}{2} \langle (D(G(\tilde{x})) - a)^2 \rangle_{\tilde{x} \sim P_G}, \tag{2}$$

$$\min_G V(G) = \frac{1}{2} \langle (D(G(x)) - c)^2 \rangle_{x \sim P_G}, \tag{3}$$

where P_G denotes the distribution of the generated samples and P_T is the distribution of the training samples. As a result, by setting $b - a = 2$ and $b - c = 1$, minimizing the loss function of the LSGAN implies minimizing the Pearson χ^2 divergence. For the conditional model, the objective functions can be defined as

$$\min_{D} V(D) = \frac{1}{2} \langle (D(x|y) - b)^2 \rangle_{x \sim P_T, y \sim P_v}$$

$$+ \frac{1}{2} \langle (D(G(\tilde{x}|y)) - a)^2 \rangle_{\tilde{x} \sim P_G, y \sim P_v}, \quad (4)$$

$$\min_{G} V(G) = \frac{1}{2} \langle (D(G(x|y)) - c)^2 \rangle_{x \sim P_G, y \sim P_v}, \quad (5)$$

where P_v denotes the conditioned vertex-level samples that are fed as inputs to the ML detector surrogate. The main advantage of the LSGAN is that by penalizing the samples that are far from the decision boundary, the generator is prompted to generate samples closer to the manifold of the true samples.

Our networks are trained adversarially for 100,000 epochs, where an epoch is defined as one pass through the training dataset. For the optimizer, in both cases we use Adam [35] with a 10^{-4} learning rate, $\beta_1 = 0.5$, and $\beta_2 = 0.9$. To balance the generator and discriminator training, the training ratio is set to 5.

A. GAN training without detector effects

As a first step in our numerical analysis, we train the MLEG using the DIS pseudodata samples without detector effects in order to establish the baseline agreement between training and synthetic data, without the complications introduced by the detector folding. In Fig. 3 we compare the training and synthetic normalized inclusive ep phase space distributions for the scattered electron in the variables ν_1 and ν_2. The uncertainty bands are generated by taking the standard deviation of 10 independently trained GANs, where for each case the training samples are prepared using the bootstrapping procedure (taking random samples with replacement).

It is useful here to define the "pull" metric between the training (JAM) and synthetic (GAN) data by

$$\text{pull} = \frac{\text{E}[\mathcal{P}(\mathcal{O}|\text{bin})]_{\text{GAN}} - \text{E}[\mathcal{P}(\mathcal{O}|\text{bin})]_{\text{JAM}}}{\sqrt{\text{V}[\mathcal{P}(\mathcal{O}|\text{bin})]_{\text{GAN}} + \text{V}[\mathcal{P}(\mathcal{O}|\text{bin})]_{\text{JAM}}}}, \quad (6)$$

where $\text{E}[\mathcal{P}(\mathcal{O}|\text{bin})]$ and $\text{V}[\mathcal{P}(\mathcal{O}|\text{bin})]$ are the expectation values and variances of the discrete probability density $\mathcal{P}$ of an observable $\mathcal{O}$. As expected, the synthetic distributions for ν_1 and ν_2 match well with the distributions from the training samples, within the statistical uncertainties, since for these variables the deviation from the training set is explicitly disfavored by the discriminator. Also shown in Fig. 3 are distributions of derived quantities that are
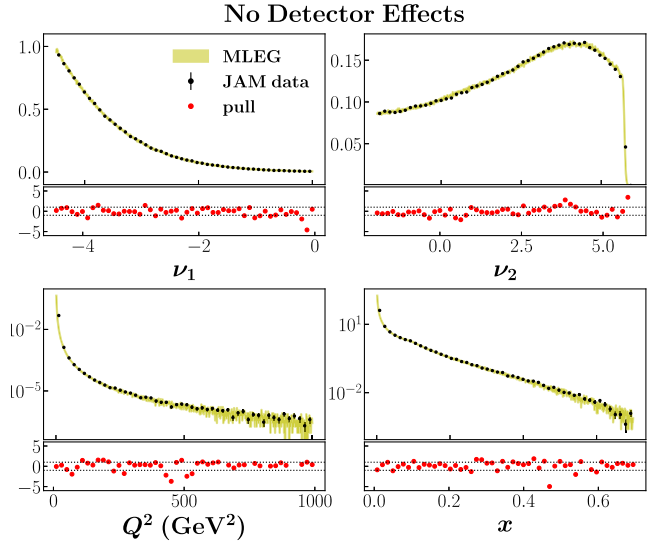

FIG. 3. Comparison of distributions of training and derived variables from JAM training samples (black circles) and GAN-generated synthetic data (yellow bands) for the case of no detector effects; the band size reflects the uncertainty evaluated using the bootstrap procedure (see text). The bottom of each panel shows the pull distributions (red circles) defined in Eq. (6), with the two horizontal dotted lines corresponding to $\pm 1\sigma$.

physically relevant for the DIS process, namely, the four-momentum transfer squared, $Q^2 = -(k - k')^2$, and the Bjorken scaling variable $x = Q^2/2P \cdot (k - k')$. While these observables are obtained by nonlinear transformations of the original variables ν_1 and ν_2, the result accurately reconstructs the matching, within uncertainties, with the corresponding spectra from the training data.

In Fig. 4 we illustrate the reduced inclusive ep DIS cross section, σ_r^{ep} (in practice the reaction involved positrons scattering from protons), as a function of Q^2 in multiple bins of x for the HERA data [36] and for the parametrization of the data from the JAM global QCD analysis [30]. These are compared with the reduced cross sections reconstructed by the GAN. Within the statistical uncertainties, the empirical results are well reproduced by the MLEG simulation in most of the regions of the phase space. Note that the agreement between the JAM fit and the HERA data deteriorates at the largest Q^2 values for each fixed-x spectrum due to the vanishing of the phase space. Nonetheless, the GAN is able to follow the pattern of the phase space distribution, such as the approximate scaling behavior, as well as the drop around the edges of the phase space. This result is quite nontrivial since the relationship between the variables that are learned (ν_1, ν_2) and the DIS variables (x, Q^2) is nonlinear, and it demonstrates the ability of the GAN to learn accurately the underlying probability distribution of the phase space.
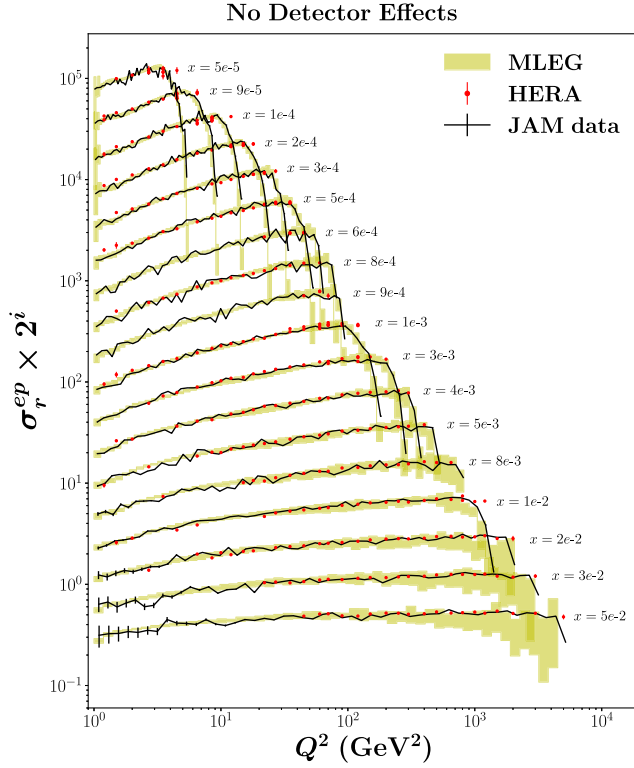
No Detector Effects

FIG. 4. Comparison of the reduced inclusive ep cross section σ_r^{ep} versus Q^2 at fixed values of Bjorken-x from the HERA collider [36] (red circles) with data generated from the JAM global QCD analysis [30] (black solid lines) and the trained GAN (yellow bands). No detector effects are included, and for clarity the cross sections are scaled by a factor 2^i, with i ranging from $i = 0$ for the highest-x value to $i = 17$ for the lowest-x value.

B. GAN training with detector effects

Having established a baseline agreement for our MLEG framework, we proceed to include detector effects, as would be in actual experimental situations, which inevitably increases the complexity of the analysis. As discussed above, we train separately an ML detector surrogate using a detector parametrization provided by the *eic-smear* software. For the trial vertex-level event samples we use directly the samples from the JAM global QCD analysis instead of the flat phase space so as to optimize the GAN training. However, we stress that in principle the model architecture for the detector surrogate can be trained with any samples.

In Fig. 5 we show the vertex- and detector-level distributions for ν_1 and ν_2, where significant distortions are observed for the latter. An issue regarding the change of variables in Eqs. (1) is that after smearing the detector-level k_z' variable can exceed the physical limit given by the incident beam energy E_e, rendering the transformation singular for those unphysical cases. However, since the change of variables, in particular for ν_2, is solely designed to increase the detector awareness in the difficult regions,
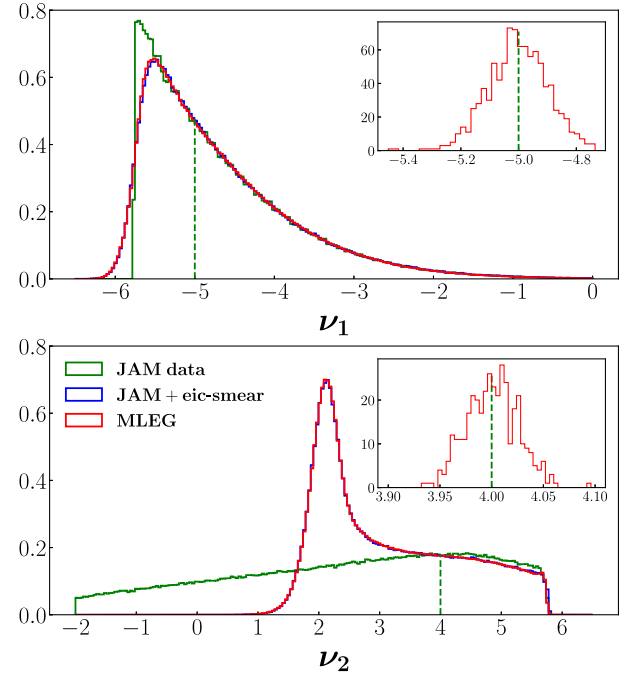

FIG. 5. Comparison of training features at the vertex level (generated, green histograms) and detector level (smeared, blue histograms) with the MLEG generated synthetic data (red histograms). The insets illustrate the local smearing effect at the points indicated by the green vertical dashed lines.

we can replace E_e in Eqs. (1) by the maximum energy found for the detector-level samples to achieve the same goal, and avoid the singularity of the variable transform. This disparity, however, creates an impression of higher levels of distortion in the ν_2 variable compared to ν_1.

We also illustrate the smearing effects by focusing on small intervals in ν_1 and ν_2, as shown in the Fig. 5 insets, to indicate the nontrivial distortion that is taking place across the phase space. Included in Fig. 5 are the corresponding predictions from the detector-level GAN output, which shows very good agreement with the training samples. Note that there are regions where GANs do not match precisely with *eic-smear*, namely, the tail regions at small and large ν_2, which correspond to the edges of the reaction phase space. For the scope of this study, the GAN output represents a reasonable true detector proxy, allowing us to carry out the vertex-level learning closure test and validate the proof of principle of our MLEG framework.

With the ML detector surrogate we proceed with training the MLEG with detector effects. In Fig. 6 we show similar results as in Fig. 3, but this time with detector effects included. As expected, the variables ν_1 and ν_2 are well reproduced, since the discriminator supervises on these variables during the training. Similarly, the predicted DIS variables x and Q^2 at the detector level are well reproduced within the uncertainties.

As the final step, we examine the quality of the MLEG at the vertex level by analysing the direct output of its
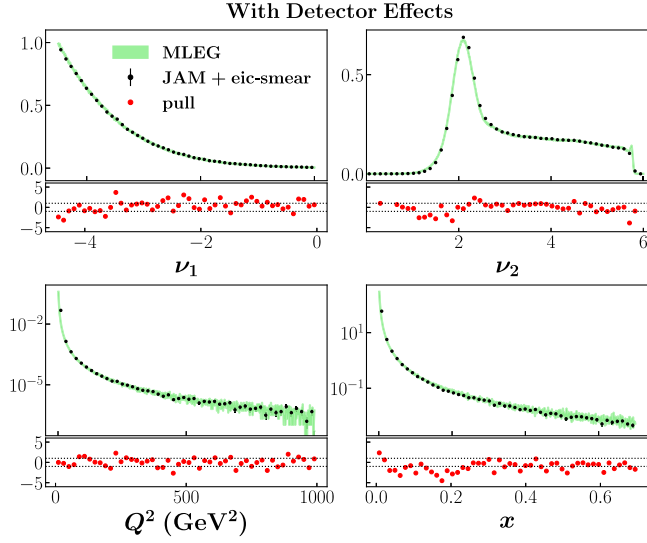
FIG. 6.　As in Fig. 3, but with detector effects present.

generator, and plot in Fig. 7 the corresponding vertex-level distributions. Relative to the detector level, the vertex-level distributions are observed to have, on average, larger values for the pull than those in Fig. 6. This is expected since we do not directly supervise at the vertex level, but instead these are inferred quantities. A more detailed examination of this is shown in Fig. 8, where we plot the reduced cross sections as in Fig. 4, but in the presence of detector effects. As expected, the uncertainties increase due to the detector effects.

One particular region where the deviations from the true result become larger compared to Fig. 4 are those at low Q^2 and high x (see, for instance, the low-Q^2 region for the $x = 0.01$ and 0.02 bins). The difficulty in accurately
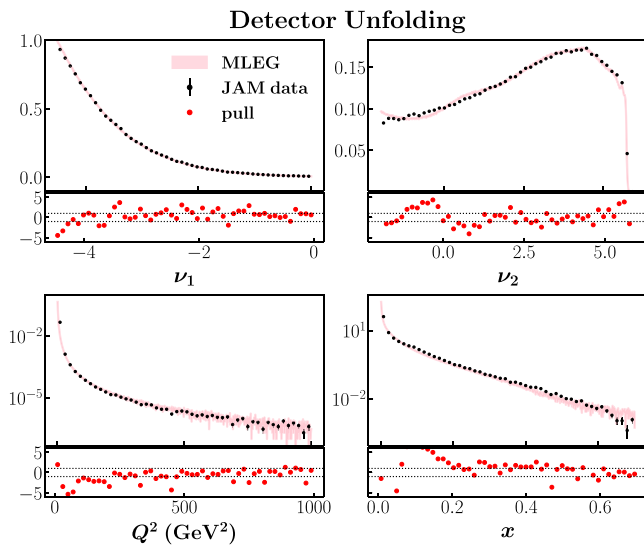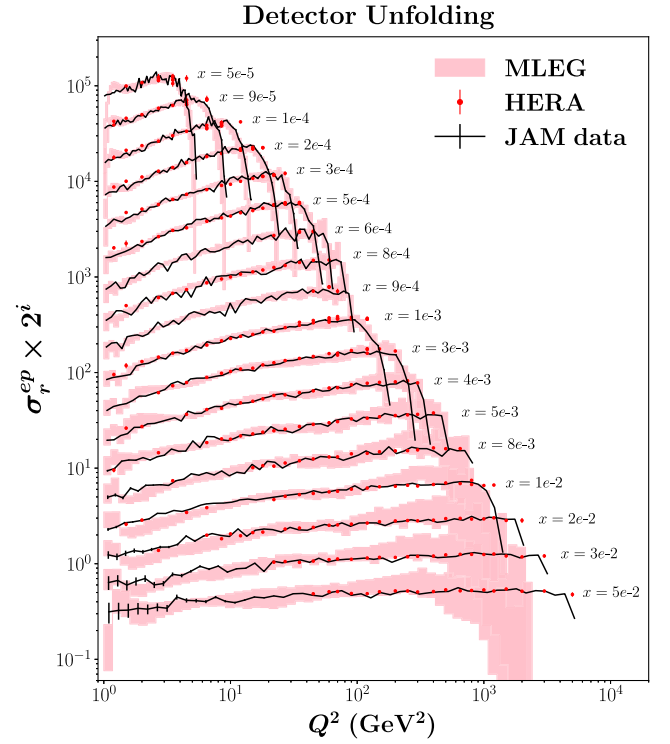

FIG. 8.　As in Fig. 4, but with the synthetic reduced cross sections generated by the GAN including detector effects and unfolding.

inferring this region can be traced back to the accuracy in reproducing ν_2 in Fig. 7 around $\nu_2 \sim 0$. To see this more clearly, in Fig. 9 we plot ν_2 as a function of Q^2 at various fixed values of x. As x increases, the range of ν_2 increases covering the negative ν_2 regions for lower values of Q^2. The observed deviations in Fig. 7 around $\nu_2 \sim 0$ can be understood by observing the corresponding regions in Fig. 6 where the detector-level distribution is compatible

FIG. 7.　As in Fig. 3, but with all the variables inferred by the unfolding procedure.
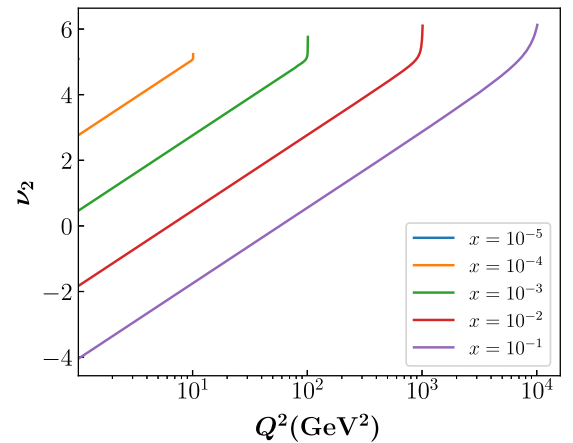
FIG. 9.　Kinematical relation between the ν_2 variable and Q^2 at several different values of x.

with zero. In this case, the inference of the underlying vertex-level distribution is ill defined.

We can understand such an effect by considering the extreme scenario where the detector does not observe a particle at all, or converts the vertex-level samples into flat noise. Clearly, in such a situation the vertex-level distribution is not recoverable. In the present situation, some regions of the phase space are subjected to some degree to such extreme effects, while other regions are not. Other than those ill-defined regions, one can see that within the uncertainties the synthetic reduced cross sections are in agreement with the true vertex-level cross sections. This can be seen as confirmation that our MLEG training passes the closure test in the presence of detector effects.

V. SUMMARY AND OUTLOOK

We have presented a new approach based on generative adversarial networks to extract physics observables from pseudodata in a physics agnostic manner. To illustrate the strategy, we developed a GAN-based MLEG capable of generating synthetic data that mimic inclusive deep-inelastic ep scattering pseudodata generated in the kinematics of the ZEUS and H1 experiments at HERA. To demonstrate the veracity of our approach we performed a closure test, extracting the original phase space distributions from synthetic particle four-momenta.

To simulate real experimental scenarios, we introduced distortions into the analysis that would be induced by a real detector, implementing a resolution smearing function, and after repeating the test obtained good agreement between original and extracted phase space distributions. Pulls quantified the uncertainty associated with the unfolding procedure, showing not only that we were able to extract the desired physics observables, but also obtain an uncertainty quantification for the unfolding procedure. To our knowledge, this is the first time that detector effects have been unfolded from pseudodata on an event basis.

While our long term goal remains to construct an MLEG for real experimental events across multiple channels involving multiple particles in the final state for QCD studies, the present analysis is a necessary and important proof of concept that demonstrates the viability of applying ML techniques to mitigate theoretical bias in experimental data analysis. Despite the fact that in our analysis we have effectively utilized only two-dimensional degrees of freedom to be reproduced by the MLEG, our main result is that it is possible to unfold detector effects at the event level. From the ML point of view, a larger number of particles in the final state amounts to a larger feature space. It is expected, therefore, that an extension of our proposed idea to include additional particles in the final state is feasible, provided that the number of final state particles remains moderate. This is the case, for example, in semi-inclusive and exclusive electron-nucleon scattering.

As an obvious improvement, and in view of its application to data analysis, we envision the implementation of a more realistic detector simulator based on GEANT to further study this technology. We expect that the use of our framework in ep scattering will be a valuable complementary tool for nuclear and particle physics programs at current and planned facilities, such as Jefferson Lab [37] and the Electron-Ion Collider [38].

ACKNOWLEDGMENTS

[1] T. Sjostrand, S. Mrenna, and P. Skands, Comput. Phys. Commun. **178**, 852 (2008).

[2] M. Bahr *et al.*, Eur. Phys. J. C **58**, 639 (2008).

[3] T. Gleisberg, S. Hoeche, F. Krauss, M. Schonherr, S. Schumann, F. Siegert, and J. Winter, J. High Energy Phys. 02 (2009) 007.

[4] P. Jimenez-Delgado, W. Melnitchouk, and J. F. Owens, J. Phys. G **40**, 093102 (2013).

[5] S. Forte and G. Watt, Annu. Rev. Nucl. Part. Sci. **63**, 291 (2013).

[6] R. D. Ball *et al.*, Eur. Phys. J. C **77**, 663 (2017).

[7] T. J. Hou *et al.*, Phys. Rev. D **103**, 014013 (2021).

[8] N. Sato, C. Andres, J. J. Ethier, and W. Melnitchouk, Phys. Rev. D **101**, 074020 (2020).

[9] S. Bailey, T. Cridge, L. A. Harland-Lang, A. D. Martin, and R. S. Thorne, Eur. Phys. J. C **81**, 341 (2021).

[10] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, Proceedings of the Conference and Workshop on Neural Information Processing Systems, (NIPS, 2014), https://papers.nips.cc/paper/2014/hash/5ca3e9b122f61f8f06494c97b1afccf3-Abstract.html.

[11] M. Paganini, L. de Oliveira, and B. Nachman, Phys. Rev. Lett. **120**, 042003 (2018).

[12] M. Paganini, L. de Oliveira, and B. Nachman, Phys. Rev. D **97**, 014021 (2018).

[13] L. de Oliveira, M. Paganini, and B. Nachman, Comput. Softw. Big Sci. **1**, 4 (2017).

[14] P. Musella and F. Pandolfi, Comput. Softw. Big Sci. **2**, 8 (2018).

[15] B. Hashemi, N. Amin, K. Datta, D. Olivito, and M. Pierini, arXiv:1901.05282.

[16] A. Butter, T. Plehn, and R. Winterhalder, SciPost Phys. **7**, 075 (2019).

[17] A. Gretton, K. Borgwardt, M. J. Rasch, B. Scholkopf, and A. J. Smola, Proceedings of the Conference and Workshop on Neural Information Processing Systems, 19 (NIPS, 2006), https://proceedings.neurips.cc/paper/2006/hash/e9fb2eda3d9c55a0d89c98d6c54b5b3e-Abstract.html.

[18] Y. Alanazi, N. Sato, P. Ambrozewicz, A. Hiller Blin, W. Melnitchouk, M. Battaglieri, T. Liu, and Y. Li, arXiv:2106.00643, https://www.ijcai.org/proceedings/2021/588.

[19] T. Karras, S. Laine, and T. Aila, arXiv:1812.04948, https://cvpr2019.thecvf.com/.

[20] O. Mogren (NIPS, 2016), http://www.cs.nott.ac.uk/~psztg/cml/2016/papers.html.

[21] A. Clark, J. Donahue, and K. Simonyan, arXiv:1907.06571.

[22] T. Salimans, I. J. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, arXiv:1606.03498.

[23] S. Arora and Y. Zhang, arXiv:1706.08224.

[24] M. Arjovsky and L. Bottou, arXiv:1701.04862.

[25] A. Srivastava et al., arXiv:1705.07761.

[26] D. Bang and H. Shim, arXiv:1804.04391v1.

[27] Software package for *eic-smear*: https://eic.github.io/software/eicsmear.html.

[28] R. Abdul Khalek et al., arXiv:2103.05419.

[29] S. Agostinelli et al., Nucl. Instrum. Methods Phys. Res., Sect. A **506**, 250 (2003).

[30] C. Cocuzza, C. E. Keppel, H. Liu, W. Melnitchouk, A. Metz, N. Sato, and A. W. Thomas, Phys. Rev. Lett. **127**, 242001 (2021).

[31] Y. Alanazi, N. Sato, T. Liu, W. Melnitchouk, P. Ambrozewicz, F. Hauenstein, M. P. Kuchera, E. Pritchard, M. Robertson, R. Strauss, L. Velasco, and Y. Li, arXiv:2001.11103, https://www.ijcai.org/proceedings/2021/293.

[32] M. Mirza and S. Osindero, arXiv:1411.1784.

[33] M. Bellagente, A. Butter, G. Kasieczka, T. Plehn, and R. Winterhalder, SciPost Phys. **8**, 070 (2020).

[34] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley, Least squares Generative adversarial networks, *2017 IEEE International Conference on Computer Vision* (ICCV, Venice, Italy, 2017), p. 2813, https://openaccess.thecvf.com/content_iccv_2017/html/Mao_Least_Squares_Generative_ICCV_2017_paper.html.

[35] D. P. Kingma and J. Ba, International Conference for Learning Representations, San Diego (2015), arXiv:1412.6980.

[36] H. Abramowicz et al., Eur. Phys. J. C **75**, 580 (2015).

[37] V. D. Burkert, Annu. Rev. Nucl. Part. Sci. **68**, 405 (2018).

[38] A. Accardi et al., Eur. Phys. J. A **52**, 268 (2016).