End-to-end physics event generator

Y. Alanazi[1], M. P. Kuchera[2], Y. Li (co-PI)[1], T. Liu[4], R. E. McClellan[4], W. Melnitchouk (PI)[4], E. Pritchard[2], R. Ramanujan[2], M. Robertson[2], N. Sato (co-PI)[4], R. R. Strauss[2], L. Velasco[3]

[1] Department of Computer Science, Old Dominion University, Norfolk, Virginia 23529
[2] Department of Physics, Davidson College, Davidson, North Carolina 28035
[3] Department of Physics, University of Dallas, Irving, Texas 75062
[4] Theory Center, Jefferson Lab, Newport News, Virginia 23606
✉ Correspondence to: yalan001@odu.edu

Introduction

In high-energy physics, particle accelerators are built to gain insight into elementary particles by colliding them onto a nuclear target or against other particles. The reactions from the collision transform the incoming particles into set of outgoing particles, known as physics "events". Existing high-energy accelerators include the LHC at CERN for proton-proton collisions, CEBAF at Jefferson Lab for po-larized electron-hadron scattering, as well as the future Electron-Ion Collider (EIC) [6].

Overview

The goal of this project is to develop a machine learning event generator (MLEG) that can faithfully reproduce particle events at the vertex level that is free of theoretical assumptions about underlying particle dynamics. By training the model at the event level using the reconstructed detector-level particle four-vectors, the model can capture all the relevant particle correlations without the need to *a priori* specify particular observables to study. The MLEG can thus be viewed as a compactified data storage utility that can provide future access to observables not conceived of at the time of the original experiments. The successfully trained MLEG will be a valuable software tool for phenomenological studies at JLab12 and beyond, providing a unique avenue for quantitatively testing the validity of theoretical approximations implemented in QCD factorization theorems.

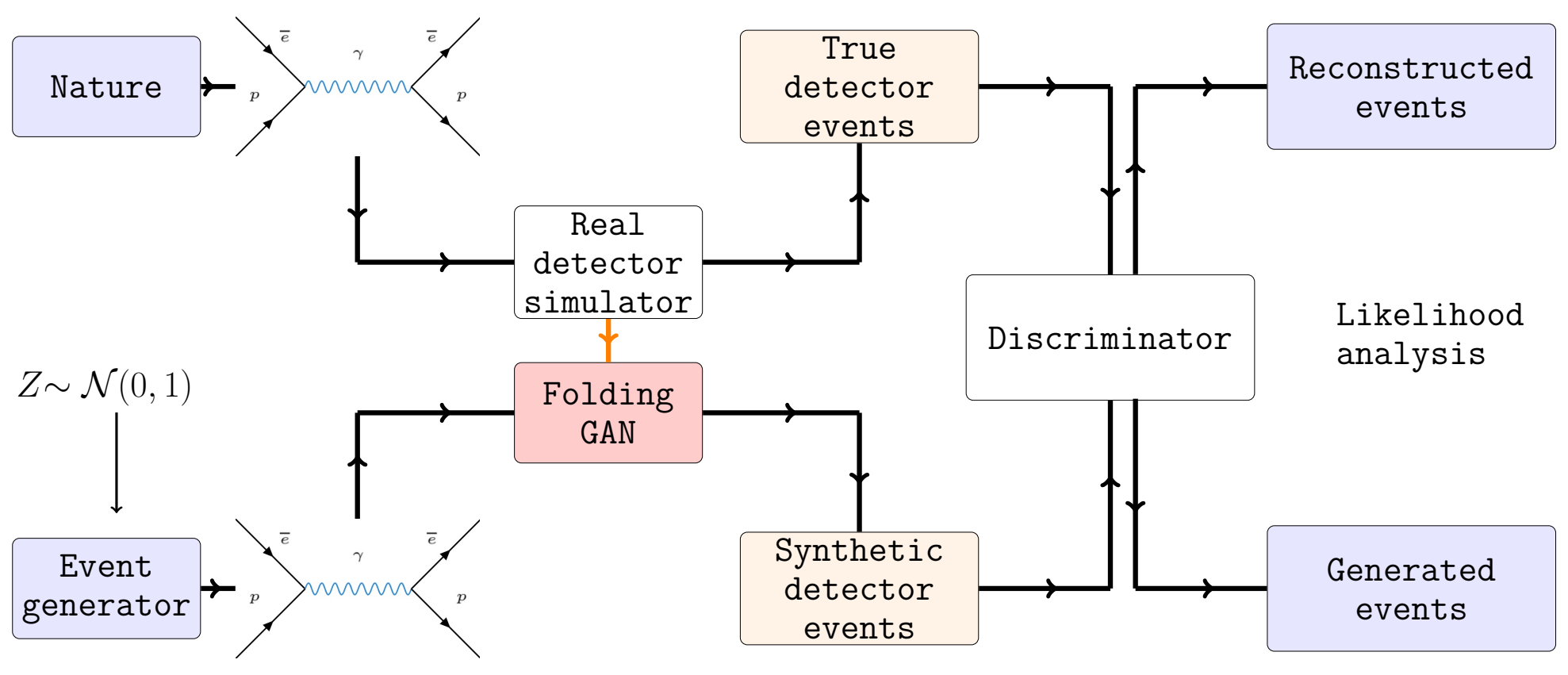

Fig. 1: Conceptual design of the project. The diagram shows the overview of the project. The (event generator) simulates the electron-proton scattering as in (Nature). The Folding GAN uses a real detector simulator information to fold the synthetic events which are passed to the the discriminator model to be compared and analysed against the reconstructed events

Methods

During the initial phase of the project, we have successfully developed an event generator that can simulates particle events.

- We use generative adversarial networks (GANs) [1] to build the MLEG. Since the GAN is trained at the event level, the major challenge has been to identify a suitable data representation.
- We incorporate maximum mean discrepancy kernel test [7] to increase the precision of the generated distributions.
- We transform the features to address the sharp peaks exist in the dataset. For example, instead of directly using p_z as a generated feature, we use the transformed variable $\mathcal{T}(p_z) = \log(E_{\rm b} - p_z)$.
- We enhance the training by augmenting new features to increase the sensitivity of the discriminator. For instance, we use a customized layer to calculate the momentum energy which than passed to the discriminator. The energy E can be calculated as $E = \sqrt{px*px+py*py+pz*pz}$.

Results

To validate the MLEG, we first train on synthetic events generated from Pythia 8 [2]. Our results [8] show a good agreement with the true distribution (Pythia). We also see as in Fig. 2 the model can capture the underlying correlations between the features.

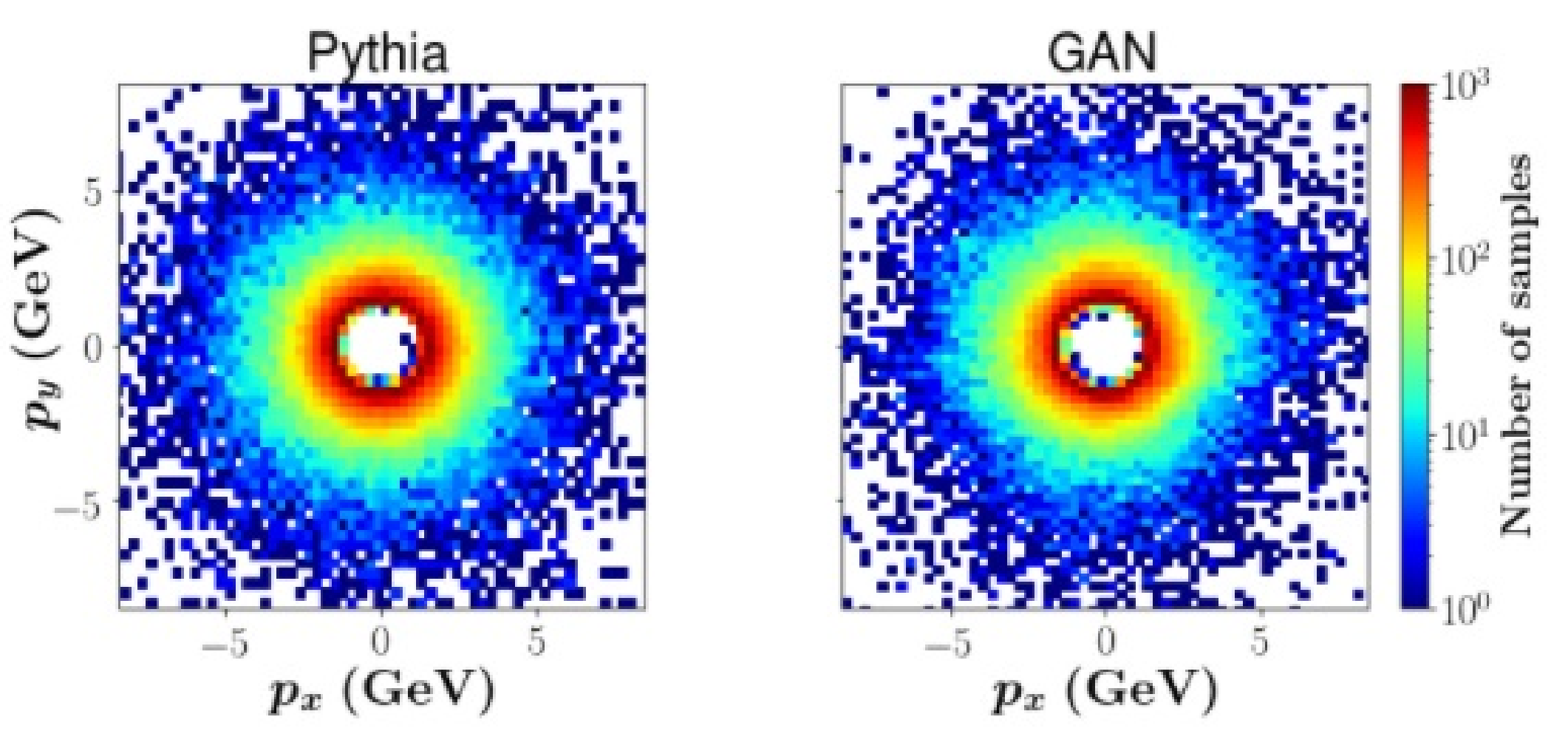

Fig. 2: 2-dimensional histogram of p_x and p_y The correlation histogram shows our model on the right side has captured the inter-correlations between p_x and p_y

For the generation of the inclusive lepton MC events, we have trained a GAN with the architecture shown in Fig. 3. After training on $\sim 10^5$ inclusive electron samples from Pythia, the GAN is able to reproduce relatively well the scattered electron phase space, as illustrated in Figs. 4–7 for the normalized yields versus Bjorken $x_{\rm bj}$, the four-momentum transfer squared Q^2, and the scattered lepton transverse momentum $p_{\rm T}$. The uncertainties on the GAN generated events are obtained by training the GAN multiple times using the bootstrapping method.

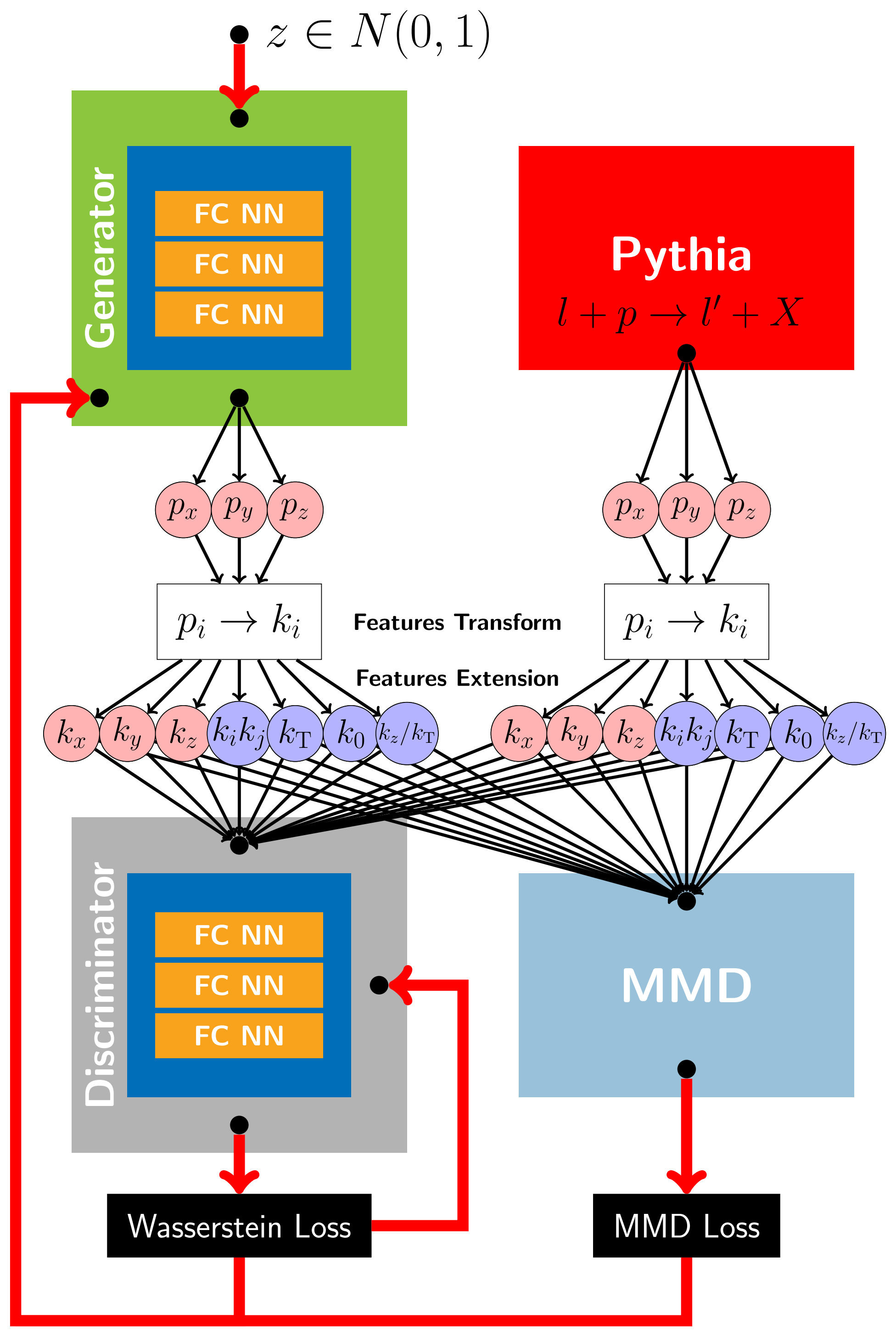

Fig. 3: The GAN setup. The original architecture proposed in [3], where the MMD [4] was used as the loss function of the GAN's generator, is extended to include the Wasserstein loss function [5] for the discriminator, and a weighted function combining Wasserstein and MMD loss for the generator. Novel feature transformation algorithms ensure four-momentum conservation in the generator, and significantly enhance the effectiveness of the discriminator. The GAN training took ≈12 hours on an Nvidia GeForce 2080Ti GPU.

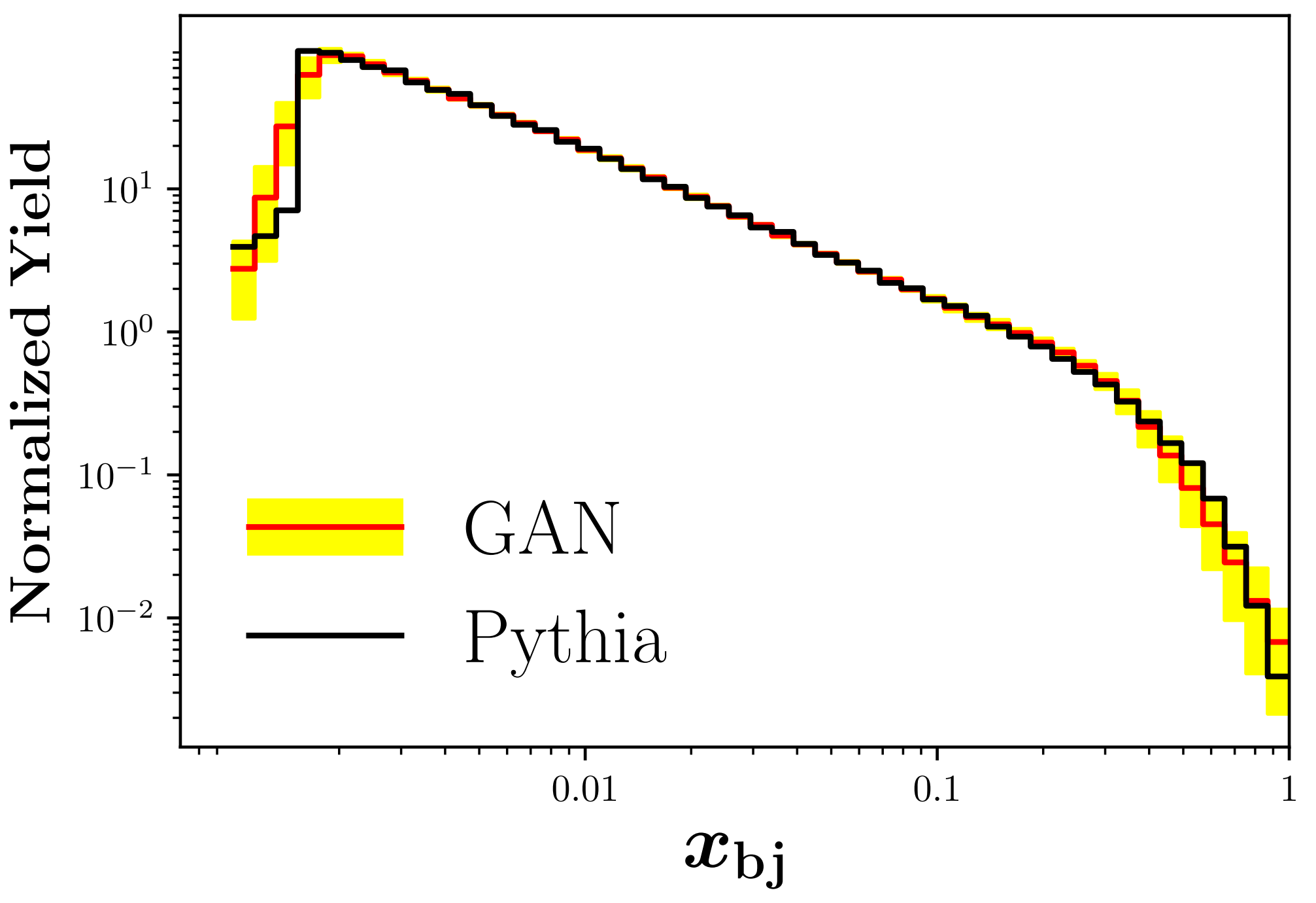

Fig. 4: Normalized $x_{\rm bj}$ distributions, generated from the GAN and Pythia 8. Note that $x_{\rm bj}$ has not been included as a feature in the GAN training.

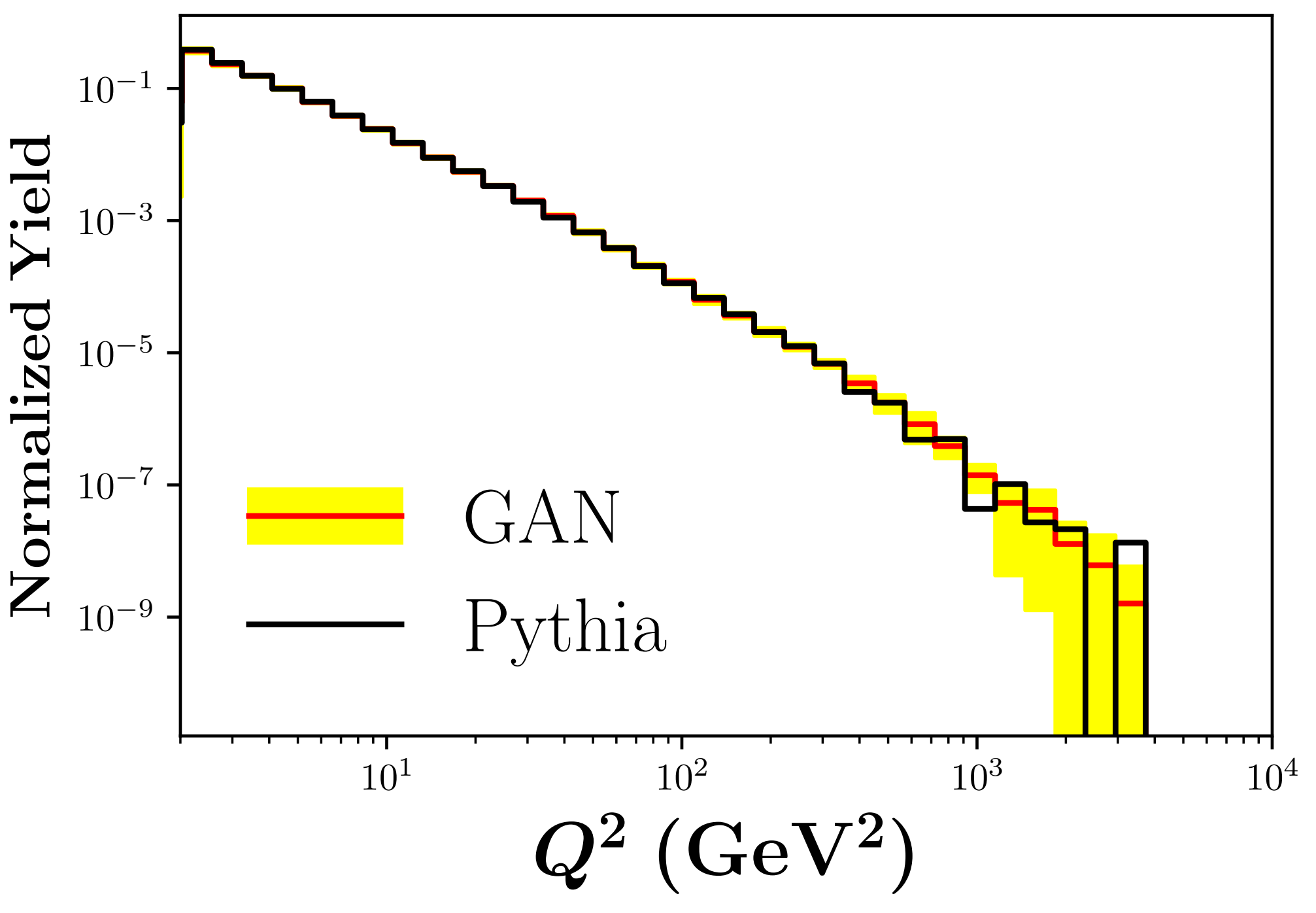

Fig. 5: Normalized Q^2 distributions. As in Fig. 4, the photon virtuality Q^2 has not been included as a feature in the GAN traning, indicating that the NNs can generate derived quantities from four-vectors.

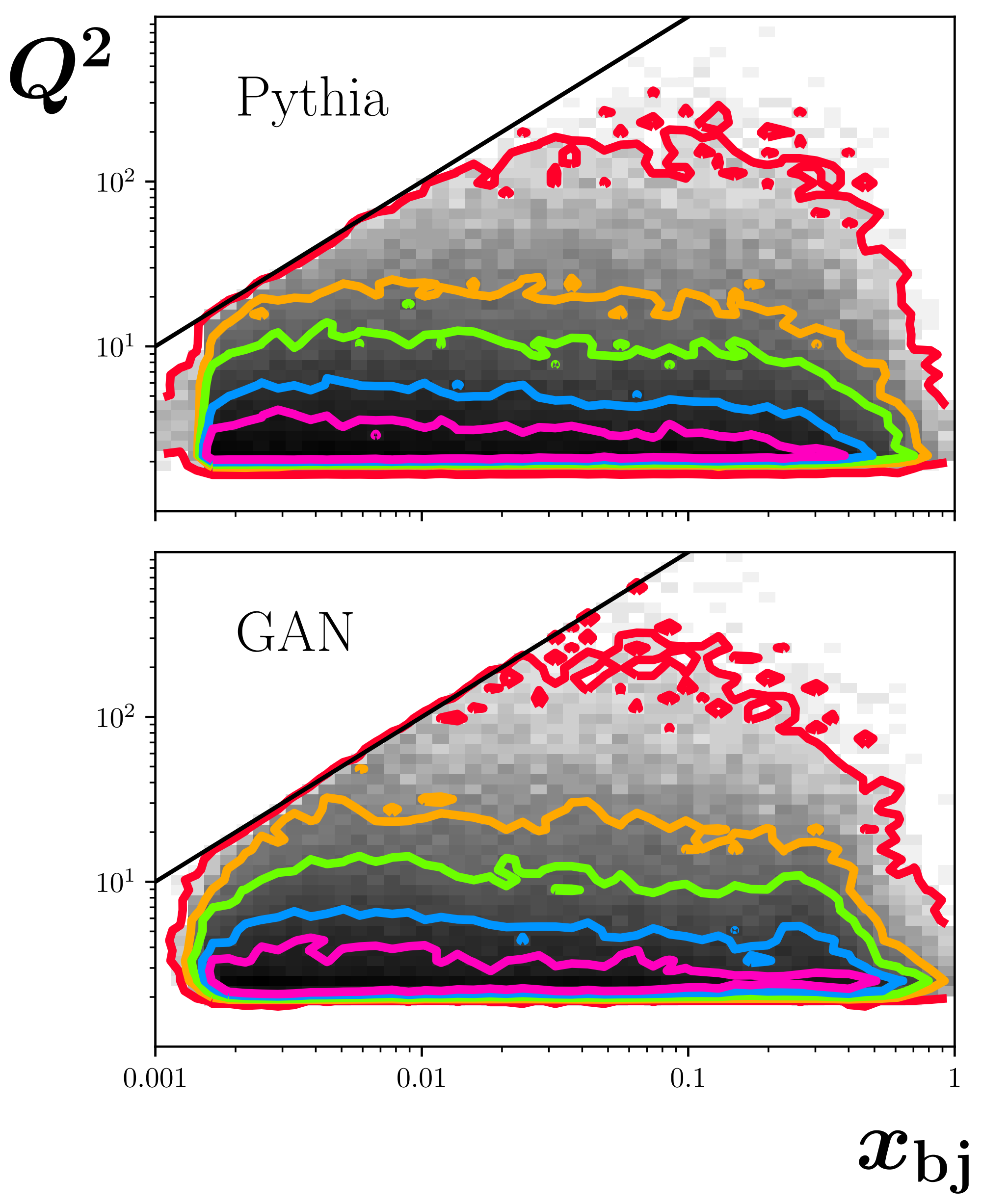

Fig. 6: $x_{\rm bj}$ – Q^2 correlation. The similarity between the iso-contours indicates that the NNs can accurately learn the correlations between $x_{\rm bj}$ and Q^2.

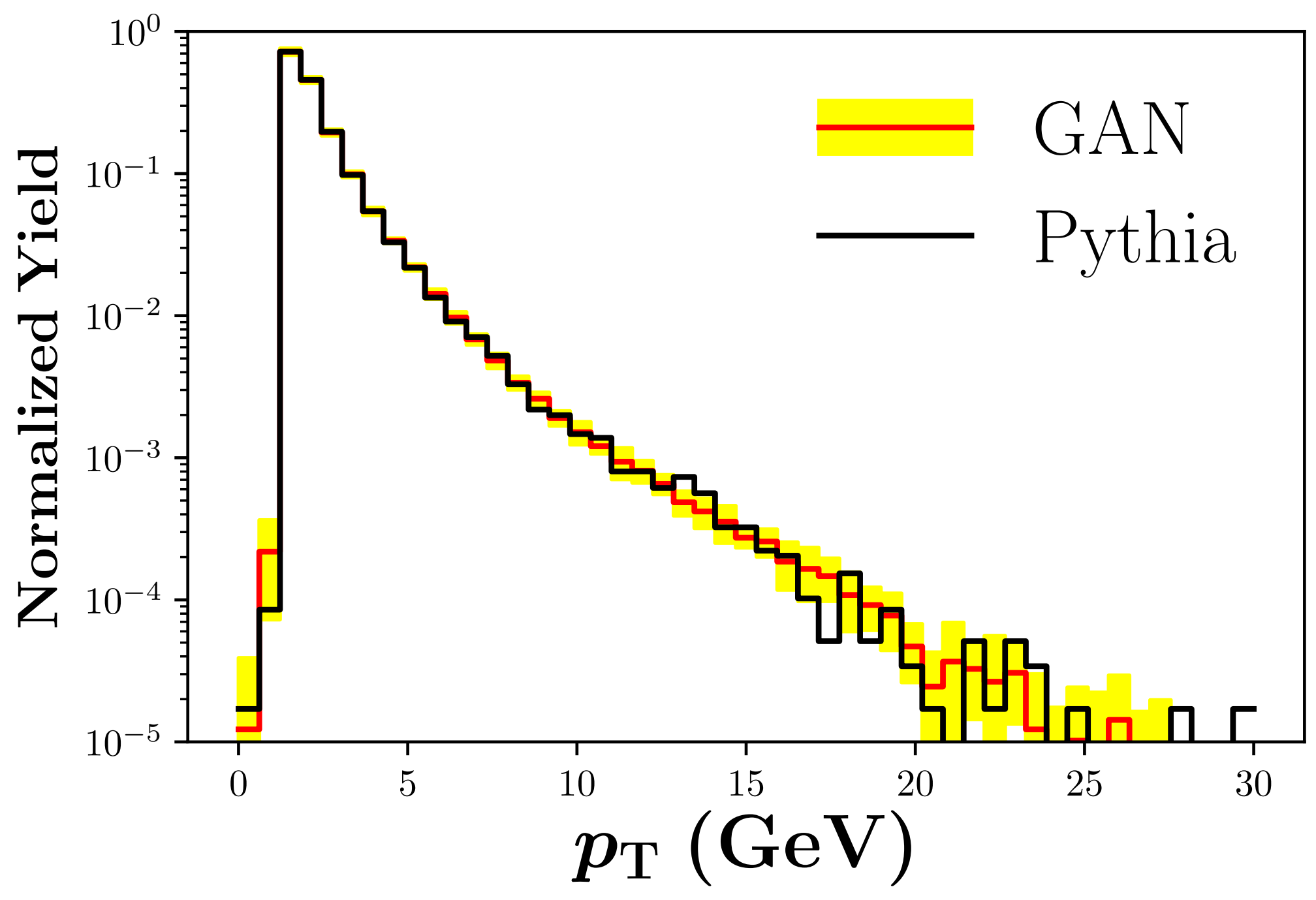

Fig. 7: Normalized scattered lepton $p_{\rm T}$ distribution. Unlike for $x_{\rm bj}$ and Q^2 in Figs. 4 and 5, the transverse momentum $p_{\rm T}$ has been included as feature in the GAN training. The NNs are able to learn the structure across ~ 5 orders of magnitude of the distribution.

Conclusions

We have explored the MLEG model in building the first event generator for electron-proton scattering that faithfully mimics particle generation at femtometer scales, without the need for theoretical input. Our MLEG design involves a GAN that generates particle multiplicities followed by a GAN that generates particle four-vectors conditioned to the outcome of the first generator.

Acknowledgments

We thank J. Qiu for helpful discussions. This work was supported by the LDRD project No. LDRD19-13.

References

[1] I. J. Goodfellow *et al.*, NIPS, pp. 2672–2680, **2014**.
[2] T. Sjöstrand *et al.*, Comput. Phys. Commun. **178**, 852 (2008).
[3] A. Butter, T. Plehn, R. Winterhalder, arXiv:1907.03764.
[4] C. Li *et al.*, NIPS, pp. 2203–2213, **2017**.
[5] I. Gulrajani *et al.*, NIPS, pp. 5767–5777, **2017**.
[6] Yasir Alanazi, N. Sato, *et al.* "A Survey of Machine Learning based Physics Event Generation." In: ijcai2021. Submitted (2021).
[7] Alexander Smola. "A Kernel Two-Sample Test".
[8] Yasir Alanazi, N. Sato, *et al.* "Simulation of electron-proton scattering events by a Feature-Augmented and Transformed Generative Adversarial Network (FAT-GAN)" arXiv:2001.11103. In: ijcai2021. Submitted (2021).