

# Automated detector simulation and reconstruction parametrization using machine learning

D. Benjamin,<sup>1</sup> S. Chekanov,<sup>1</sup> W. Hopkins,<sup>1</sup> Y. Li,<sup>2</sup> and J. R. Love<sup>1</sup>

<sup>1</sup>*High Energy Physics Division, Argonne National Laboratory,  
9700 S. Cass Avenue, Argonne, IL 60439, USA*

<sup>2</sup>*Computational Science Division, Argonne National Laboratory,  
9700 S. Cass Avenue, Argonne, IL 60439, USA*

(Dated: March 23, 2020)

## Abstract

Rapidly applying the effects of detector response to physics objects (e.g. electrons, muons, showers of particles) is essential in high energy physics. Currently available tools for the transformation from truth-level physics objects to reconstructed detector-level physics objects involve manually defining resolution functions. These resolution functions are typically derived in bins of variables that are correlated with the resolution (e.g. pseudorapidity and transverse momentum). This process is time consuming, requires manual updates when detector conditions change, and can miss important correlations. Machine learning offers a way to automate the process of building these truth-to-reconstructed object transformations and can capture complex correlation for any given set of input variables. Such machine learning algorithms, with sufficient optimization, could have a wide range of applications: improving phenomenological studies by using a better detector representation, allowing for more efficient production of Geant4 simulation by only simulating events within an interesting part of phase space, and studies on future experimental sensitivity to new physics.

## 9 I. INTRODUCTION

10 A cornerstone of particle collision experiments is the Monte Carlo (MC) simulation of  
11 physics processes resulting from collisions of high-energy particles, followed by the simulation  
12 of detector responses and object reconstruction. The MC simulation produces objects (jets,  
13 electrons, muons, etc) with properties (four momenta, particle types) which entirely depend  
14 on the physics processes occurring. These objects are commonly referred to as “truth”  
15 objects. These objects are altered by interactions with the detector and are reconstructed  
16 with experimental algorithms. Such objects, that have undergone a transformation due to  
17 detector interactions and reconstruction will be referred to as “reco” objects in this paper.

18 With the increased complexity of high-energy collider experiments, such as those at the  
19 Large Hadron Collider (LHC), the detector simulations become increasing complex and time  
20 consuming. Parameterized detector simulations, such as Delphes [1], have been proven to be  
21 a vital tool for physics performance and phenomenological studies (i.e. to estimate the sensitiv-  
22 ity of an experiment to a new physics model). Producing an approximation of the detector  
23 in Delphes is a laborious process as developers must manually define resolution functions  
24 based on an experiment’s performance studies. An approximation of the detector responses  
25 and experimental object reconstruction can, however, also be performed with machine learn-  
26 ing (ML) by training a neural network (NN) using the Geant4-based simulations that have  
27 gone through an experiment’s reconstruction algorithm. Such an NN could then automat-  
28 ically learn the transformation functions from truth MC objects (jets and other identified  
29 particles) to objects modified by a detector and experimental reconstruction algorithms.

30 The main advantage of a detector parametrization based on ML, as compared to a  
31 manually-constructed analytic parametrization such as Delphes, is that a neural network  
32 can automatically learn the features introduced by detailed full simulations avoiding the  
33 need to handcraft parameters to represent resolutions and inefficiencies. An NN trained  
34 using realistic detector simulation could learn the transformation from the truth to the reco  
35 quantities without dedicated studies of resolution functions and could automatically be up-  
36 dated when there are changes to the detector or the reconstructions algorithms. Another  
37 advantage is that the NN approach can introduce a complex interdependence of variables  
38 which is currently difficult to implement in parameterized simulations.

39 The organization of this paper is as follows. In Section II, the traditional method of

parametrizing detector and reconstruction effects are described. Section III contains a description of the samples used to train the NN. The details of the NNs structure and parameters are reported in Sections IV. Finally, the results from the trained NN are presented in Section V.

## II. TRADITIONAL PARAMETERIZED FAST SIMULATIONS

In abstract terms, a typical variable  $\xi_i^{\text{reco}}$  that characterizes a reconstructed particle/jet, such as transverse momentum ( $p_T^{\text{reco}}$ ) or pseudorapidity ( $\eta^{\text{reco}}$ ), can be viewed as the result of a multivariate transform,  $F$ , of the original variable  $\xi_1^{\text{truth}}$  at truth level:

$$\xi_1^{\text{reco}} = F(\xi_1^{\text{truth}}, \xi_2^{\text{truth}}, \xi_3^{\text{truth}}, \dots, \xi_N^{\text{truth}}).$$

Generally, such a transform depends on several other variables  $\xi_2^{\text{truth}} \dots \xi_N^{\text{truth}}$  characterizing this (or other) objects at truth level. For example, the extent at which jet transverse momentum,  $p_T$  is modified by a detector depends on the original truth-level transverse momentum ( $\xi_1^{\text{truth}} = p_T^{\text{truth}}$ ), pseudorapidity ( $\xi_2^{\text{truth}} = \eta^{\text{truth}}$ ), and other effects that can be inferred from truth quantities. Similarly, if particular detector regions in the azimuthal angle ( $\phi$ ) have low efficiency, this would introduce an additional dependence of this transform on  $\phi$ .

Typical parameterized simulations ignore the full range of correlations between the truth-level variables. In most cases, the above transform is reduced to a single variable, or two (as in the case of Delphes simulations where the energy resolution of clusters depends on the original energies of particles and their positions in  $\eta$ ). In order to take into account correlations between multiple parameters characterizing transformations to reconstruction objects, a grid in the hypercube with the dimension  $N_b^k$ , where  $k=1,2,3,\dots$  and  $N_b$  is the number of histogram bins for the distributions  $(\xi^{\text{reco}} - \xi^{\text{truth}})/\xi^{\text{truth}}$ , representing the “resolution”, must be created. This methodology results in a large number of histograms when there are many correlated variables that affect the resolution.

It should be pointed out that the calculation speed for parameterized simulations of one variable that depends on  $N$  other variables at the truth level is proportional to  $N_b^k$  since each object at the truth level should be placed inside the grid defined by  $N_b$  bins. Therefore, complex parameterisations of resolutions and efficiencies for  $N > 2$  becomes computationally

68 intensive.

### 69 III. MONTE CARLO SIMULATED EVENT SAMPLES

70 Monte Carlo events used for this analysis were produced using the Madgraph generator [2].  
71 The simulated processes are a combination of equal event samples with top pair production  
72 ( $t\bar{t}$ ) and photons produced in association with jets ( $\gamma$ +jets), which give a high rate of jets  
73 in different environments. Hadronic jets were reconstructed with the FASTJET package [3]  
74 using the anti- $k_t$  algorithm [4] with a distance parameter of 0.4. The detector simulation  
75 was performed with the Delphes package with a detector geometry which is similar to the  
76 ATLAS geometry. The event samples used for the following study are available from the  
77 HepSim database [5]. In this paper, only the transformation of  $p_T$  from truth jets (which have  
78 truth particle constituents) to reconstructed jets (which have calorimeter cell constituent)  
79 was performed. However, the methodology should be object and parameter agnostic. Only  
80 truth jets which are matched to a reconstructed Delphes jet are considered in this study.  
81 For the matching criteria the reconstructed jet that has the smallest  $\Delta R \equiv \sqrt{\Delta\phi^2 + \Delta\eta^2}$ ,  
82 where  $\Delta\phi \equiv \phi^{\text{truth}} - \phi^{\text{reco}}$  and  $\Delta\eta \equiv \eta^{\text{truth}} - \eta^{\text{reco}}$ , with respect to the truth jet is chosen. If  
83 this minimum  $\Delta R$  is greater than 0.2, the truth jet is discarded. No other requirements are  
84 made on truth and reconstructed Delphes jets other than the  $p_T > 15$  GeV requirement in  
85 Delphes. Only matched jets are used since the aim of the study is to test whether an NN  
86 can learn changes in detector resolution as a function of kinematic properties of the jet (e.g.  
87  $p_T, \eta, \phi, m$ ). The final number of training jets used is two million while 500,000 jets were  
88 used as an independent test sample. The distributions of quantities used as the input for  
89 the NN,  $p_T, \eta, \phi, m$ , are shown in Figure 1.

90 To facilitate training and to smooth the hypersurface formed by the input variables, the  
91 input variables are scaled to be in the range [0,1]. This avoids the  $p_T$  and the mass from  
92 having a disproportional affect on the training of the NN. The output variable,  $(p_T^{\text{reco}} -$   
93  $p_T^{\text{truth}})/p_T^{\text{truth}}$ , is also scaled to have values between 0 and 1. Only jets that are within the 1<sup>st</sup>  
94 and 99<sup>th</sup> percentile of the  $(p_T^{\text{reco}} - p_T^{\text{truth}})/p_T^{\text{truth}}$  distribution are considered.

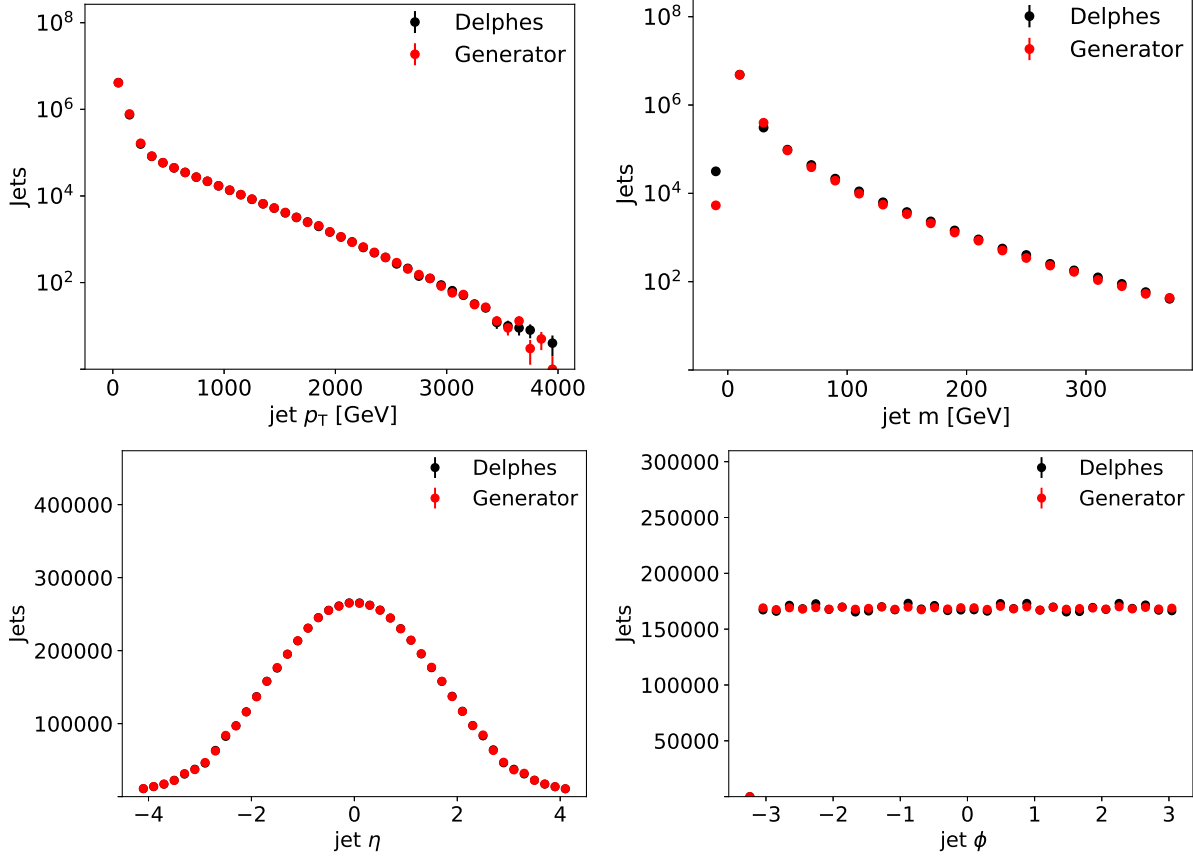


FIG. 1. Distributions for input variables for truth (red) and reco quantities (black).

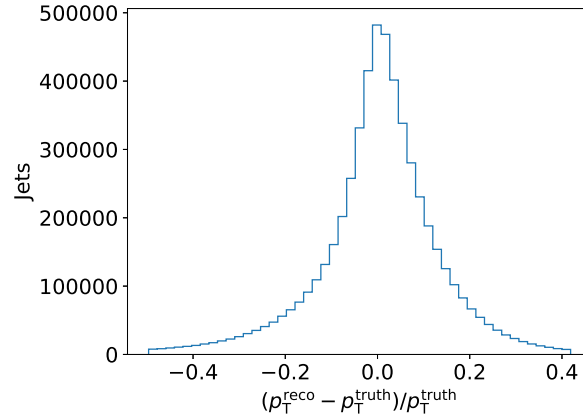


FIG. 2. Resolution of the  $p_T$  (relative differences between truth and reco  $p_T$ ).

#### 95 IV. JET TRUTH-TO-RECO TRANSFORMATION WITH ML

96 To test the viability of using ML to transform truth objects to reco objects, we studied  
 97 the truth-to-reco transformation for jets. Jet truth-level quantities, such as jet  $p_T$ ,  $\eta$ ,  $\phi$  and

jet mass ( $m$ ) are used as training inputs to an NN while the output is an array of nodes that represent the binned probability density function (PDF) of the resolution for a single variable (such as jet  $p_T$ ). Additional input can consist of any variable that can influence the resolution of a jet, such as jet flavor at the truth level, jet radius, etc. Figure 3 shows a schematic representation of the NN architecture for modelling the detector response for a single output variable. The aim is to have the NN learn the shape of the resolution PDF, for example for the  $p_T$ , depending on other input variables such as the  $\eta$  of the object. A binned output (multi-categorization) was used so that the precision of the resolution PDF modelling can be chosen.

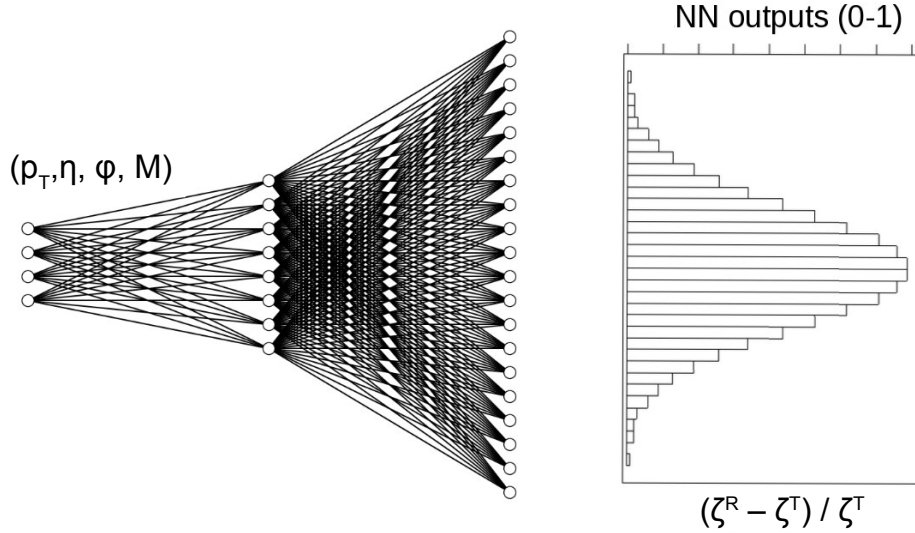


FIG. 3. A schematic representation of the NN architecture for modelling the detector response and affect of reconstruction algorithms on truth-level input variables. The output nodes of this NN represent a binned PDF for the resolution of single variable, e.g.  $(p_T^{\text{reco}} - p_T^{\text{truth}})/p_T^{\text{truth}}$ , while other variables such  $\eta$ ,  $\phi$ , mass ( $m$ ) are auxiliary variables that affect  $(p_T^{\text{reco}} - p_T^{\text{truth}})/p_T^{\text{truth}}$ .

As a first step towards parameterized detector simulations with ML, it is instructive to investigate how a transformation from the truth to reco objects can be performed, leaving aside the question of introducing objects that are created by misreconstructions or objects that are lost due to inefficiencies. As far as the authors are aware, this is the first attempt at using an NN to learn the transformation of truth physics objects into reco objects.

An NN is trained with four input parameters, the scaled jet  $p_T$ ,  $\eta$ ,  $\phi$ , and  $m$ , and consist of five fully connected layers with 100 nodes each, with each node having a rectifier linear

unit (ReLU) activation function. The output layer has 400 nodes and a softmax activation function. The NN is trained using the Adam [6] optimizer with a learning rate of  $10^{-4}$  and is implemented using Keras with a TensorFlow [7] backend. Finally, categorical cross-entropy is used as the loss function.

To optimize the NN, several output layer configurations are tested by varying the number of output nodes from 50 to 500, all with a softmax activation function. The configuration with 400 output nodes resulted in the best performance, measured by how well the NN could mimic the Delphes  $p_T$  spectrum and resolution (see below for details), with the least number of total NN parameters. The figure of merit for choosing the number of output nodes was the maximal deviation of the ratio between the  $p_T$  spectrum produced by the NN and the  $p_T$  spectrum produced by Delphes (see the bottom panel of Figure ?? for the final ratios). The configuration with the smallest maximal deviation with the least number of nodes was chosen (the deviation didn't improve when going to 500 nodes).

Additionally, for the NN training, several batch-size and number-of-epoch combinations are probed to test whether these parameters cause a change in sensitivity to small subsamples (specifically, forward jets) of the training sample. Changing the batch size without modifying the number of epochs would result in a change in the total number updates to the NN parameters during back propagation. To avoid this, the number of backpropagations ( $N_{bp}$ ) are held constant by keeping the ratio of the number of epochs ( $N_e$ ) and batch size ( $N_b$ ) constant since  $N_{bp} = \frac{N_t}{N_b} N_e$  where  $N_t$  is the number of training jets. The performance of the resulting NNs is evaluated by comparing the Kolmogorov-Smirnov (KS) test statistic for the  $(p_T^{\text{reco}} - p_T^{\text{truth}})/p_T^{\text{truth}}$  of forward jets in Delphes and for the NN-produced jets. Batch size and number of epochs of 5, 10, 20, 100, 200, 1000 are tested resulting negligible differences in the KS test statistic for the different NNs.

Finally, the loss function for both the training (2 million jets) and test sample (500,000 jets) is evaluated for each epoch. The values of the loss function for the training and testing sample are found to be within 1% of each other at the end of training.

## V. RESULTS

After the NN has been trained to learn the PDF of  $(p_T^{\text{reco}} - p_T^{\text{truth}})/p_T^{\text{truth}}$ , the resulting learned PDF is compared to the Delphes PDF using the test sample in Figure 4a. Good

144 agreement is observed between the Delphes and NN PDFs, showing that the NN has learned  
 145 the bulk distribution.

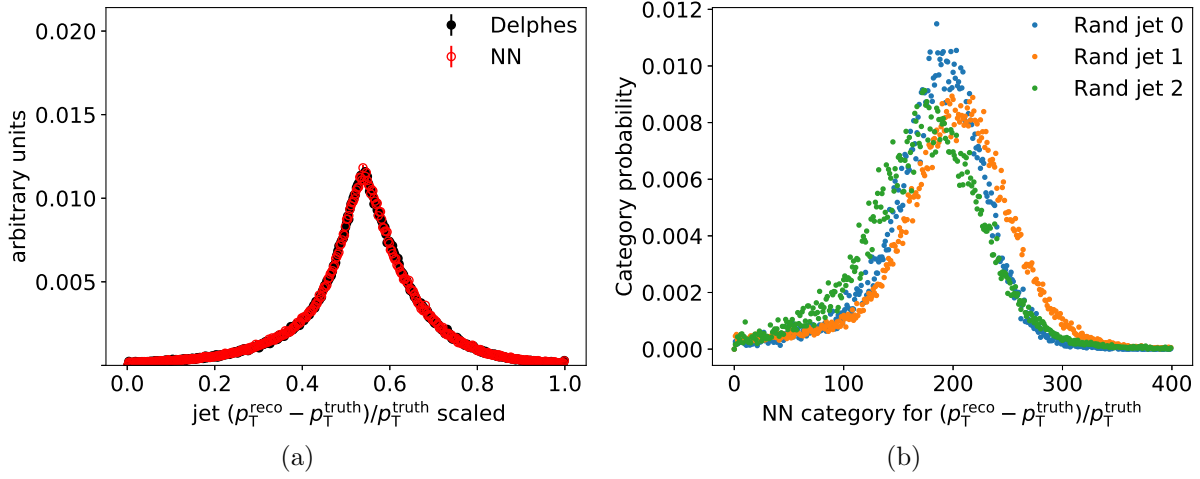


FIG. 4. NN-generated jet  $(p_T^{\text{reco}} - p_T^{\text{truth}})/p_T^{\text{truth}}$  compared to Delphes reco jet  $(p_T^{\text{reco}} - p_T^{\text{truth}})/p_T^{\text{truth}}$   
 (a). Representative values of the NN output after training for three randomly selected truth jets  
 which have different input values (b).

146 The NN output represents a binned PDF for each jet based on its input parameters (i.e.  
 147  $p_T$ ,  $\phi$ ,  $\eta$ , and  $m$ ). The PDFs for a set of three randomly selected jets are shown in Figure 4b  
 148 which features shapes expected for typical resolution function with variations due to changes  
 149 in jet input parameters. These PDFs are then randomly sampled to produce an NN jet that  
 150 mimics the reco jet. A comparison of the NN-generated and Delphes jet  $p_T$  distribution for  
 151 the test sample is shown in Figure 5. The NN reproduces the jet  $p_T$  distribution of Delphes  
 152 within 5% for reconstructed jets with  $p_T > 20$  GeV.

153 To test whether the NN learned correlations between input parameters and the  $p_T$  res-  
 154 olution, the jets were divided into central ( $|\eta| < 3.2$ ) and forward ( $|\eta| > 3.2$ ) jets. The  $p_T$   
 155 resolution is then compared between the two regions for both the Delphes jets as well as the  
 156 NN-generated jets. These two regions in the detector simulation have different calorimeter  
 157 responses which results in different jet  $p_T$  resolutions in these two  $|\eta|$  regions. The resulting  
 158 resolutions for both regions are shown in Figure 6 using the training sample. The training  
 159 sample was chosen for this comparison because forward jets make up a small subsample of  
 160 all jets, as can be seen in Figure 1.

161 The mean and standard deviation of the resolution (demonstrated in Figure 4) as a



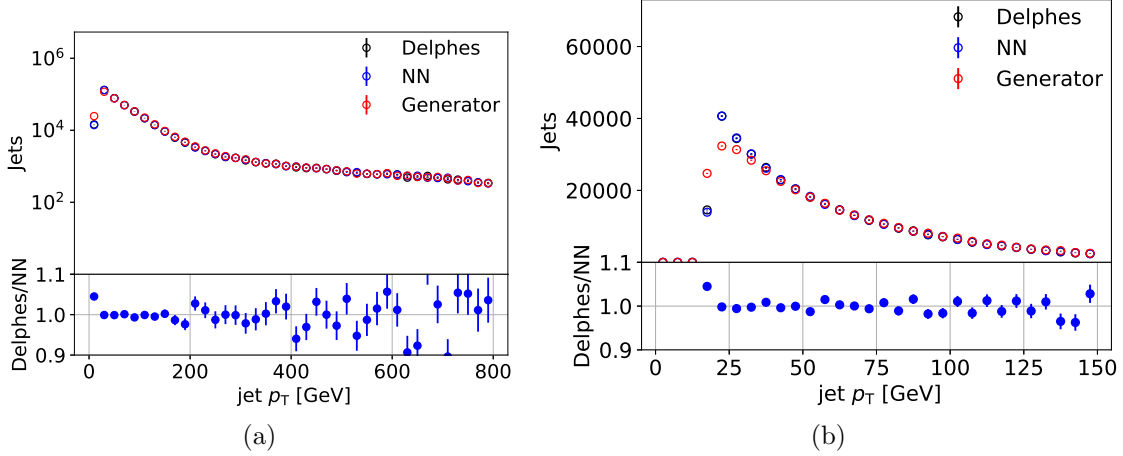


FIG. 5. Delphes and NN-generated jet  $p_T$  distributions for a wide (a) and narrow (b)  $p_T$  range.

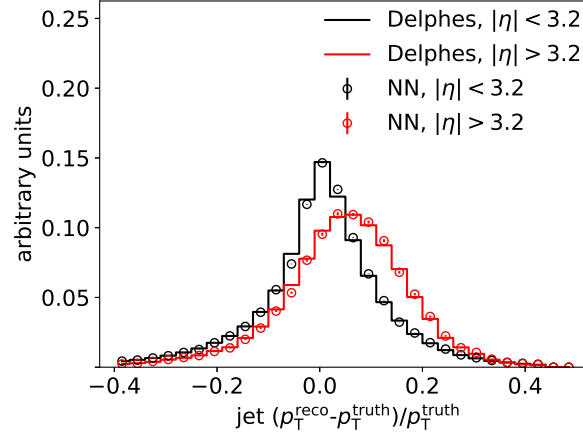


FIG. 6. Jet  $p_T$  resolution for the training sample for both the central and forward region.

function of  $p_T$  is shown in Figure 7. The mean of the resolution for the NN is systematically higher than the resolution for Delphes but this effect is small when considering the width of the resolution. The standard deviation of the resolutions, however, are the same for the NN and Delphes across the  $p_T$  range showing that the NN accurately predicts the resolutions for a large range in  $p_T$ .

In order to produce an NN with optimal performance and a minimal amount of parameters, a genetic algorithm (GA) is used to optimize the NN hyperparameters. The number of layers, number of neurons in each layer, and choice of the learning rate are scanned to find the optimal configuration. The GA we utilize is an evolutionary algorithm that mimics the process of natural selection and which was previously used in the determination of the

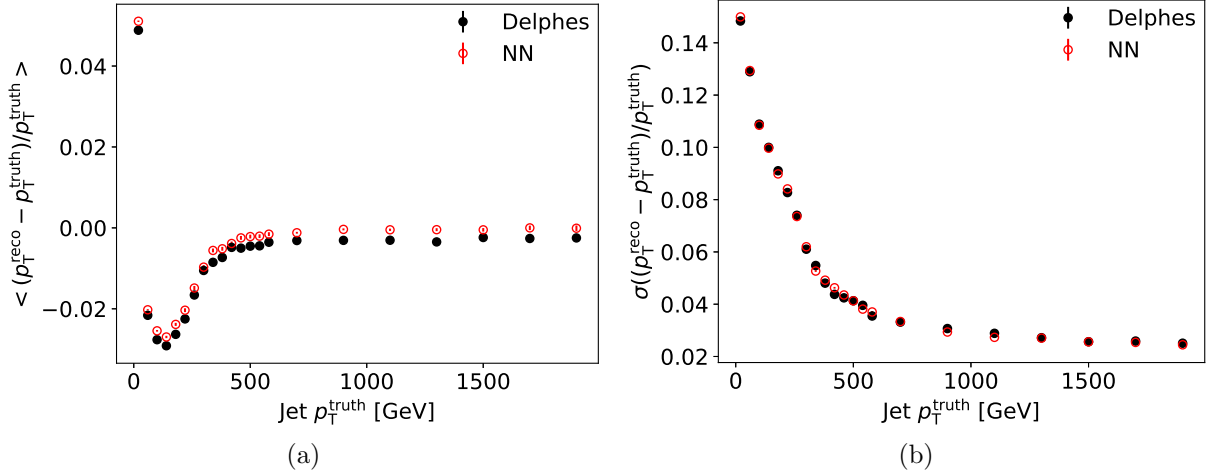


FIG. 7. The mean (a) and standard deviation of the jet  $p_T$  resolution for Delphes and NN-generated as a function truth jet  $p_T$ .

parameters of a complex force field [8, 9]. Due to the small scale of our problem, the initial configuration was found to be equivalent to the optimal configuration found by the GA.

## VI. CONCLUSION

A truth-level to reconstruction-level transformation using an NN classifier is presented. This approach does not require the determination of analytic resolution functions since an NN can automatically learn the resolutions during the training procedure. The jet  $p_T$  distribution produced by the NN is within 5% of the  $p_T$  distribution from Delphes for jets with  $15 < p_T < 400$  GeV. The NN also effectively learned the truth-to-reconstruction transformation without requiring manual binning to capture the differences in resolutions of particular subsamples (i.e. central and forward jets). Finally, the NN was able to learn the both the mean and width of the resolution over a wide range of jet  $p_T$  (15 GeV to 2 TeV). This automatic learning of correlations between the input variables (i.e.  $p_T$  and  $\eta$ ) and the resolution is one of the attractive features of using an ML-based transformation, allowing for rapid deployment of detector parametrizations.

Additional improvements could be made by including more information about the objects (e.g. whether a  $b$ -quark is present in a jet, kinematic information from other objects in the event) potentially making this method more robust. This method could be extendable to additional reconstructed quantities and could be used to automatically model the ATLAS

and CMS detector. Another improvement left for future studies is the use of mixture density networks [10] to estimate resolution functions as a combination of Gaussians. The method described in this paper allows for automated detector parametrization which can facilitate phenomenological studies, efficient truth event selection, and upgrade studies.

## ACKNOWLEDGMENTS

We gratefully acknowledge the computing resources provided on a high-performance computing cluster operated by the Laboratory Computing Resource Center at Argonne National Laboratory. The submitted manuscript has been created by UChicago Argonne, LLC, Operator of Argonne National Laboratory (“Argonne”). Argonne, a U.S. Department of Energy Office of Science laboratory, is operated under Contract No. DE-AC02-06CH11357. The U.S. Government retains for itself, and others acting on its behalf, a paid-up nonexclusive, irrevocable worldwide license in said article to reproduce, prepare derivative works, distribute copies to the public, and perform publicly and display publicly, by or on behalf of the Government. The Department of Energy will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan. <http://energy.gov/downloads/doe-public-access-plan>. Argonne National Laboratory’s work was funded by the U.S. Department of Energy, Office of High Energy Physics under contract DE-AC02-06CH11357.

- 
- [1] J. de Favereau, C. Delaere, P. Demin, A. Giammanco, V. Lemaître, A. Mertens, and M. Selvaggi. DELPHES 3, A modular framework for fast simulation of a generic collider experiment. *JHEP*, 02:057, 2014.
  - [2] J. Alwall, R. Frederix, S. Frixione, V. Hirschi, F. Maltoni, et al. The automated computation of tree-level and next-to-leading order differential cross sections, and their matching to parton shower simulations. *JHEP*, 07:079, 2014.
  - [3] Matteo Cacciari, Gavin P. Salam, and Gregory Soyez. FastJet User Manual. *Eur. Phys. J. C*, 72:1896, 2012.

- 216 [4] Matteo Cacciari, Gavin P. Salam, and Gregory Soyez. The anti- $k_t$  jet clustering algorithm.  
217 *JHEP*, 04:063, 2008.
- 218 [5] S.V. Chekanov. HepSim: a repository with predictions for high-energy physics experiments.  
219 *Advances in High Energy Physics*, 2015:136093, 2015. Available as [http://atlaswww.hep.  
220 anl.gov/hepsim/](http://atlaswww.hep.anl.gov/hepsim/).
- 221 [6] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *arXiv  
222 e-prints*, page arXiv:1412.6980, Dec 2014.
- 223 [7] Martín Abadi et al. TensorFlow: Large-scale machine learning on heterogeneous systems,  
224 2015. Software available from tensorflow.org.
- 225 [8] Ying Li, Hui Li, Frank C. Pickard, Badri Narayanan, Fatih G. Sen, Maria K. Y. Chan, Subra-  
226 manian K. R. S. Sankaranarayanan, Bernard R. Brooks, and Benoît Roux. Machine learning  
227 force field parameters from ab initio data. *Journal of Chemical Theory and Computation*,  
228 13(9):4492–4503, 2017. PMID: 28800233.
- 229 [9] Md Mahbubul Islam, Grigory Kolesov, Toon Verstraelen, Efthimios Kaxiras, and Adri C. T.  
230 van Duin. ereaxff: A pseudoclassical treatment of explicit electrons within reactive force field  
231 simulations. *Journal of Chemical Theory and Computation*, 12(8):3463–3472, 2016. PMID:  
232 27399177.
- 233 [10] Christopher M. Bishop. Mixture density networks. Technical report, 1994.