

# Replication of detector simulations using supervised machine learning

D. Benjamin, S. Chekanov, W. Hopkins, Y. Li, J. R. Love,

*HEP Division, Argonne National Laboratory,  
9700 S. Cass Avenue, Argonne, IL 60439, USA*

(Dated: May 9, 2019)

## Abstract

Accurately and computationally rapidly modeling stochastic detector response for complex LHC experiments involving many particles from multiple interaction points, up to 200 interactions per proton-proton crossing in the HL-LHC requires the development of novel techniques. A study aimed at finding a fast transformation from truth-level physics objects to reconstructed detector-level physics objects is presented. This study used Delphes fast simulation based on an LHC-like detector geometry for inputs for a multi-categorizing machine learning (ML) algorithms. This ML transfer algorithms, with sufficient optimization, could have a wide range of applications: improving phenomenological studies by using a better detector representation, speeding up fast simulations based on parametric description of LHC detector responses, and allowing for more efficient production of Geant-4 simulation by only simulating events within an interesting part of phase space.

## 7 I. INTRODUCTION

8 A cornerstone of particle collision experiments is Monte Carlo (MC) simulations of physics  
9 processes followed by simulations of detector responses. With increased complexity of such  
10 experiments, such as those at the Large Hadron Collider (LHC), the detector simulations  
11 become increasing complex and time consuming. For example, the time required to simulate  
12 Geant4 [1] hits and to reconstruct from such hits physics objects (electrons, muons, taus,  
13 jets) requires a factor 100-1000 more CPU time than the creation of typical Monte Carlo  
14 events that represent physics processes according to theoretical models (“truth level” MC  
15 event generation). A possible method to speed up simulations of detector responses is  
16 to apply neural networks (NN) trained using the Geant4-based simulations, and use such  
17 supervised NN for transforming truth-level MC objects (jets and other identified particles)  
18 to objects modified by detectors (“detector-level”).

19 A typical simulation of detector responses stochastically modifies positions and energies  
20 of particles and jets created by MC generators at the truth-level. Another important compo-  
21 nent of such simulations is to introduce additional particles due to misreconstructed energy  
22 deposits in active detector volumes (examples include misreconstructed electrons or photons  
23 which are, in fact, hadronic jets). The latter effects represent a significant complication for  
24 the so-called “fast” or “parameterized” detector simulations, such as Delphes [2]. Never-  
25 theless, parameterized detector simulations have been proven to be a vital tools for physics  
26 performance and phenomenological studies.

27 The main advantage of detector parameterization based on machine learning is that a  
28 neural networks can automatically learn the features introduced by detailed full simula-  
29 tions, therefore, handcrafting parameters to represent resolutions and inefficiencies, as it  
30 was done in Delphes and for upgrade studies [? ], is not required. A neural network trained  
31 using realistic detector simulation should memorize the transformation from truth-level to  
32 the detector-level quantities without manual binning of quantities by analyzers. Another  
33 advantage is that the NN approach can introduce a complex interdependence of variables  
34 which is currently difficult to implements in parameterized simulations.

35 As a first step towards parameterized detector simulations using machine learning tech-  
36 niques, it is instructive to investigate how a transformation from the truth-level MC to  
37 detector-level objects can be performed, leaving aside the question of introducing objects

38 that are created by misreconstructions.

## 39 II. TRADITIONAL PARAMETERIZED FAST SIMULATIONS

40 In abstract terms, a typical variable  $f_i$  that characterizes a particle/jet, such as transverse  
 41 momentum ( $p_T$ ), pseudorapidity ( $\eta$ ), can be viewed as a multivariate transform  $F$  of the  
 42 original variable  $\xi_1^T$  at truth-level:

$$\xi_1 = F(\xi_1^T, \xi_2^T, \xi_3^T, \dots, \xi_N^T).$$

43 Generally, such a transform depends on several other variables  $\xi_2^T \dots \xi_N^T$  characterizing  
 44 this (or other) objects at the truth level. For example, the extent at which jet transverse  
 45 momentum,  $p_T$  is modified by a detector depends on the original truth-level transverse  
 46 momentum ( $\xi_1^T = p_T^T$ ), pseudorapidity  $\eta$ , flavor of jets and other effects that can be inferred  
 47 from the truth level. Similarly if particular detector modules in the azimuthal angle ( $\phi$ ) are  
 48 not active, this would introduce an additional dependence of this transform on  $\phi$ .

49 Typical parameterized simulations ignore the full range of correlations between the vari-  
 50 ables. In most cases, the above transform is reduced to a single variable, or two (as in the  
 51 case of Delphes simulations where energy resolution of clusters depend on the original ener-  
 52 gies of particles and their positions in  $\eta$ ). In order to take into account correlations between  
 53 multiple parameters characterizing transformations to the detector level, the following steps  
 54 have to be undertaken:

- 55 • create a grid in the hypercube with the dimension  $N_b^N$ , where  $N_b$  is the number of  
 56 histogram bins for the distributions  $f_1 - f_i^N$  representing “resolution” smearing. This  
 57 can be done numerically, using frequencies, or using analytically using “resolution  
 58 functions”.
- 59 • calculate “efficiencies” that model losses of particles/jets for each variable.

60 It should be pointed out that the calculation speed for parameterized simulations of one  
 61 variable that depends on  $N$  other variables at the truth level depends as  $N_b^N$  since each  
 62 object at the truth level should be placed inside the grid defined by  $N_b$  bins. Therefore,  
 63 complex parameterisations of resolutions and efficiency’s for  $N > 2$  becomes CPU intensive.

### 64 III. MACHINE LEARNING APPROACH FOR FAST SIMULATION

65 Unlike the traditional approach for fast simulation using parameterized density functions  
 66 for resolution variables and probability values for efficiency, a neural network approach  
 67 offers an opportunity to formulate this problem in terms of neural-network nodes and their  
 68 connections that scale as  $N_b^N \cdot N$ , which can speed up the fast simulations and, at the same  
 69 time, can be used for learning more complex full simulations in an automated way.

70 In the case of objects, such as jets, a typical truth-level input are jet transverse momen-  
 71 tum,  $\eta$ ,  $\phi$  and jet mass  $m$ , while the output is an array of output nodes that represent the  
 72 binned probability density function (pdf) of the resolution for a single variable (such as jet  
 73  $p_T$ ). Additional input variables can be jet flavor at the truth level, jet radius etc., i.e. any  
 74 variable that can influence the output of such neural network. Figure 1 shows a schematic  
 75 representation of the NN architecture for modelling detector response for a single output  
 76 variable. In this example, we show a single hidden layer (in principle, the NN can also be  
 77 deep with several hidden layers)

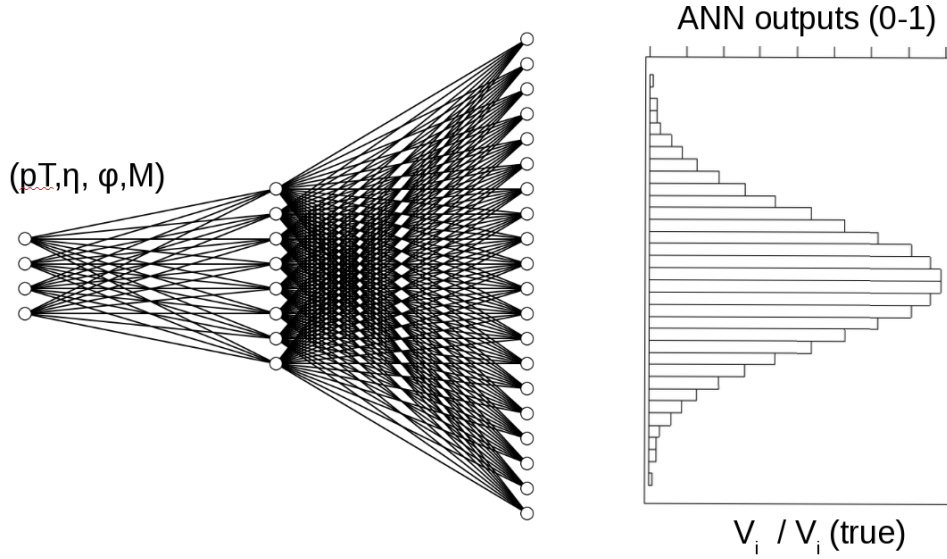


FIG. 1. A schematic representation of the NN architecture for modelling the detector response to truth-level input variables.

## 78 IV. MONTE CARLO SIMULATED EVENT SAMPLES

79 Monte Carlo events used for this analysis were generated using the Madgraph genera-  
80 tor [3]. The simulated processes were a combination of  $t\bar{t}$ +jets and  $\gamma$ +jets, which give a  
81 high rate of jets and lepton. Hadronic jets were reconstructed with the FASTJET package [4]  
82 using the anti- $k_T$  algorithm [5] with a distance parameter of 0.4. The detector simulation  
83 was performed with the Delphes package [2] with an ATLAS-like detector geometry. The  
84 event samples used in this paper, before and after the fast simulation, are available from  
85 the HepSim database [6]. In this paper only the transformation from truth-level jets to  
86 detector-level jets and only for  $p_T$  was performed, however the methodology should be ob-  
87 ject and parameter agnostic. Only truth jets which have been matched to a reconstructed  
88 Delphes jet are used. For the matching criteria the reconstructed jet that has the smallest  
89  $\Delta R = \sqrt{\Delta\phi^2 + \Delta\eta^2}$ , where  $\Delta\phi = \phi^{\text{truth}} - \phi^{\text{reco}}$  and  $\Delta\eta = \eta^{\text{truth}} - \eta^{\text{reco}}$ , with respect to the  
90 truth jet is chosen. If this minimum  $\Delta R$  is greater than 0.2, the truth-level jet is discarded.  
91 The poor modeling of the NN below  $p_T \sim 50$  GeV is due to the  $p_T > 15$  GeV requirement  
92 made on the reconstructed and truth jets. The requirement on the truth jets in combination  
93 with the width of the  $p_T^{\text{truth}} - p_T^{\text{reco}}$ , shown in Figure 3, results in insufficient information  
94 being available to transform truth-level jet  $p_T$ s to reconstruction-level jet  $p_T$ s. The final  
95 number of training jets used is two million while 500,000 jets were used as a testing sample.

96 The distributions of quantities used as the input for the NN,  $p_T$ ,  $\eta$ ,  $\phi$ ,  $m$ , are shown in  
97 Figure 2.

98 To facilitate gradient descent in all direction of the input variables, the input variables  
99 are all scaled to be in the range [0,1]. This avoids the  $p_T$  and the mass from having a  
100 disproportional affect on the training of the NNs. The output variable,  $p_T^{\text{truth}} - p_T^{\text{reco}}$ , is  
101 also scaled to have values between 0 and 1. Only objects that are within the 1<sup>st</sup> and 99<sup>th</sup>  
102 percentile of the  $p_T^{\text{truth}} - p_T^{\text{reco}}$  distribution are considered in this study since objects outside  
103 this range are typically not used in physics analyses.

## 104 V. NEURAL NETWORK STRUCTURES

105 An NN is trained with four input parameters, the scaled  $p_T$ ,  $\eta$ ,  $\phi$ , and  $m$ , and consist  
106 of five layers with 100 nodes each and with each node having a rectifier linear unit (ReLU)

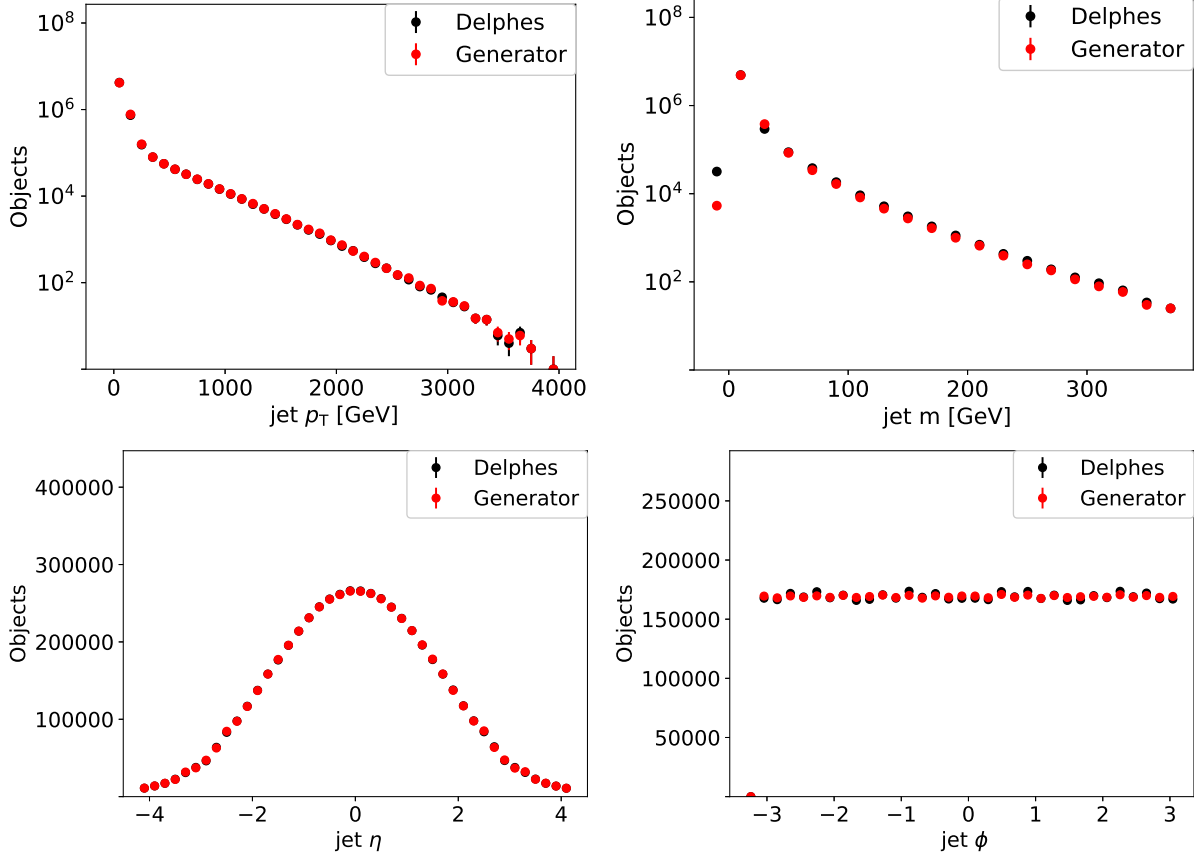


FIG. 2. Input variable shapes for truth-level (red) and detector-level quantities (black).

activation function. The output layer has 400 nodes with a softmax activation function. Finally, the NN is trained over 10 epochs with batch size 10 using the Adam [7] optimizer with a learning rate of . The NN is implemented using Keras [8] with a tensorflow [9] backend.

## VI. RESULTS

After the NN has been trained to learn the pdf of  $p_T^{\text{truth}} - p_T^{\text{reco}}$ , the resulting learned pdf is compared to the Delphes pdf using the testing sample in Fig. 4. Good agreement is observed between the Delphes and NN pdfs, showing that the NN has learned the bulk distribution.

The NN predicts a pdf for each jet based on its input parameters (i.e.  $p_T$ ,  $\phi$ ,  $\eta$ , and  $m$ ). The pdfs for a set of randomly selected jets are shown in Fig. 4. These pdfs are then randomly sampled to produce a NN jet that mimic the detector-level jet. A comparison

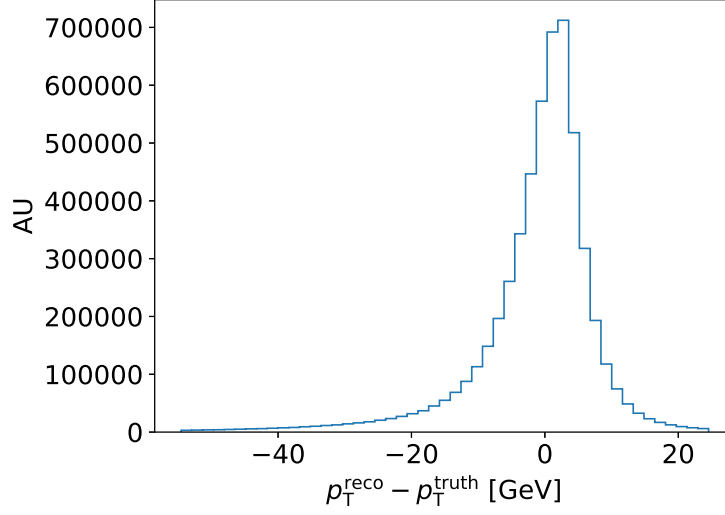


FIG. 3. Differences between truth-level and detector-level  $p_T$ .

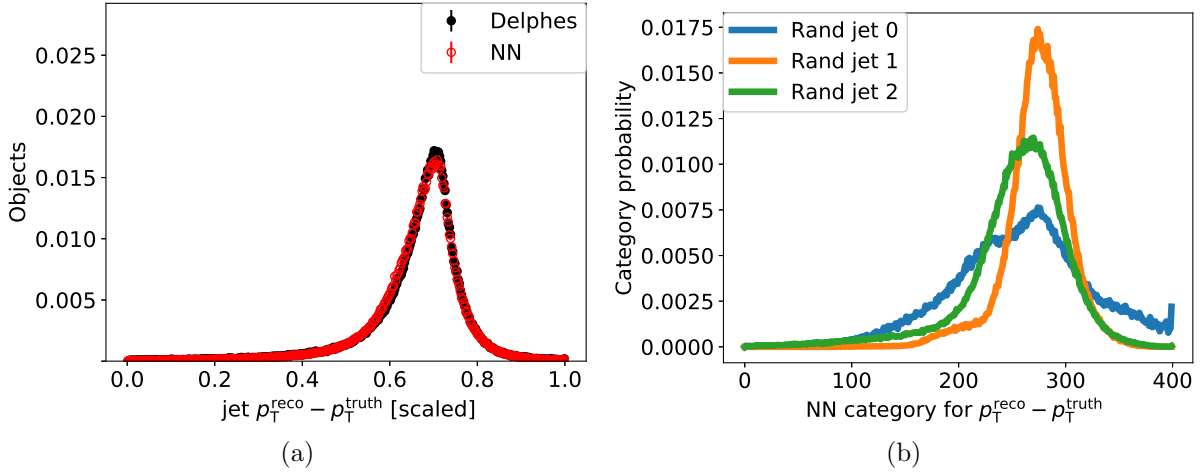


FIG. 4. NN-generated jet  $p_T^{\text{truth}} - p_T^{\text{reco}}$  compared to detector-level jet  $p_T^{\text{truth}} - p_T^{\text{reco}}$  (a). Right: NN-generated jet pdfs for three randomly selected truth-level jets (b).

119 of the NN-generated and Delphes jet  $p_T$  distributions for the testing sample and for NNs  
 120 trained with batch sizes 10000, 1000, 10, and 5 are shown in Fig. 5 while the distributions  
 121 at lower  $p_T$  are shown in Fig. 6. The NN reproduces the jet  $p_T$  distribution of Delphes  
 122 within 5% for reconstructed jets with  $p_T > 20$  GeV for all batch size less than 100.

123 To test whether the NN learned correlations between input parameters and the  $p_T$  res-  
 124 olution, defined as  $\frac{p_T^{\text{truth}} - p_T^{\text{reco}}}{p_T^{\text{truth}}}$ , the jets were divided into central ( $|\eta| < 3.2$ ) and forward  
 125 ( $|\eta| > 3.2$ ) jets, then the  $p_T$  resolution is compared between the two regions for both the  
 126 Delphes jets as well as the NN-generated jets. These two regions in the detector simulation

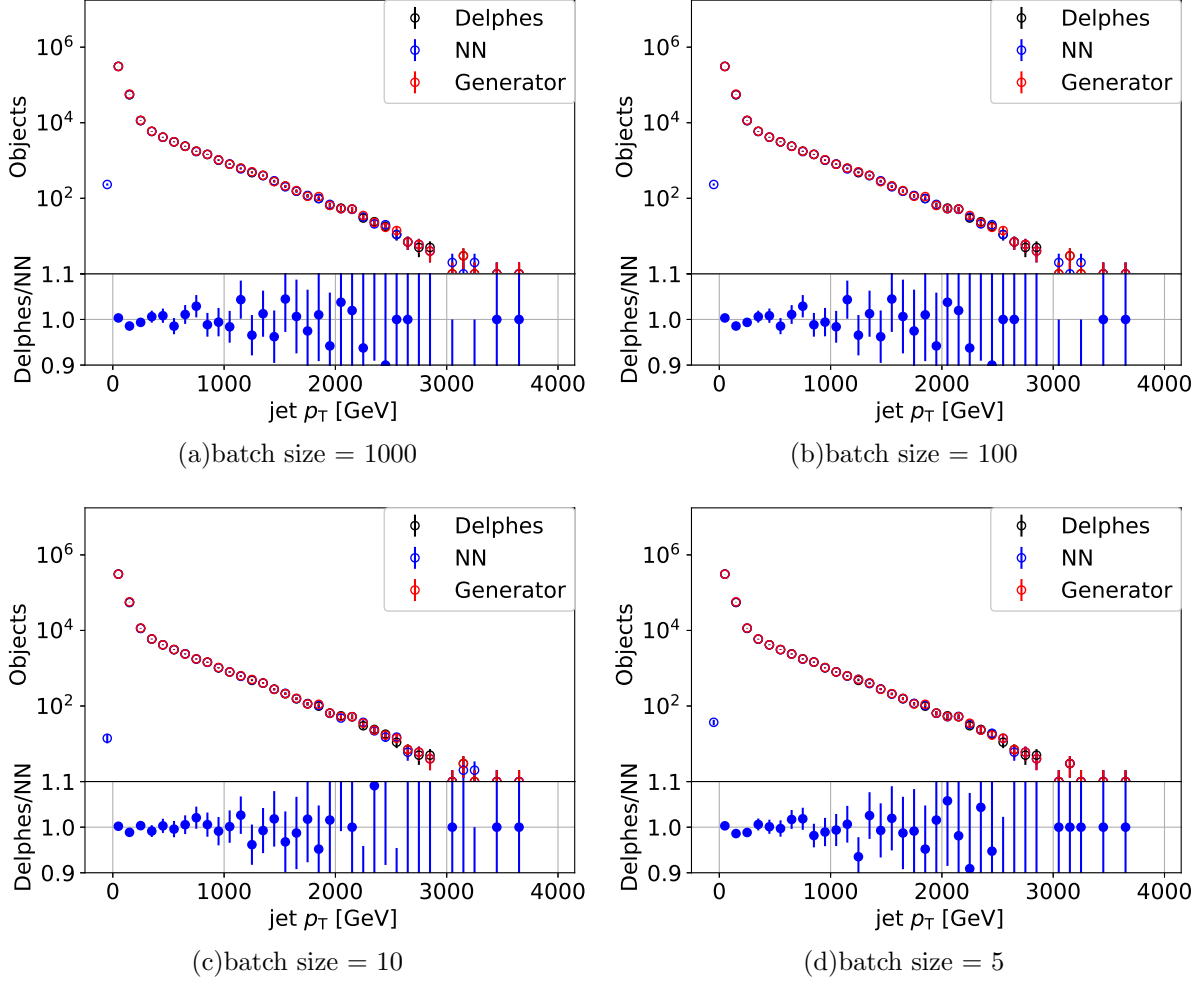


FIG. 5. NN-generated jet  $p_T$  distribution compared to truth-level and detector-level jet  $p_T$  distributions for batch sizes 10000 (a), 1000 (b), 10 (c), and 5 (d)..

have different calorimeter resolutions which results in different jet  $p_T$  resolutions and thus the  $p_T$  resolution is correlated with  $|\eta|$ . The resulting resolutions for both regions and for the training samples and jets are shown in Fig. 7. To quantify how the training batch size affects the NNs ability to reproduce the resolution of the forward region, which has significantly lower statistics and is a small subsample of the total training sample, the two-sample Kolmogorov–Smirnov statistic was calculated for NN and Delphes  $p_T$  resolution distribution for both the central and forward regions. The training sample was chosen for these tests because it had significantly more jets and due the low number of jets that have  $|\eta| > 3.2$ , as can be seen in Fig. 2. The NN clearly performs better for the rare subsample (the forward region) when trained on smaller batch sizes but the NN performs worse for batch size of five.



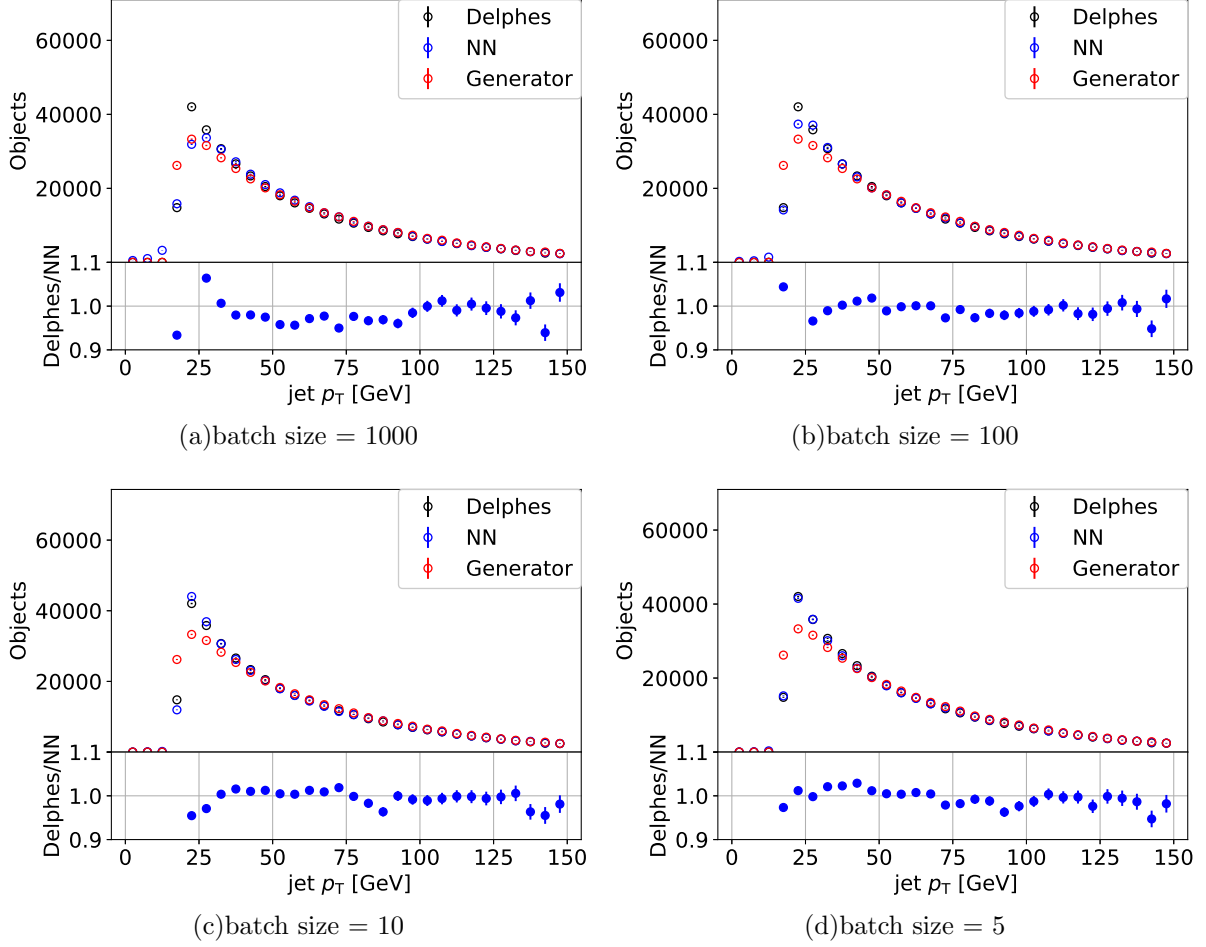


FIG. 6. NN-generated jet  $p_T$  compared to truth-level and detector-level jet  $p_T$  distributions for batch sizes 10000 (a), 1000 (b), 10 (c), and 5 (d).

## VII. HYPERPARAMETER SCAN

## VIII. CONCLUSION

We have shown that a truth-level quantity can be transformed to a reconstruction-level quantity using a multi-categorizing NN. The NN learned the changes in resolutions of different regions of the detector. For the NN to learn these changes in resolutions for small subsets of objects, such as for forward jets, small batch size training was necessary. The NN learned the truth-to-reconstruction transformation without requiring manual binning to capture the differences in resolutions of particular subsamples. This method should be easily extendable to additional reconstructed quantities and could be used to model the ATLAS and CMS detector. The method described in this paper thus allows for automated

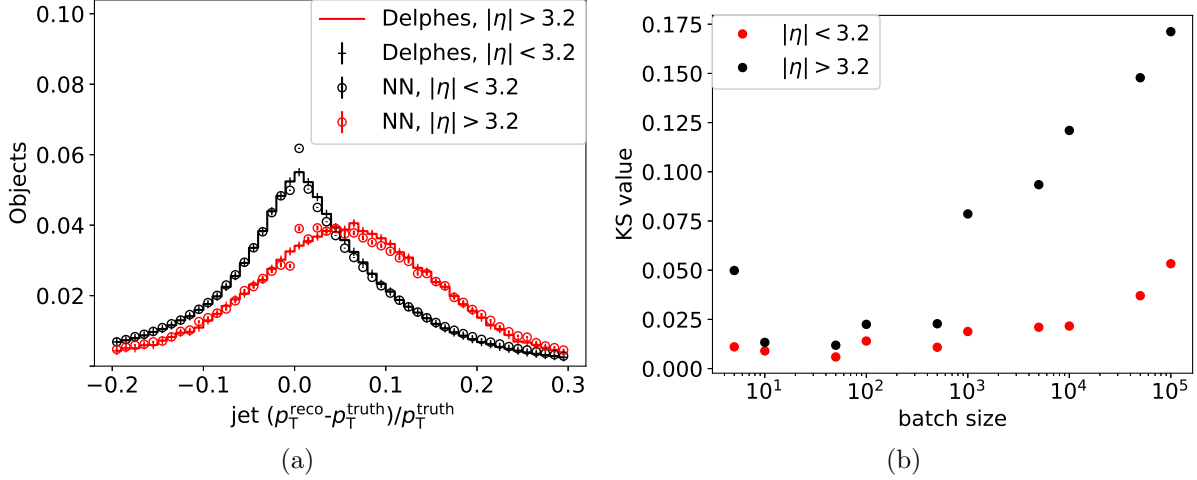


FIG. 7. Resolutions for the jet  $p_T$  (a) for the training sample for both the central and forward region using a batch size of 10 during the training of the model. The KS statistic for NN and Delphes resolutions as a function of batch size (b) is also shown. The red dots show the KS statistic for the central region while the black dots show the KS statistic for the forward region.

147 detector parameterization which can facilitate phenomenological, efficient Geant-4 simulation,  
 148 and upgrade studies.

## 149 ACKNOWLEDGMENTS

150 The submitted manuscript has been created by UChicago Argonne, LLC, Operator of  
 151 Argonne National Laboratory (“Argonne”). Argonne, a U.S. Department of Energy Office  
 152 of Science laboratory, is operated under Contract No. DE-AC02-06CH11357. The U.S. Gov-  
 153 ernment retains for itself, and others acting on its behalf, a paid-up nonexclusive, irrevocable  
 154 worldwide license in said article to reproduce, prepare derivative works, distribute copies to  
 155 the public, and perform publicly and display publicly, by or on behalf of the Government.  
 156 The Department of Energy will provide public access to these results of federally sponsored  
 157 research in accordance with the DOE Public Access Plan. [http://energy.gov/downloads/](http://energy.gov/downloads/doe-public-access-plan)  
 158 [doe-public-access-plan](http://energy.gov/downloads/doe-public-access-plan). Argonne National Laboratory’s work was funded by the U.S.  
 159 Department of Energy, Office of High Energy Physics under contract DE-AC02-06CH11357.

---

160 [1] S. Agostinelli et al. GEANT4: A Simulation toolkit. *Nucl. Instrum. Meth. A*, 506:250, 2003.

- 161 [2] 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. 162 *JHEP*, 02:057, 2014.
- 163 [3] J. Alwall, R. Frederix, S. Frixione, V. Hirschi, F. Maltoni, et al. The automated computation 164 of tree-level and next-to-leading order differential cross sections, and their matching to parton 165 shower simulations. *JHEP*, 07:079, 2014.
- 166 [4] Matteo Cacciari, Gavin P. Salam, and Gregory Soyez. FastJet User Manual. *Eur. Phys. J. C*, 167 72:1896, 2012.
- 168 [5] Matteo Cacciari, Gavin P. Salam, and Gregory Soyez. The anti- $k_t$  jet clustering algorithm. 169 *JHEP*, 04:063, 2008.
- 170 [6] S.V. Chekanov. HepSim: a repository with predictions for high-energy physics experiments. 171 *Advances in High Energy Physics*, 2015:136093, 2015. Available as [http://atlaswww.hep. 173 anl.gov/hepsim/](http://atlaswww.hep. 172 anl.gov/hepsim/).
- 174 [7] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *arXiv 175 e-prints*, page arXiv:1412.6980, Dec 2014.
- 176 [8] François Chollet et al. Keras. <https://keras.io>, 2015.
- 177 [9] Martín Abadi et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. 178 Software available from tensorflow.org.