Loss Functionals for Learning Likelihood Ratios

Shahzar Rizvi
Department of Statistics
University of California
Berkeley, CA 94720
shahzar@berkeley.edu

Mariel Pettee
Physics Division
Lawrence Berkeley National Laboratory
Berkeley, CA 94720
mpettee@lbl.gov

Benjamin Nachman
Physics Division
Lawrence Berkeley National Laboratory
Berkeley, CA 94720
bpnachman@lbl.gov
Berkeley Institute for Data Science, University of California, Berkeley, CA 94720, USA

Abstract

The likelihood ratio is a crucial quantity for statistical inference that enables hypothesis testing, construction of confidence intervals, reweighting of distributions, and more. For modern data- or simulation-driven scientific research, however, computing the likelihood ratio can be difficult or even impossible. Approximations of the likelihood ratio may be computed using parametrizations of neural network-based classifiers. By evaluating four losses in approximating the likelihood ratio of univariate Gaussians and simulated high-energy particle physics datasets, we recommend particular configurations for each loss and propose a strategy to scan over generalized loss families for the best overall performance.

1 Introduction

Claiming a scientific discovery requires a hypothesis test, e.g. a statistical threshold for claiming that one's experimental data reject the null hypothesis H_0 in favor of an alternative hypothesis H_1. By the Neyman-Pearson lemma [1], the strongest ("uniformly most powerful") measure of whether the experimental data x support H_0 vs. H_1 is a likelihood ratio test. These tests are particularly widespread in reporting results in High-Energy Physics (HEP), but are also commonly used in statistical analyses for many diverse scientific domains. The need for likelihood ratios goes beyond hypothesis testing, too—they can also be used to reweight a distribution to align with a target distribution, such as reweighting simulation samples to match real data [2–9].

In practice, H_0 and H_1 are defined by the sets of parameters θ_0 and θ_1, yielding probability densities $p(x \mid \theta_0)$ and $p(x \mid \theta_1)$. However, these densities may be unknown or very difficult to compute, as in the case of complex simulation models, rendering the computation of their ratio impossible. In such cases, it is well-known in statistics literature [10–12] that we can directly approximate the likelihood ratio $\mathcal{L}(x)$ by configuring a neural network classifier to use the "Likelihood Ratio Trick" (Theorem 1.1):

Theorem 1.1 ("Likelihood Ratio Trick"). *For a convex loss functional of the form*

$$L[f] = -\int \mathrm{d}x \Big(p(x \mid \theta_0) A(f(x)) + p(x \mid \theta_1) B(f(x)) \Big), \tag{1}$$

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

where $f : \mathbb{R}^n \to \mathbb{R}$ is a learnable function, the extremum is achieved when

$$-\frac{B'(f(x))}{A'(f(x))} = \frac{p(x \mid \theta_0)}{p(x \mid \theta_1)} = \mathcal{L}(x). \tag{2}$$

As long as $-B'(f)/A'(f)$ is a monotone rescaling of f, a neural network classifier configured to minimize $L[f]$ will approximate the likelihood ratio $\mathcal{L}(x)$.

In Eq. 1, $A(f)$ and $B(f)$ can take a number of forms to define a neural network classifer loss. In the limit of infinite training data, the choice of $A(f)$ and $B(f)$ should be irrelevant. In practice, however, this choice could significantly alter the effectiveness of Theorem 1.1.

To understand these effects in practice, we directly compare the four loss functionals defined in Table 1 with various choices of activation functions. The minimizations of each of the four loss functionals directly corresponds to minimizing some statistical divergence, up to an overall scaling and offset [13].

Loss Name	$A(f)$	$B(f)$	Related Divergence
Binary Cross-Entropy (BCE)	$\ln(f)$	$\ln(1-f)$	$2(\text{Jensen-Shannon} - \log 2)$
Mean Squared Error (MSE)	$-(1-f)^2$	$-f^2$	$\frac{1}{2}\,(\text{Triangular} - 1)$
Maximum Likelihood Classifier (MLC)	$\ln(f)$	$1-f$	Kullback-Liebler
Square Root (SQR)	$-\frac{1}{\sqrt{f}}$	$-\sqrt{f}$	$2\left(\text{Hellinger}^2 - 1\right)$

Table 1: The four primary loss functionals considered for comparing the effectiveness of the Likelihood Ratio Trick under different configurations.

On top of those four loss functionals, we also consider two parametric families of loss functionals, created by generalizing the MSE and SQR loss functionals. While there are several possible parametrizations from which to choose[1], we selected the following for simplicity: for the MSE loss, we considered a power parameter $p \in \mathbb{R}$, where $p = 2$ is the default value, and for the SQR loss, we considered a root parameter $r \in \mathbb{R}$, where $r = 1$ is the default value. We exclude the cases $p \in (0, 1)$ and $r = 0$ as the corresponding loss functionals are non-convex, and as such the likelihood ratio trick does not apply. Overall, this yields the following two families of losses:

Loss Name	$A(f)$	$B(f)$
p-MSE	$-(1-f)^p$	$-f^p$
r-SQR	$-f^{-\frac{r}{2}}$	$-f^{\frac{r}{2}}$

Table 2: Generalizations of MSE and SQR losses to entire families of losses parametrized by p and r. Values of $p = 2$ and $r = 1$ correspond to the original definitions of the losses.

All the loss functionals are compared on the task of estimating the likelihood ratio for one-dimensional Gaussian datasets (Section 2) as well as HEP simulation datasets (Section 3). We conclude in Section **??** with general recommendations for specifying the loss function when applying Theorem 1.1 for scientific analyses.

2 One-Dimensional Gaussians

In this section, we examine which of the loss functionals (with which kinds of activation functions) perform the best on a simple example of a one-dimensional Gaussians dataset. We consider two one-dimensional Gaussian distributions (see Appendix A) with slightly different means and unit variances: $X_0 \sim \text{Normal}(\mu = +0.1, \sigma = 1)$ and $X_1 \sim \text{Normal}(\mu = -0.1, \sigma = 1)$ and estimate the likelihood ratio between them using Theorem 1.1. In particular, we train neural network classifiers to minimize the loss functionals and compare their performances in estimating the likelihood ratio.

[1] For example, to enforce non-singular behavior at $r = 0$ for SQR, one could consider $A(f) = (1 - f^{-\frac{r}{2}})/|r|$ and $B(f) = (1 - f^{\frac{r}{2}})/|r|$. Another interesting parametrization is $A(f) = (f^q - 1)/q$ and $B(f) = 1 - f^{(q+1)}/(q+1)$, which is minimized at $q = 1$.

The neural networks parametrize the learned function f as $\phi(z)$, where z is the pre-activation output of the network and ϕ is the final activation function. For the BCE and MSE losses, the extremum is achieved when $\mathcal{L} = f/(1-f)$, so f is restricted to the range $f \in (0,1)$. Three appropriate activation functions $\phi : \mathbb{R} \to (0,1)$ were considered for these two losses: the somewhat standard logistic loss $\sigma(z) = (1+e^{-z})^{-1}$, the cumulative distribution function of a Gaussian $\Phi(z)$, and $\frac{1}{\pi}(\tan^{-1}(z) + \frac{\pi}{2})$. The MLC and SQR losses instead yield $\mathcal{L} = f$, so three different activation functions $\phi : \mathbb{R} \to (0,\infty)$ were chosen to suit the range: the typical choice ReLU(z), as well as z^2 and e^z.

2.1 Methods

For each pairing of loss functional and activation function, we trained 100 independent, identical classifier models on two Gaussian datasets of 10^6 samples each. Each classifier was evaluated on the interval $(-6,6)$ and transformed into an estimator for the likelihood ratio on the same interval.

To numerically compare the performances of different classifiers in learning the likelihood ratio, we computed their empirical mean absolute errors over 10^5 samples. For the estimated likelihood ratio $\hat{\mathcal{L}}$, the mean absolute error is defined as $\text{MAE}[\hat{\mathcal{L}}] = \mathbb{E}[|\mathcal{L}(X) - \hat{\mathcal{L}}(X)|]$. We computed the MAE for each classifier architecture as an empirical average over the 100 different classifiers to quantify how well each estimator approximated the likelihood ratio. This allowed us to numerically compare how each activation function affected the performance of the loss functional.

We then compared how the choice of the parameters p and r changed the effectiveness of the p-MSE and r-SQR losses in learning the likelihood ratio. We scanned over values of p and r in the interval $[-2,2]$. For each value of p and r considered, we trained 20 models on the corresponding p-MSE or r-SQR loss functional and averaged the mean absolute errors of the 20 models together to find the p^* and r^* that minimized the MAE. Each model used the best-performing activation function for their respective losses. As shown in the next section, these were $\sigma(z)$ for MSE and e^z for SQR.

2.2 Results

Table 3 displays the empirical MAEs for the various loss functional and activation function pairs.

Loss Name	$\sigma(z)$	Error	$\Phi(z)$	Error	$\frac{1}{\pi}(\tan^{-1}(z) + \frac{\pi}{2})$	Error
BCE	**0.0081**	0.0003	0.0113	0.0004	0.0089	0.0003
MSE	**0.0081**	0.0003	0.0110	0.0004	0.0100	0.0003
p^*-MSE	**0.0046**	0.0001	—	—	—	—
Loss Name	**ReLU(z)**	**Error**	**z^2**	**Error**	**e^z**	**Error**
MLC	0.0148	0.0004	0.0683	0.0096	**0.0083**	0.0003
SQR	0.0367	0.0099	0.6756	0.0044	**0.0075**	0.0003
r^*-SQR	—	—	—	—	**0.0034**	0.0001

Table 3: Mean absolute errors are computed for various loss functional configurations in the classification of two univariate Gaussians of 10^6 samples. 100 independent and identical classifiers were trained for each configuration to calculate the uncertainties. Errors represent one standard deviation. The activation functions in the first column represent the typical choices for each loss functional.

Without modifying the typical setup for these loss functionals—i.e. using $\sigma(z)$ for BCE and MSE and ReLU(z) for MLC and SQR—one might conclude from the results in Table 3 that BCE and MSE are the better overall estimators of the likelihood ratio. SQR loss, in particular, has an error nearly five times as high as BCE or MSE in this context.

BCE and MSE losses do perform best with their typical choice of activation, $\sigma(z)$. However, selecting e^z instead of ReLU(z) greatly improves the performance of MLC and SQR, reducing the MAE associated with each loss functional by 44% and 80%, respectively.

An additional dramatic error reduction is introduced by scanning over p and r in the generalized forms of MSE and SQR losses. The values p^* and r^* minimizing the MAE were $p^* = 1.24$[2] and $r^* = 0.018$. The scan over r and some discussion regarding the optimization is presented in Section 4, and the scan over p is shown in Appendix B. The use of the p^*-MSE and r^*-SQR loss functionals reduce the MAE from the standard MSE and SQR loss functionals by 43% and 55%.

Overall, the four standard loss functionals (without optimizing over loss families) all perform about the same after accounting for appropriate choices of activation functions, but the SQR loss functional performs the best. Optimizing over p and r results in a reduction in MAE by about half, with the r^*-SQR performing the best, followed by the p^*-MSE.

3 High-Energy Particle Physics Simulation

To test the conclusions from the univariate Gaussian case on a more complex scientific dataset, we compared the same loss functional configurations on the task of estimating the likelihood ratio for simulated high-energy particle physics datasets [14] with four observables: leading jet transverse momentum (p_T), rapidity (y); azimuthal angle (ϕ), and invariant mass (m).

Datasets The datasets consist of particle-level and detector-level simulated QCD jets originating from Z + jets events. Z + jets events from proton-proton collisions generated at $\sqrt{s} = 14$ TeV were simulated using HERWIG 7.1.5 [15–17] with the default tune and PYTHIA 8.243 [18–20] tune 21 [21] (ATLAS A14 central tune with NNPDF2.3LO). For the generated events, the p_T of the Z boson is required to be larger than 150 GeV. Events then are passed through the DELPHES 3.4.2 fast detector simulation [22] of the CMS detector. The datasets consist of the highest-momentum jet from Z boson events with $p_T \geq 200$ GeV. This process ultimately yields about 1.6 million jets for each simulation.

3.1 Methods

As the true likelihood ratios are unknown for these samples, we fit Normalizing Flows [23] to each sample to estimate their generating distributions. These flows were validated by training an additional neural network classifier to distinguish between data sampled from the flows and the original datasets, then verifying that the AUC was approximately 0.5. We then used the flows as proxies for the underlying distributions of these datasets, creating new datasets by sampling from the flows. We repeated the same studies from Sec. 2, and continued to use the MAE as our figure-of-merit.

3.2 Results

Table 4 displays the empirical MAEs over the different classifiers for the various loss functional and activation function pairs.

Loss Name	$\sigma(z)$	Error	$\Phi(z)$	Error	$\frac{1}{\pi}(\tan^{-1}(z) + \frac{\pi}{2})$	Error
BCE	**0.1461**	0.0003	0.1485	0.0003	0.1469	0.0003
MSE	**0.1462**	0.0003	0.1484	0.0003	0.1476	0.0003
p^*-MSE	**0.1460**	0.0003	—	—	—	—
Loss Name	**ReLU(z)**	**Error**	**z^2**	**Error**	**e^z**	**Error**
MLC	0.2011	0.0008	0.2368	0.0127	**0.1467**	0.0003
SQR	0.2683	0.0051	0.8011	0.0186	**0.1463**	0.0003
r^*-SQR	—	—	—	—	**0.1447**	0.0002

Table 4: Mean absolute errors are computed for various loss functional configurations in the classification of two simulated high-energy physics datasets of $1.6 \cdot 10^6$ samples each. 100 independent and identical classifiers were trained for each configuration to calculate the uncertainties. Errors represent one standard deviation. The activation functions in the first column represent the typical choices for each loss functional.

[2] A similar MAE was achieved with $p^* = 1.08$, but 1.24 was selected for its numerical stability.

As shown in Table 4, the same activation functions that perform the best for each loss from Section 2 are also optimal in the physics case. BCE and MSE losses perform best with the default choice of activation function, $\sigma(z)$. For MLC and SQR losses, using e^z instead of the default choice of ReLU(z) improves the MAE by 27% and 45%, respectively.

Scanning over p and r found that the values $p^* = 1.92$ and $r^* = -0.25$ minimize the MAE. See Section 4 and Appendix B for the scans. These optimizations over p and r result in comparably more marginal reductions of the MAE from the standard MSE and SQR loss functionals by 0.1% and 1%.

4 Landscapes of Generalized Losses

The scans of r for the Gaussian and physics datasets are shown in Figure 1. (Scans of p are shown in App. B). Generally, values of r close to 0 tended to result in the smallest mean absolute errors. The vertical features at $r = 0$ are due to the non-convexity of the r-SQR loss functional at $r = 0$; as such, the Likelihood Ratio Trick does not apply.

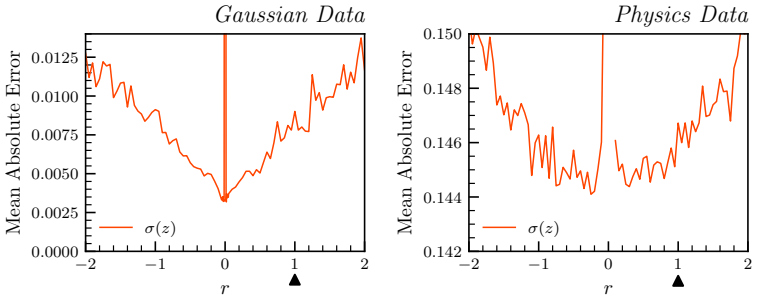

Figure 1: MAEs for classifiers trained on r-SQR losses for $r \in [-2, 2]$, computed from 20 independent and identical classifiers for each r. Arrows indicate the default choice of $r = 1$.

A deeper investigation of the trends displaying in these plots is omitted, but we describe the results of the investigation here. In simpler, two-parameter classifiers $\sigma(ax + b)$, we can directly examine the shape of the loss as a function of the parameters a and b. For such classifiers, the magnitudes of r and p correspond to the steepness of the loss landscape around the minimum (a^*, b^*).

In particular, values of r close to 0 yield shallow landscapes for which there is a large space of classifiers with close-to-optimal performance. That is, small perturbations to (a^*, b^*) do not change the performance of the classifiers very much from optimality. In contrast, large values of r corresponded to steeper landscapes for which only the optimal classifier performs well; small perturbations to (a^*, b^*) result in much worse performance.

5 Conclusion

Choosing optimal activation functions for each loss functional is crucial to achieving the best possible likelihood ratio estimate. Our experiments suggest that of the activations we considered, $\sigma(z)$ (for BCE and MSE) and e^z (for MLC and SQR) are the best.

For simple cases such as one-dimensional Gaussians, all four losses considered performed similarly with appropriately chosen activation functions. However, the mean absolute error in estimating the likelihood ratio was about 2 times smaller when optimized values of p and r were selected.

In the more complex case of simulated HEP data, all four losses still performed similarly after selecting the appropriate activation functions. However, optimizing over p and r only yielded marginal improvements (at most 1%) in reducing the MAE. The inconsistency in improvement from the optimization of p and r requires further investigation.

Overall, we recommend pairing the BCE and MSE losses with the logistic activation and the MLC and SQR losses with the exponential activation. The loss landscape will vary with each new application, so we recommend that future researchers perform a scan along p or r to find an optimum value as part of hyperparameter optimization.

A Datasets and Training Details

All of our classifiers were implemented as neural networks using KERAS [24] with a TENSORFLOW [25] backend and ADAM [26] optimizer. Each classifier consisted of three hidden layers with 64, 128, and 64 nodes, sequentially. Rectified Linear Unit (ReLU) activation functions were used for the intermediate layers, with the activation for the output layer depending on the loss used to train the neural network and the parametrization being tested. Each of the three hidden layers is followed by a dropout layer with dropout probability of 10%.

Unless otherwise stated, the networks were trained with 1,000,000 samples (750,000 used for training and 250,000 used for validation). 100,000 separate samples were used to evaluate the networks' performances (in particular, to calculate their mean absolute errors). Each network was trained for up to 100 epochs with a batch size of 10%, as in [13]. If the validation loss did not decrease for 10 consecutive epochs, the training was stopped (early stopping with a patience of 10). No detailed hyperparameter optimization was done.

B Generalized MSE Losses

The scans discussed in 2.1 and 3.1 indicated that values of p slightly above 1 or slightly below 0 yield better losses with which to train classifiers for the likelihood ratio trick.

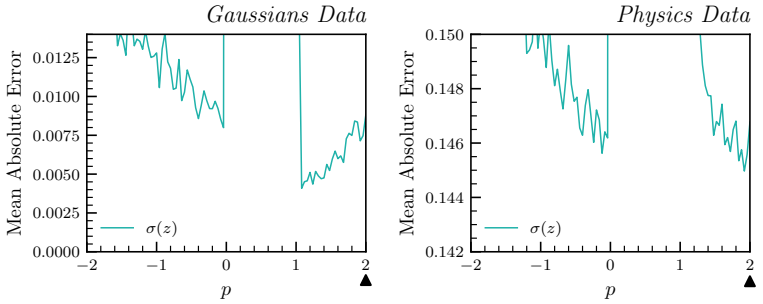

Figure 2: Mean absolute errors for classifiers trained on p-MSE losses. For each $p \in [-2, 2]$, 20 independent and identical classifiers were trained using the corresponding p-MSE loss to estimate the MAE. Arrows indicates the default choice of $p = 2$.

Figure 2 displays the results of these scans. The vertical features for $p \in [0, 1)$ are due to the non-convexity of the p-MSE loss functional for those values of p; as a result, the likelihood ratio trick does not work for such values.

In the Gaussians data, small values of p reduce the MAE by about half in comparison to the standard choice of $p = 2$. However, in the high-energy particle physics dataset, values of p other than $p = 2$ do not perform much better; our scan found that values of p close to 2 minimized the MAE out of all values of p we searched. Using the optimized value of $p^* = 1.92$ over $p = 2$ resulted in a marginal improvement: the MAE was reduced by less than 1% (see 4). However, it is possible that scanning over a larger range of values for p, e.g. $p \in [2, 3]$, could result in a better value of p^*.

Acknowledgments and Disclosure of Funding

We are grateful to Jesse Thaler for very helpful feedback about figures-of-merit and ways to generalize our loss functions. We thank Vinicius Mikuni for the idea of using normalizing flows to model the likelihood ratios of the physics datasets. M.P. thanks Shirley Ho and the Flatiron Institute for their hospitality while preparing this work. S.R., M.P., and B.N. are supported by the Department of Energy, Office of Science under contract number DE-AC02-05CH11231.

References

[1] J. Neyman and E. S. Pearson. On the Problem of the Most Efficient Tests of Statistical Hypotheses. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 231:289–337, 1933. ISSN 02643952. URL http://www.jstor.org/stable/91247.

[2] Anders Andreassen, Patrick T. Komiske, Eric M. Metodiev, Benjamin Nachman, and Jesse Thaler. OmniFold: A Method to Simultaneously Unfold All Observables. *Physical Review Letters*, 124(18), May 2020. doi: 10.1103/physrevlett.124.182001. URL https://doi.org/10.1103%2Fphysrevlett.124.182001.

[3] Alex Rogozhnikov. Reweighting with Boosted Decision Trees. *Journal of Physics: Conference Series*, 762:012036, oct 2016. doi: 10.1088/1742-6596/762/1/012036. URL https://doi.org/10.1088%2F1742-6596%2F762%2F1%2F012036.

[4] D Martschei, M Feindt, S Honc, and J Wagner-Kuhr. Advanced Event Reweighting using Multivariate Analysis. *Journal of Physics: Conference Series*, 368(1):012028, jun 2012. doi: 10.1088/1742-6596/368/1/012028. URL https://dx.doi.org/10.1088/1742-6596/368/1/012028.

[5] Anders Andreassen, Ilya Feige, Christopher Frye, and Matthew D. Schwartz. JUNIPR: a Framework for Unsupervised Machine Learning in Particle Physics. *The European Physical Journal C*, 79(2), Feb 2019. doi: 10.1140/epjc/s10052-019-6607-9. URL https://doi.org/10.1140%2Fepjc%2Fs10052-019-6607-9.

[6] Anders Andreassen and Benjamin Nachman. Neural Networks for Full Phase-Space Reweighting and Parameter Tuning. *Physical Review D*, 101(9), May 2020. doi: 10.1103/physrevd.101.091901. URL https://doi.org/10.1103%2Fphysrevd.101.091901.

[7] Roel Aaij et al. (LHCb Collaboration). Observation of the Decays $\Lambda_b^0 \to \chi_{c1} p K^-$ and $\Lambda_b^0 \to \chi_{c2} p K^-$. *Physical Review Letters*, 119(6), aug 2017. doi: 10.1103/physrevlett.119.062001. URL https://doi.org/10.1103%2Fphysrevlett.119.062001.

[8] M. Aaboud et al. (ATLAS Collaboration). Search for Pair Production of Higgsinos in Final States with at Least Three b-Tagged Jets in $\sqrt{s} = 13$ TeV pp Collisions using the ATLAS Detector. *Phys. Rev. D*, 98:092002, Nov 2018. doi: 10.1103/PhysRevD.98.092002. URL https://link.aps.org/doi/10.1103/PhysRevD.98.092002.

[9] Leander Fischer, Richard Naab, and Alexandra Trettin. Treating detector systematics via a likelihood free inference method, 2023.

[10] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.

[11] Masashi Sugiyama, Taiji Suzuki, and Takafumi Kanamori. *Density Ratio Estimation in Machine Learning*. Cambridge University Press, 2012. doi: 10.1017/CBO9781139035613.

[12] Benjamin K Miller, Christoph Weniger, and Patrick Forré. Contrastive Neural Ratio Estimation. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 3262–3278. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/159f7fe5b51ecd663b85337e8e28ce65-Paper-Conference.pdf.

[13] Benjamin Nachman and Jesse Thaler. E Pluribus Unum Ex Machina: Learning from Many Collider Events at Once. *Physical Review D*, 103(11), June 2021. doi: 10.1103/physrevd.103.116013. URL https://doi.org/10.1103%2Fphysrevd.103.116013.

[14] Anders Andreassen, Patrick Komiske, Eric Metodiev, Benjamin Nachman, and Jesse Thaler. Pythia/Herwig + Delphes Jet Datasets for OmniFold Unfolding, November 2019. URL https://doi.org/10.5281/zenodo.3548091.

[15] Manuel Bähr, Stefan Gieseke, Martyn A. Gigg, David Grellscheid, Keith Hamilton, Oluseyi Latunde-Dada, Simon Plätzer, Peter Richardson, Michael H. Seymour, Alexander Sherstnev, and Bryan R. Webber. Herwig++ Physics and Manual. *The European Physical Journal C*, 58 (4):639–707, nov 2008. doi: 10.1140/epjc/s10052-008-0798-9. URL https://doi.org/10.1140%2Fepjc%2Fs10052-008-0798-9.

[16] Johannes Bellm, Stefan Gieseke, David Grellscheid, Simon Plätzer, Michael Rauch, Christian Reuschle, Peter Richardson, Peter Schichtel, Michael H. Seymour, Andrzej Siódmok, Alexandra Wilcock, Nadine Fischer, Marco A. Harrendorf, Graeme Nail, Andreas Papaefstathiou, and Daniel Rauch. Herwig 7.0/Herwig++ 3.0 Release Note. *The European Physical Journal C*, 76 (4), Apr 2016. doi: 10.1140/epjc/s10052-016-4018-8. URL https://doi.org/10.1140%2Fepjc%2Fs10052-016-4018-8.

[17] Johannes Bellm, Stefan Gieseke, David Grellscheid, Patrick Kirchgaeßer, Frashër Loshaj, Graeme Nail, Andreas Papaefstathiou, Simon Plätzer, Radek Podskubka, Michael Rauch, Christian Reuschle, Peter Richardson, Peter Schichtel, Michael H. Seymour, Andrzej Siódmok, and Stephen Webster. Herwig 7.1 Release Note. Technical report, 2017. URL http://cds.cern.ch/record/2265397. 7 pages, 7 figures. Herwig is available from https://herwig.hepforge.org/.

[18] Torbjörn Sjöstrand, Stephen Mrenna, and Peter Skands. A Brief Introduction to PYTHIA 8.1. *Computer Physics Communications*, 178(11):852–867, Jun 2008. doi: 10.1016/j.cpc.2008.01.036. URL https://doi.org/10.1016%2Fj.cpc.2008.01.036.

[19] Torbjörn Sjöstrand, Stephen Mrenna, and Peter Skands. PYTHIA 6.4 Physics and Manual. *Journal of High Energy Physics*, 2006(05):026–026, May 2006. doi: 10.1088/1126-6708/2006/05/026. URL https://doi.org/10.1088%2F1126-6708%2F2006%2F05%2F026.

[20] Torbjörn Sjöstrand, Stefan Ask, Jesper R. Christiansen, Richard Corke, Nishita Desai, Philip Ilten, Stephen Mrenna, Stefan Prestel, Christine O. Rasmussen, and Peter Z. Skands. An Introduction to PYTHIA 8.2. *Computer Physics Communications*, 191:159–177, Jun 2015. doi: 10.1016/j.cpc.2015.01.024. URL https://doi.org/10.1016%2Fj.cpc.2015.01.024.

[21] ATLAS Pythia 8 Tunes to 7 TeV Data. Technical report, CERN, Geneva, 2014. URL https://cds.cern.ch/record/1966419. All figures including auxiliary figures are available at https://atlas.web.cern.ch/Atlas/GROUPS/PHYSICS /PUBNOTES/ATL-PHYS-PUB-2014-021.

[22] J. de Favereau, C. Delaere, P. Demin, A. Giammanco, V. Lemaître, A. Mertens, M. Selvaggi, and The DELPHES 3. collaboration. Delphes 3: a modular framework for fast simulation of a generic collider experiment. *Journal of High Energy Physics*, 2014(2):57, Feb 2014. ISSN 1029-8479. doi: 10.1007/JHEP02(2014)057. URL https://doi.org/10.1007/JHEP02(2014)057.

[23] Will Grathwohl, Ricky T. Q. Chen, Jesse Bettencourt, and David Duvenaud. Scalable Reversible Generative Models with Free-form Continuous Dynamics. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rJxgknCcK7.

[24] Francois Chollet et al. Keras, 2015. URL https://github.com/fchollet/keras.

[25] TensorFlow: Large-scale machine learning on heterogeneous systems.

[26] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6980.