Learning new physics from a machine

Raffaele Tito D'Agnolo[1] and Andrea Wulzer[2,3,4]

[1]*SLAC National Accelerator Laboratory, 2575 Sand Hill Road, Menlo Park, California 94025, USA*
[2]*Theoretical Physics Department, CERN, 1211 Geneva , Switzerland*
[3]*Theoretical Particle Physics Laboratory (LPTP), Institute of Physics, EPFL, 1511 Lausanne, Switzerland*
[4]*Dipartimento di Fisica e Astronomia, Università di Padova and INFN, Sezione di Padova, via Marzolo 8, I-35131 Padova, Italy*

(Received 19 October 2018; published 8 January 2019)

We propose using neural networks to detect data departures from a given reference model, with no prior bias on the nature of the new physics responsible for the discrepancy. The virtues of neural networks as unbiased function approximants make them particularly suited for this task. An algorithm that implements this idea is constructed, as a straightforward application of the likelihood-ratio hypothesis test. The algorithm compares observations with an auxiliary set of reference-distributed events, possibly obtained with a Monte Carlo event generator. It returns a p value, which measures the compatibility of the reference model with the data. It also identifies the most discrepant phase-space region of the data set, to be selected for further investigation. The most interesting potential applications are model-independent new physics searches, although our approach could also be used to compare the theoretical predictions of different Monte Carlo event generators, or for data validation algorithms. In this work we study the performance of our algorithm on a few simple examples. The results confirm the model independence of the approach, namely that it displays good sensitivity to a variety of putative signals. Furthermore, we show that the reach does not depend much on whether a favorable signal region is selected based on prior expectations. We identify directions for improvement towards applications to real experimental data sets.

DOI: 10.1103/PhysRevD.99.015014

I. INTRODUCTION

Today in fundamental physics we have at our disposal powerful theoretical models. They are in principle able to describe the outcome of all present and near-future experiments. In high-energy physics and cosmology these models are the Standard Model (SM) and ΛCDM, respectively. In the following we call them reference models. It is technically possible for the reference models to describe all present and future data, but that does not mean that they will. Future experiments will be able to explore phenomena that we have never observed before or to measure known phenomena with unprecedented accuracy. Furthermore, we are convinced that new physics (i.e., physical laws that are not yet established) exists because of the open problems of the reference models. Searching for new physics, which concretely means searching for discrepancies between the data and the reference model, is the absolute priority of our field.

In general the problem can be phrased in terms of many repeated measurements $\mathcal{D} = \{x_i\}$ (called events in high-energy physics) of a multidimensional random variable x. The statistical distribution for x can be predicted on the basis of the physical laws that constitute the reference model. The goal is to test the reference model distribution against the actual data. Several strategies exist to carry out this test. However the vast majority of them are not suited to discover discrepancies because of the nature of the problem at hand. The main challenge stems from the fact that the true underlying data distribution, possibly including new physics effects, will be "similar" to the reference one. We expect this because of existing constraints on new physics. Notice that similar does not mean that the effect of new physics cannot be large. However if it is large, it will be localized in a low-probability region of the space of observations where only a small fraction of the events is present. Alternatively the effect can be spread in a large region of the x space, but in this case it will be a small modification of the reference distribution. Essentially the problem is that our prior knowledge suggests that the vast majority of the collected events will agree with the reference model. At the same time this prior knowledge is insufficient to know where to look for discrepancies.

The most widely employed approach to the problem is to search for specific new physics models. In any such model, one can identify *a priori* the subset of data where large departures from the reference model should be concentrated or know how to exploit small, correlated deviations across the data set. Once a specific new physics model or a set of models are specified, one constructs hypothesis tests using standard techniques (see [1] for a concise review). The clear advantage of this approach is that it is physically informative even if the compatibility of the data with the reference model is confirmed. The disadvantage is that a statistical test which is designed to be sensitive to one specific hypothesis is typically insensitive to data departures of a different nature. This substantiates the widespread concern that we might not be able to discover new physics, even if present in the data, because it does not belong to the class of hypothetical models that we are searching for.

Motivated by the above observation, a number of attempts have been made [2–14] to construct model-independent new physics search strategies. However it is important to remark that a model-independent hypothesis test is an ill-defined concept in statistics. Testing one hypothesis unavoidably requires an alternative (in general composite) hypothesis to compare with. Technically what is needed is a set of alternative hypothetical distributions, depending on free parameters, which are also called an alternative probability "model" in statistics. In physics instead a model is a set of physical laws that allows one to predict these distributions. Therefore we call a search strategy model independent (in the physics sense) when the alternative distributions do not follow from a physical model but are selected with other criteria. The most important criterion is flexibility, namely the ability of the distributions to adapt themselves, for an appropriate choice of the free parameters, to the true underlying data distribution. This will ensure sensitivity to a large variety of new physics scenarios, including those that are not predicted by any of the models that have been constructed until now. The idea behind the present paper is to use artificial neural networks to parametrize the alternative distributions.

Neural networks are increasingly important tools in high-energy physics. Applications include jet physics [15–39], new physics searches [40–47], detector simulation [48–50] and the NNPDF fit to parton distribution functions [51], where they have been successfully applied for a long time [52]. The main reason for their success is precisely their virtues as efficient and unbiased approximants [53–60]. They are often introduced as a convenient alternative to piecewise constant functions (histograms) for the fit to distributions [61–63]. Employing them to parametrize alternative distributions for model-independent new physics searches is thus a highly motivated attempt. To the best of our knowledge this possibility has not been previously discussed. Most applications of neural networks to new physics searches aim at enhancing the sensitivity to prespecified models of the resonant or nonresonant type. Using machine learning techniques for model-independent new physics searches has been proposed in [64]; however Gaussian mixture models are employed rather than neural networks and the overall strategy is quite different from ours. Reference [47] uses neural networks, but with the purpose of enhancing the sensitivity to resonant bumps that emerge in a prespecified kinematical variable. What we do is conceptually very similar to anomaly detection, where neural networks are already employed extensively. However the purpose of anomaly detection is to identify rare events in the data sample. Our purpose is instead to find an anomalous behavior, relative to the reference model, of the entire data sample.

The paper is organized as follows. In Sec. II we introduce the conceptual foundations of our approach, explaining in detail the advantages of using neural networks for model-independent new physics searches. We will see that our strategy is a straightforward application of maximum likelihood estimation and likelihood-ratio hypothesis testing, which are easily turned into a neural network training problem as shown in Sec. III. In Sec. IV we perform several numerical experiments to illustrate the virtues of our algorithm and its limitations. A slightly different perspective on the foundations of our method, which offers more flexibility in the implementation, is discussed in Sec. V. Our conclusions are reported in Sec. VI, together with a discussion of other possible applications. These are comparisons of different Monte Carlo generators and data validation algorithms.

II. CONCEPTUAL FOUNDATIONS

Consider repeated measurements $\mathcal{D}=\{x_i\}$, $i=1,\ldots,\mathcal{N}_{\mathcal{D}}$ of a d-dimensional random variable x, and let $n(x|\mathrm{R})$ be its differential distribution as predicted by the reference model "R". Here and in what follows we denote as differential distribution the probability density function (PDF) of x normalized to the total number of expected events in the experiment, namely

$$n(x) = NP(x), \qquad N = \int dx n(x). \tag{1}$$

Testing the reference model for compatibility with the observed data set $\mathcal{D}$ unavoidably requires comparison with an alternative hypothesis $n(x|\mathbf{w})$. In general the alternative hypothesis is composite, labeled by a number of free parameters $\mathbf{w}$. We are interested in problems where the distribution according to which the data are truly distributed is similar (in the sense specified in the introduction) to the reference one; hence it is convenient to parametrize $n(x|\mathbf{w})$ in terms of $n(x|\mathrm{R})$. Taking also into account that $n(x|\mathbf{w})$ is necessarily positive and that we will use log-likelihood ratios for hypothesis testing, we best express it as

$$n(x|\mathbf{w}) = n(x|\mathrm{R})e^{f(x;\mathbf{w})}, \tag{2}$$

in terms of a set of real functions $\mathcal{F} = \{f(x;\mathbf{w}),\ \forall\ \mathbf{w}\}$.

Once the set of alternative hypotheses is specified in this parametrized form, the optimal statistical test for the reference model is defined by the Neyman-Pearson construction [65], based on the maximum likelihood principle. The idea is to compare the reference with the best-fit distribution $n(x|\hat{\mathbf{w}})$, obtained at the point $\mathbf{w} = \hat{\mathbf{w}}$ that maximizes the likelihood. This leads to the test statistic

$$\begin{aligned} t(\mathcal{D}) &= 2\log\left[\frac{e^{-N(\hat{\mathbf{w}})}}{e^{-N(\mathrm{R})}}\prod_{x\in\mathcal{D}}\frac{n(x|\hat{\mathbf{w}})}{n(x|\mathrm{R})}\right] \\ &= -2\underset{\{\mathbf{w}\}}{\mathrm{Min}}\left[N(\mathbf{w}) - N(\mathrm{R}) - \sum_{x\in\mathcal{D}} f(x;\mathbf{w})\right], \end{aligned} \tag{3}$$

where $N(\mathrm{R})$ is the expected number of events in the reference model and $N(\mathbf{w})$ is the expected in the alternative hypothesis, namely

$$N(\mathbf{w}) = \int dx n(x|\mathbf{w}) = \int dx n(x|\mathrm{R})e^{f(x;\mathbf{w})}. \tag{4}$$

In order to associate a probability to the value of t (t_{obs}) obtained with the observed data set, the PDF of t in the reference hypothesis needs to be computed by repeatedly evaluating t on a large sample of toy data sets. From this distribution we obtain the observed p value

$$p_{\mathrm{obs}} = \int_{t_{\mathrm{obs}}}^{\infty} dt P(t|\mathrm{R}), \tag{5}$$

defined as usual as the probability that the reference model produces a data set that is more in tension with itself (has larger t) than the observed data.

The basic idea of the present paper is to parametrize the alternative hypothesis with neural networks. We take $f(x;\mathbf{w})$ to be fully connected neural networks, with free parameters $\mathbf{w}$ that correspond to the weights and biases of the network. In order to turn this idea into a concrete algorithm, the only missing step is to show how the minimization in Eq. (3) can be transformed into a neural network training problem. This step is taken in Sec. III, while here we further elaborate on the conceptual foundations of our method and on the comparison with existing approaches. A brief introduction to neural networks is reported in the Appendix.

A. Model-dependent tests

The Neyman-Pearson formula in Eq. (3) makes clear that the problem of searching for departures from the reference model expectations (i.e., for new physics) merely reduces to the one of selecting an appropriate alternative hypothesis. Different choices produce different test statistics, with widely different performances. One extreme situation is when compelling theoretical arguments allow us to select a single (simple) alternative hypothesis "NP," with no free parameters, for how new physics should look like. In this case Eq. (3) reduces to

$$t_{\mathrm{id}}(\mathcal{D}) = 2\log\left[\frac{e^{-N(\mathrm{NP})}}{e^{-N(\mathrm{R})}}\prod_{x\in\mathcal{D}}\frac{n(x|\mathrm{NP})}{n(x|\mathrm{R})}\right]. \tag{6}$$

According to the so-called Neyman-Pearson lemma [65], t_{id} is the optimal discriminant between the reference and the new physics hypotheses. It is the one that produces the smallest median p value if NP is the true distribution of the data sample.[1] We denote this test statistic as "ideal" because it is the one which is most suited to discover data departures from the reference model, but we can use it only when the true data distribution is known *a priori*.

In the following we employ the ideal test statistic as a figure of merit to assess the performances of our method. However apart from this it is clear that it cannot play a role in the design of model-independent new physics searches, where the goal is to be as agnostic as possible on the alternative hypothesis. Notice indeed that any unjustified assumption on the alternative hypothesis can result in complete loss of sensitivity. For instance suppose that an ideal test is constructed by taking NP to be a narrow resonant peak in an invariant mass distribution, on top of a smoothly falling SM background. The distribution ratio $n(x|\mathrm{NP})/n(x|\mathrm{R})$ appearing in Eq. (6) is nearly equal to 1 (hence its log is zero) in the whole mass range, aside from a narrow region around the resonance mass where it is larger. Therefore only the events that fall in that region contribute to t. This is perfectly fine if the resonance is present in the data just as we predicted it, because in this case signal events will fall in that region producing a large t and in turn a small p value. However if the resonance mass is different from the one we assumed, signal events will fall outside that region and they will not contribute to t. Therefore even if the resonance truly exists in the data, the ideal test would completely miss it.

Several ways exist to mitigate the model dependence of the ideal test, still remaining within the domain of "partially model-dependent" new physics searches. For instance the BumpHunter [2] approach essentially employs a composite alternative hypothesis with three free parameters that correspond to the resonance production rate, width and mass. The maximum likelihood fit to the parameters gives a $n(x|\hat{\mathbf{w}})$ distribution which resembles the one of the true peak, making signal events automatically fall in the region

[1]The theorem says that the condition $t_{\mathrm{id}} > t_c$ defines the critical region with highest power $1-\beta \equiv P(t_{\mathrm{id}} > t_c|\mathrm{NP})$ at given size $\alpha \equiv P(t_{\mathrm{id}} > t_c|\mathrm{R})$ [1]. This statement coincides with the one above because $1-\beta$ is a monotonically increasing function of α and the median p value is the value of α that corresponds to $\beta = 1/2$.

where $n(x|\hat{\mathbf{w}})/n(x|\mathrm{R})$ is large such that their contribution to t is large. This method ensures good sensitivity to a generic resonance, but of course it is completely blind to signals that are nonresonant or that display a resonant peak in a different kinematical variable than the one that has been selected for the test. More generally one can construct tests based on signal topologies, by assuming the production of a certain type of particle (or particles) with certain decay chains and modeling the production and the decay in terms of phenomenological parameters.

B. Model independence and neural networks

We call "model-independent" a new physics search where the alternative hypothesis does not follow from physical considerations, but rather it is selected for technical convenience, with the aim of defining a test that is sensitive to the largest possible variety of putative signals. We have seen that being able to mimic the true underlying distribution is essential for a successful test. Therefore flexibility, i.e., the ability to approximate many functions, is the first important requirement on the set of functions $\mathcal{F}$ that define the alternative distribution through Eq. (2). Piecewise constant functions are the most standard and widely employed approximants. Hence it is not surprising that this choice of $\mathcal{F}$ produces the binned histogram goodness-of-fit test,[2] which is the simplest approach to model-independent new physics searches. This test is constructed by dividing the space of observations in bins and taking $\mathcal{F}$ to assume a constant value w_α in each bin $\alpha = 1, \ldots N_{\mathrm{bin}}$. Since each w_α is an independent parameter, the minimization in Eq. (3) can be trivially performed analytically, giving

$$t_{\mathrm{gof}}(\mathcal{D}) = 2\sum_{\alpha=1}^{N_{\mathrm{bin}}}\left[N_\alpha(\mathrm{R}) - O_\alpha + O_\alpha \log\frac{O_\alpha}{N_\alpha(\mathrm{R})}\right], \qquad (7)$$

where O_α is the number of counts observed in each bin and $N_\alpha(\mathrm{R})$ denotes the expected number in the reference model hypothesis.

The binned histogram method suffers from well-known limitations, the first one being the arbitrariness in the choice of the binning. A reasonable prescription is to employ the smallest bin size compatible with the experimental resolution on the variable of interest. The second and more severe limitation is that the reach of the goodness-of-fit method is reduced by histogram bins that are in good agreement with the reference model. This point is conveniently illustrated by taking the limit where the number of countings is large in each bin, such that O_α are Gaussian distributed and Eq. (7) reduces to the χ^2 formula. Nondiscrepant bins are those where the true model coincides with the reference one; therefore their total contribution to t follows the distribution that is expected in the reference model, a χ^2 with a number of degrees of freedom (d.o.f.) equal to the number of nondiscrepant bins. The mean and the variance of the nondiscrepant contribution are thus equal to the number of nondiscrepant bins. Instead each bin where there is a discrepancy obviously contributes on average more than a nondiscrepant bin; however if there are only a few of them, their total contribution can be much smaller than the one of the nondiscrepant bins and not appreciably change the total value of t.

Removing nondiscrepant bins improves the sensitivity of the test. Hence the binned histogram goodness-of-fit method only works if applied to a restricted set of bins, i.e., to restricted signal regions that have been selected on the basis of prior expectations on the putative signal. Needless to say, the test loses any sensitivity if these expectations are not met by the actual signal.

As mentioned in the introduction, the problem of nondiscrepant bins is not at all an academic one. Existing constraints on new physics models tell us that the vast majority of the data collected in present and future high-energy physics and cosmology experiments will agree with the reference model (i.e., the SM and ΛCDM, respectively). Still we are unable to identify sharply and systematically the data where new physics cannot be present, so ideally the whole set of data will have to be employed in the analysis. This will produce enough nondiscrepant bins to wash out essentially any signal that we might expect. Nonetheless the limitations of the binned histogram method can be partially amended, usually at the price of introducing some amount of model dependence. Approaches based on binned histograms include SLEUTH at D0 [3,4], searches at H1 [5,6], the VISTA and SLEUTH algorithms at CDF [8,9], the CMS algorithm MUSiC [10,11], ATLAS general searches [12–14], and Ref. [7].

Here however we want to explore a different direction by questioning the starting point of the construction, i.e., the choice of $\mathcal{F}$ as piecewise constant functions. We instead define $\mathcal{F}$ as an artificial neural network. It is quite easy to argue against piecewise constant functions and in favor of neural networks and we are not the first ones to do it [61–63]. Neural networks are often introduced exactly as a convenient alternative to binned histograms for the estimation of distributions.[3]

The first argument is that piecewise constant functions are discontinuous and rapidly oscillating. The best fit to the data,

$$f(x;\hat{\mathbf{w}}) = \left\{\log\frac{O_\alpha}{N_\alpha(\mathrm{R})} \quad \text{if } x \in \mathrm{bin}_\alpha, \quad \text{for } \alpha = 1, \ldots, N_{\mathrm{bin}}\right\}, \qquad (8)$$

[2]As the name suggests, this test is typically discussed (see e.g., [66]) in the context of parameters fitting, where the histogram is employed to fit a number "m" of parameters that characterize the expected distribution.

[3]We thank G. Cowan for explaining this so clearly in his lecture [67].

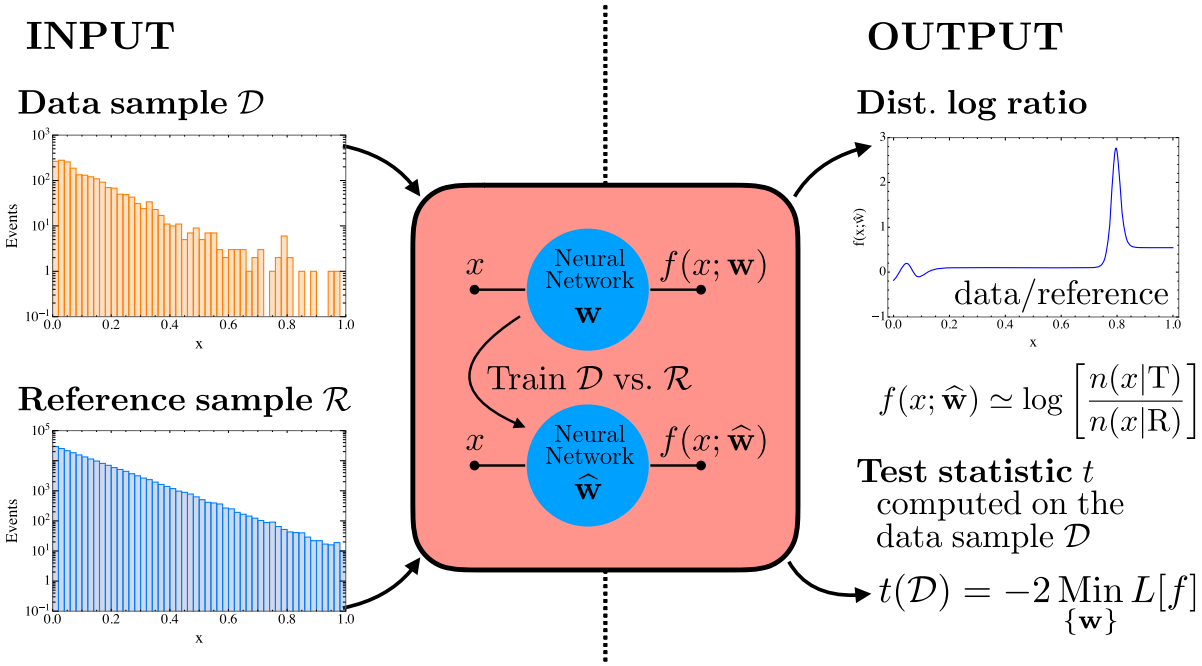

FIG. 1. A schematic representation of the implementation of our strategy.

can have large gradients, which randomly assume positive and negative values in adjacent bins, because of statistical fluctuations. Functions of this sort are not at all credible hypotheses on how the true distribution really looks like. Nevertheless these are the ones that we compare with the reference model when we carry out the goodness-of-fit test. Neural networks are on the contrary smooth functions.

The second advantage of neural networks is that they are more "efficient" approximants. Consider a peak of width $\sigma \ll 1$ in the distribution of a one-dimensional variable. Reproducing this feature requires a number of bins, i.e., of free parameters, of order $1/\sigma \gg 1$.[4] A neural network can instead reproduce (see for instance the Appendix and Ref. [68] for a pedagogical introduction) an arbitrarily sharp peak with only three neurons, i.e., with a limited number of parameters.

Last, but not least, there is the problem of the curse of dimensionality. The number of events that are needed to approximate a function by means of an histogram grows exponentially with the dimensionality of the variable x. While a complete proof is still missing, evidence suggests (see for instance [58–60]) that neural networks can break the curse of dimensionality, requiring fewer events to approximate multivariate distributions. This is of course an extremely desirable property because we would like to search for new physics employing as many variables as possible, reducing in this way the risk of losing sensitivity because of an erroneous choice of observables. On the other hand we have at our disposal a limited number of events to train the neural network.

III. THE ALGORITHM

The algorithm aims at comparing a given data sample $\mathcal{D} = \{x_i\}$, $i = 1, \ldots, \mathcal{N}_\mathcal{D}$, with the reference model prediction for the distribution of x, $n(x|\mathrm{R})$. Normally the prediction does not come in analytical form but rather in the form of a reference sample $\mathcal{R} = \{x_i\}$, with $i = 1, \ldots, \mathcal{N}_\mathcal{R}$, which is distributed according to the reference model. One data and one reference sample are thus the inputs of our algorithm, which produces as output the test statistic $t(\mathcal{D})$ in Eq. (3) and the best-fit log-ratio $f(x;\hat{\mathbf{w}})$. The former quantity will eventually be employed to construct the hypothesis test and turned into a p value as explained at the beginning of Sec. II. The latter function measures the data disagreement with observation locally in phase space. It can thus be employed to select the most discrepant data for further investigation and to perform a number of sanity checks. A schematic representation of the algorithm is shown in Fig. 1. A summary of the notation introduced in Sec. II and in the remainder of this section can be found in Table I.

In the construction of the algorithm we make no explicit assumption on how the reference sample is produced; however we do assume that it is quite large, e.g., $\mathcal{N}_\mathcal{R} = 100N(\mathrm{R})$, in order to eliminate its statistical fluctuations. This is not an issue if the reference sample is produced by a first-principles Monte Carlo event generator, but it might become a problem if instead the reference sample is obtained by extrapolation from a control region. In this case the impact of statistical fluctuations in the

[4]A similar estimate applies if we take $\mathcal{F}$ to be the Fourier series. Extending the series up to frequencies of order $1/\sigma \gg 1$ is needed to see the peak.

TABLE I. Summary of notation.

	Distributions
$n(x\|\mathrm{R})$	Distribution of the variable x in the reference model R
$n(x\|\mathrm{NP})$	Distribution of the variable x in the new physics model NP
$n(x\|\mathrm{T})$	True distribution of x
$n(x\|\hat{\mathbf{w}})$	Distribution of x estimated by the neural network (NN)
	Events
$N(\mathrm{R})$	Number of expected events in the reference model R
$N(\hat{\mathbf{w}})$	Number of events in the data estimated by the NN
	Test statistic
$t(\mathcal{D})$	Test statistic computed by the NN on the data sample $\mathcal{D}$
$t_{\mathrm{id}}(\mathcal{D})$	Ideal test statistic (requires prior knowledge of the signal)
$P(t\|\mathrm{R})$	Probability distribution of the test statistic t in the reference model R
$P(t\|\mathrm{NP})$	Probability distribution of the test statistic t in the new physics model NP
	Normalization
$\int n(x)dx = N$	$n(x)$: Event distribution
$\int P(x)dx = 1$	$P(x)$: Probability distribution

reference sample, which we ignore in what follows, should be duly taken into account.

Two problems need to be solved in order to evaluate the test statistic in Eq. (3) with the elements at our disposal. The first one is that $n(x|\mathrm{R})$ is not known in analytical form; hence we do not know how to compute the integral for $N(\mathbf{w})$ in Eq. (4). The second one is that in order to carry out the minimization numerically, exploiting the powerful existing tools for neural network training, we should first express Eq. (3) as a loss function. However we can solve both problems at the same time. We estimate $N(\mathbf{w})$ by the Monte Carlo method, namely we write[5]

$$N(\mathbf{w}) = \frac{N(\mathrm{R})}{\mathcal{N}_{\mathcal{R}}} \sum_{x\in\mathcal{R}} e^{f(x;\mathbf{w})}. \tag{9}$$

Equation (3) thus becomes

$$\begin{aligned} t(\mathcal{D}) &= -2\min_{\{\mathbf{w}\}}\left[\frac{N(\mathrm{R})}{\mathcal{N}_{\mathcal{R}}}\sum_{x\in\mathcal{R}}(e^{f(x;\mathbf{w})}-1) - \sum_{x\in\mathcal{D}} f(x;\mathbf{w})\right] \\ &\equiv -2\min_{\{\mathbf{w}\}} L[f(\cdot,\mathbf{w})], \end{aligned} \tag{10}$$

[5]There is an equality in the equation that follows because we assume a large enough reference sample to reduce the Monte Carlo integration error to a negligible level.

where L has precisely the form of a loss function. It can be written as a single sum over events by introducing a target variable y which is set to 0 for the events in $\mathcal{R}$ and to 1 and for those in $\mathcal{D}$. Explicitly, we have

$$L[f] = \sum_{(x,y)}\left[(1-y)\frac{N(\mathrm{R})}{\mathcal{N}_{\mathcal{R}}}(e^{f(x)}-1) - yf(x)\right]. \tag{11}$$

The minimization of L with respect to the neural network parameters $\mathbf{w}$ can thus be carried out as a standard supervised training process. The test statistic is simply minus 2 times the loss at the end of training. The trained neural network $f(x;\hat{\mathbf{w}})$ is the maximum likelihood fit to the data and reference distributions log-ratio. It is the best approximant, within the neural network parametrization, of the true underlying data distribution $n(x|\mathrm{T})$:

$$f(x,\hat{\mathbf{w}}) \simeq \log\left[\frac{n(x|\mathrm{T})}{n(x|\mathrm{R})}\right]. \tag{12}$$

Notice that training unavoidably requires some sort of regularization because our loss function (11) is unbounded from below; namely it approaches negative infinity if f diverges at some value of x belonging to the $\mathcal{D}$ (i.e., $y = 1$) class. Notice that the problematic situation occurs only when the divergence in f is sharply localized, such that $f(x)$ stays finite for all $x \in \mathcal{R}$. Otherwise the positive exponent that we have in the loss function for the $\mathcal{R}$ (i.e., $y = 0$) class overcompensates the negative divergence. We avoid these dangerous configurations by enforcing an upper bound (set by the so-called "weight clipping" parameter W) on the absolute value of each weight. This forbids the neural network to diverge and to produce sharp features on a scale $\Delta x \lesssim 1/W$. Given that infinitely sharp features cannot show up in the true distribution because of experimental resolution smearing, for any concrete problem it will be possible to choose W large enough not to limit the approximation capabilities of the neural network. We use $W = 100$ in the following.

To obtain a p value that tests the agreement between data and the reference model we proceed as discussed at the beginning of Sec. II. First we train the network using the actual data sample and a large reference sample distributed according to the R model, as pictorially shown in Fig. 1. This gives us the observed value of the test statistic t_{obs}. Then we repeat the training on many toy experiments generated according to the reference distribution; i.e., we use the same reference sample, network architecture and training parameters as before, but we substitute the data sample with toy reference samples. For each of these samples we compute t and thus obtain $P(t|\mathrm{R})$. The p value is then computed in the usual way [see Eq. (5)].

Before moving forward it is worth clarifying some assumptions that our method relies on. First, we assumed knowledge of the expected number of events, $N(\mathrm{R})$, which

appears in the definition of the loss function in Eq. (11). This can be problematic because the total event rate is often not well predicted by high-energy physics simulations. The simplest way out is to take $N(\mathrm{R})$ equal to the number of data that have been observed in the actual experiment. This is conservative as it assumes perfect agreement of the observed number of events with the reference model prediction. In what follows we keep working under the assumption that $N(\mathrm{R})$ is known *a priori*, but this assumption can be easily eliminated as previously explained. Furthermore in real-life applications (and in most of the examples we discuss) the signal component is small and the total number of events is not a significant discriminant.

Much more problematic is assuming the Monte Carlo to provide a perfect description of the reference distribution shape. This is not realistic because Monte Carlo generators are subject to systematic uncertainties, which for large enough statistics unavoidably result in a significant tension with the data. These uncertainties are routinely modeled as nuisance parameters and treated with the profile likelihood ratio formalism [69,70]. The basic idea is that we should first of all identify the value of the nuisance parameters that best describe the data, taking of course also into account auxiliary measurements and not only the data set of interest. Next we use these values in the reference distribution prediction of Eq. (3). A proper tune of the reference model Monte Carlo to the data is a prerequisite for any new physics search; hence this problem is in some sense orthogonal to the one that we are addressing. However the interplay and the possible synergies between the two aspects should be carefully studied. Especially the possibility of incorporating in the network the fit to data of some of the nuisance parameters to reduce systematic uncertainties. This is left to future work.

A. Summary of the algorithm

(1) Train the network on the data, using the loss function in Eq. (11).
   (a) *Input*.—One data sample $\mathcal{D}$ and one reference sample $\mathcal{R}$.
   (b) *Output*.—(i) Value of the test statistic on the data sample t_{obs} and (ii) log-ratio of the data and reference probability distribution functions $f(x;\hat{\mathbf{w}}) \simeq \log[n(x|\mathrm{T})/n(x|\mathrm{R})]$.

(2) Generate several toy data samples "$\mathcal{D}$" that mimic the expected outcome of the experiment if the reference model is true. Train the *same* network on these toy data samples, using all the same parameters for training.
   (i) *Input*.—The same reference sample as above and the toy data samples.
   (ii) *Output*.—Distribution of the test statistic in the reference hypothesis $P(t|\mathrm{R})$. See e.g., Fig. 3.

(3) Use $P(t|\mathrm{R})$ and t_{obs} to compute the p value: $p = \int_{t_{\mathrm{obs}}}^{\infty} P(t|\mathrm{R})dt$. See e.g., Figs. 4 and 6, where the p values are reported as Z scores. In those figures we plot a whole set of p's obtained on hundreds of different data samples to assess the performance of our algorithm.

(4) If p is sufficiently small to signal a tension with the reference hypothesis, use the log-ratio $f(x;\hat{\mathbf{w}})$ to learn the nature of the discrepancy.

B. Performances on a simple case study

We now turn to a first illustration of the performances of our algorithm. We start with a simple example, which we study more quantitatively and systematically in the next section. We consider an univariate problem $x \in [0,1]$. The reference model (or background) is a steeply falling exponential distribution

$$P(x|\mathrm{R}) \propto e^{-8x}, \quad \text{and} \quad N(\mathrm{R}) = \int_0^1 dx n(x|\mathrm{R}) = 2000 \equiv B. \tag{13}$$

We consider the possible presence in the data of a small resonant signal component $S = 10$, distributed as

$$P(x|\mathrm{S}) \propto e^{-\frac{(x-\bar{x})^2}{2\sigma^2}}, \text{ with } \bar{x} = 0.8 \text{ and } \sigma = 0.02. \tag{14}$$

The new physics distribution for x therefore is

$$n(x|\mathrm{NP}) = \frac{S+B}{1+S/B}\left[P(x|\mathrm{R}) + \frac{S}{B}\mathrm{P}(x|\mathrm{S})\right], \tag{15}$$

with a signal over background ratio $S/B = 5 \times 10^{-3}$ and a total number of expected events $N(\mathrm{NP}) = S + B = 2010$. The model is depicted in the left panel of Fig. 5. We generate one large ($\mathcal{N}_{\mathcal{R}} = 200\,000$) reference sample $\mathcal{R}$ according to the reference PDF and several data samples $\mathcal{D}$ that follow either the reference or the new physics distributions. The number of data events is selected at random taking into account Poisson fluctuations around the expected numbers $N(\mathrm{R}) = 2000$ and $N(\mathrm{NP}) = 2010$. We train a four-neuron (1,4,1) neural network[6] on each data set and we obtain the corresponding $t(\mathcal{D})$ and $f(x;\hat{\mathbf{w}})$ as previously described. Since $n(x|\mathrm{R})$ is fully known, in our toy example we can also compute the best-fit distribution $n(x|\hat{\mathbf{w}})$ using the log-ratio learned by the neural network in Eq. (2). An initial learning rate of 10^{-3} is chosen, and training is stopped after 150 000 rounds. The results are displayed in Fig. 2 for six representative data samples. The ones on the first and on the second row have been obtained from the NP and from the R distributions, respectively.

[6]The notation for the neural network architecture is explained in more detail in the Appendix. The (1,4,1) network has one-dimensional input and output and a hidden layer with four neurons.

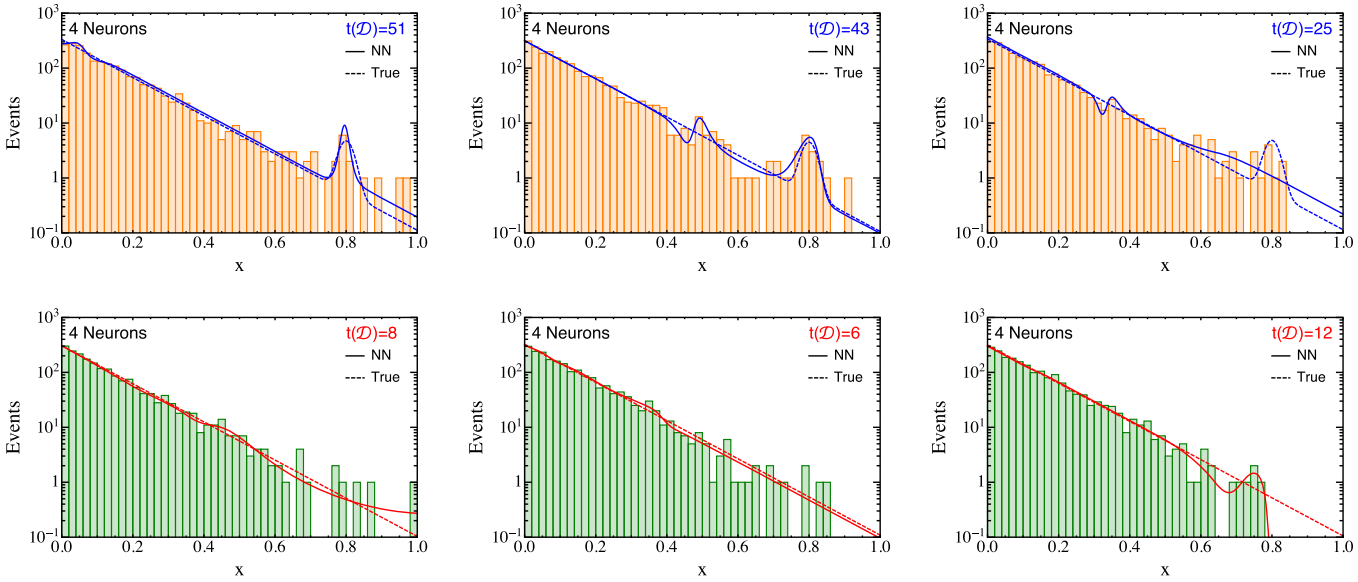

FIG. 2. The distribution learned by a neural network with a single four-neuron hidden layer (solid line), compared with the distribution used to generate the data (dashed line) and the binned histogram of the training data set. The value of the test statistic $t(\mathcal{D})$ obtained by the network is reported in the upper right corner of each plot. The higher values of $t(\mathcal{D})$ in blue signal that the network is discriminating between data sets containing new physics (top row) and data sets following the reference hypothesis (bottom row).

The figure illustrates a number of interesting points. First of all, we see that in all cases the distribution learned by the neural network is very much correlated with the data sample that was used for training. Still it does not follow the data too closely, producing smooth curves that are quite "credible" hypotheses on the true underlying distribution. This should be contrasted with the discontinuous piecewise constant distribution, i.e., the envelope of the histogram, that one would effectively rely on if the same data sets were studied with the binned histogram method. We also see that in the bulk region, i.e., at small x, the neural network is able to reproduce very accurately the true distribution, thanks to the large statistic. This is important because mismodeling the bulk would produce a large spurious contribution to t, that would obscure the genuine signal in the tail. The NP-generated data samples produce an excess in the tail of the distribution, which is more or less in agreement with the true peak at $x = 0.8$, depending on how many events happened to fall in that region. The distributions obtained with the background data samples can also depart considerably from the reference distribution (which is the true one for background samples); however the departures occur in regions where only few events are present and hence they give a limited contribution to t. We also remark that the size of t is in clear correspondence with how different the reference distribution and the distribution learned by the neural network are. The six values shown in the figure already indicate that t possesses some discriminating power between the signal and the background. We study this systematically in the following section.

IV. NUMERICAL EXPERIMENTS

In this section we test our method by performing several numerical experiments on one- and two-dimensional samples. A summary of the notation needed to interpret the figures in this section can be found in Table I. In all the new physics scenarios discussed here we have generated hundreds of toy data samples to assess the median significance of the algorithm and its correlation with the ideal significance. So the single value of the test statistic, t_{obs}, that one would observe in a real experiment is presented as a distribution given a putative new physics model. Correspondingly the single observed p value (or Z score) becomes an entire distribution.

The numerical experiments performed here have been selected with the aim of illustrating the following aspects.

(i) *Model independence.*—The goal of our approach is to be sensitive to a signal that is unknown *a priori*. Ideally it should detect any kind of new physics that could be present in the data. We verify this through several examples in Sec. IV A.

(ii) *(In)sensitivity to cuts.*—It is impossible to identify the appropriate search region without prior assumptions on the nature of the signal. Furthermore we argued in Sec. III that the loss of sensitivity due to the presence of a large number of data points in agreement with the reference model is the main limitation of the binned histogram goodness-of-fit approach. In Sec. IV B we show that instead the performances of our method do not depend on whether a favorable signal region is selected based on prior knowledge of the signal.

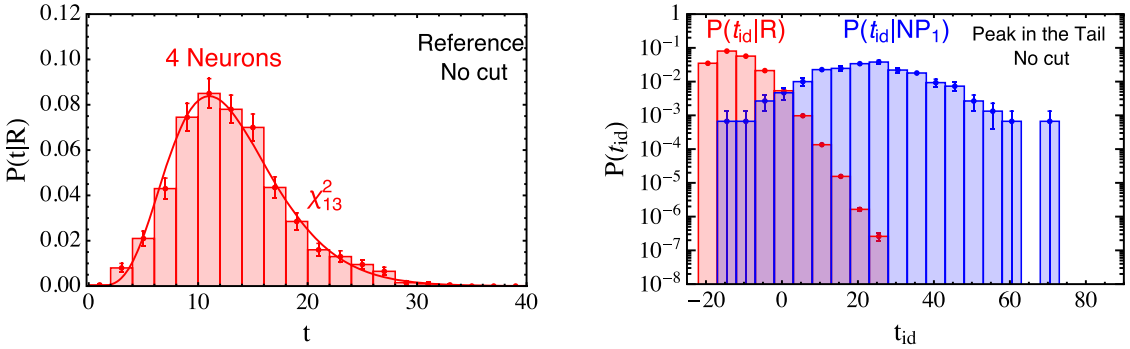

FIG. 3. Left panel: Test statistic distribution in the reference model, compared with the χ^2 PDF with 13 d.o.f. The relation between the χ^2 and our test statistic is discussed in Secs. IV A and IV D. Right panel: Ideal test statistic distribution in the reference and in our first new physics scenario: NP_1.

(iii) *Two dimensions.*—We apply our method to two-dimensional distributions, with the aim of studying to what extent the reach deteriorates if the relevant variable that differentiates the signal from the background is not known *a priori*. The results are presented in Sec. IV C.

(iv) *Dependence on hyperparameters.*—The neural network architecture, the initial learning rate and the number of training rounds are the free parameters of our algorithm, collectively denoted as hyperparameters. We study the performance dependence on these parameters in Sec. IV D.

Before discussing these points, a general methodological remark is in order. It is not completely straightforward to quantify the performances of our method. Clearly in each example we can compute the median p value of our test, but this is a valid figure of merit only in comparison with some independent quantification of the actual difference between the reference distribution and the new physics that we assumed in the example. This aspect is particularly important for comparing the sensitivity of our test to new physics signals of different nature, for instance comparing the sensitivity to a peak with the one to an anomalous growth of the distribution in the tail. What we need is to assess in absolute terms how difficult it is to discover new physics in the example under consideration. For this purpose we employ the "ideal" test, defined in Eq. (6). Namely for each toy example we evaluate t_{id}, defined by exploiting the complete knowledge of the new physics distribution, on a large set of reference-distributed toy data samples. This gives us the PDF of t_{id} in the reference hypothesis. Next we use this distribution to compute the ideal p value p_{id} for each one of the toy data samples generated according to the new physics distribution. The ideal p value can then be compared with the one obtained with our test, either individually on each sample or globally in terms of the median over repeated toys. Notice that the ideal test is the one with smallest median p value, since it is obtained using a complete knowledge of the signal. Therefore we cannot hope to obtain a similar significance with our test, where we assume no previous knowledge of the signal whatsoever. Still we can assess the success or failure of our method by how much significance we lose in comparison with the ideal test.

A. Model independence

In all the examples considered in the present subsection, $x \in [0, 1]$ and its reference distribution is the exponential in Eq. (13). Physically we might interpret x as an invariant mass measured at the LHC, with its steeply falling SM distribution modeling parton luminosities. Since the reference distribution is the same for all example signals, the preparatory stages of our test can be carried out once and for all. These consist in generating a $\mathcal{N}_{\mathcal{R}} = 200\,000$ reference sample and in computing the test statistic PDF by training the neural network on toy Monte Carlo samples generated according to the reference model. A (1,4,1) neural network is employed, the initial learning rate is 10^{-3} and 150 000 training rounds are performed using the RMSprop algorithm [71]. Evaluating $t(\mathcal{D})$ on 1000 reference-distributed toys produces the PDF in the left panel of Fig. 3. Thanks to this distribution we can compute the p value associated with $t(\mathcal{D})$ evaluated on the data samples generated according to the new physics distribution.

Notice however that we can meaningfully estimate the p value only if t does not exceed the maximal value obtained with our toy Monte Carlo samples. If t is larger, we can only set a lower bound on the p value, which we obtain from the 68% upper limit for 0 successes (binomially distributed) and N trials, i.e., $p < 1 - (0.32)^{1/N}$. With the $N = 1000$ Monte Carlo samples at our disposal, this corresponds to $p < 1.1 \times 10^{-3}$ or to a significance $Z > 3.05\sigma$.[7] However $P(t|\mathrm{R})$ is quite well approximated by a χ^2 distribution with 13 d.o.f., which is not surprising because 13 is the number of free parameters of the (1,4,1) network that we are employing. We return to this point in

[7]We adopt the standard definition $Z = \Phi^{-1}(1 - p)$, where Φ^{-1} is the quantile of the Gaussian distribution.

Sec. IV D; for the moment we just exploit this fact to extend our estimate of the significance to values of t above the maximum. Namely, for those we report the estimate of the significance obtained with the χ^2 approximation instead of the lower bound obtained with the toys.

The first new physics model that we discuss (dubbed NP_1 in what follows) is the one introduced in Eqs. (14) and (15). It mimics the presence of a resonance in the tail of the SM invariant mass distribution. We generate 300 toy Monte Carlo samples according to the new physics distribution in Eq. (15), and we train a neural network for each, with the same algorithm used for the reference-distributed data. The resulting distribution for t, $P(t|\mathrm{NP}_1)$, is displayed in the right panel of Fig. 4. By comparing with $P(t|\mathrm{R})$ we see that our test statistic has a considerable discriminating power between the two hypotheses. The median t in the NP_1 toy samples is 36, which is slightly above the maximum value that we obtained with the reference data. The median significance for the NP_1 signal hypothesis is thus above 3.05σ, and it can be estimated to be 3.2σ using the χ^2 approximation.

For a better assessment of the performances of our method we compare them to those of the ideal test presented in Sec. II [see the discussion below Eq. (6)]. We estimate the ideal test statistic PDF by means of a very large set of 10 000 000 reference model toy data samples, and we compare it with the values of t_{id} on the 300 new physics data samples with which we trained the network. The result is shown in the left panel of Fig. 3. The sensitivity of the ideal test is as expected much higher than ours. The median t_{id} on new physics samples is 23 and it corresponds to an ideal significance $Z_{\mathrm{id}} = 4.7\sigma$. We can thus conclude that the difference in sensitivity amounts to roughly 1.5σ. This is confirmed if we look at the correlation between Z_{id} and Z on each individual data sample, reported in the right panel of Fig. 4. Notice that the vertical band of points that seemingly breaks the correlation is an artifact due to new physics samples with a t_{id} that is larger than the maximum t_{id} obtained in the 10 000 000 reference toys. For these samples, a lower bound on Z_{id} of 5.2σ (corresponding to zero observed over 10 000 000 trials at 68% C.L.) is reported in the plot.

The second example (NP_2) is nonresonant new physics, showing up as a quadratic growth with energy in the tail of the reference model distribution. In this case the signal is distributed as

$$P(x|\mathrm{S}_2) \propto x^2 e^{-8x}, \tag{16}$$

and the total expected number of signal event is taken to be $S = 90$. The signal and background are combined to define the NP_2 distribution as in Eq. (15). The model is depicted in the central panel of Fig. 5. The median ideal significance for the chosen value of S equals 4.4σ, very much comparable with the one of the NP_1 signal. This ensures a fair comparison between the two. The performances of our algorithm, shown in the left column of Fig. 6, are essentially identical to those we obtained for NP_1. The median significance is 3.1σ and the correlation between Z_{id} and Z again reveals a significance loss of around 1.5σ.

Finally, we discuss another resonant signal, emerging this time in the bulk of the reference model distribution. The signal distribution is

$$P(x|\mathrm{S}_2) \propto e^{-\frac{(x-\bar{x})^2}{2\sigma^2}}, \quad \text{with} \quad \bar{x} = 0.2, \qquad \sigma = 0.02, \tag{17}$$

and $S = 35$. The model is depicted in the right panel of Fig. 5. The median ideal significance is 4.1σ. We see in the right column of Fig. 6 that accordingly the median significance of our algorithm (2.6σ) is slightly reduced compared to NP_1 and NP_2. The correlation between Z_{id} and Z is equally sharp.

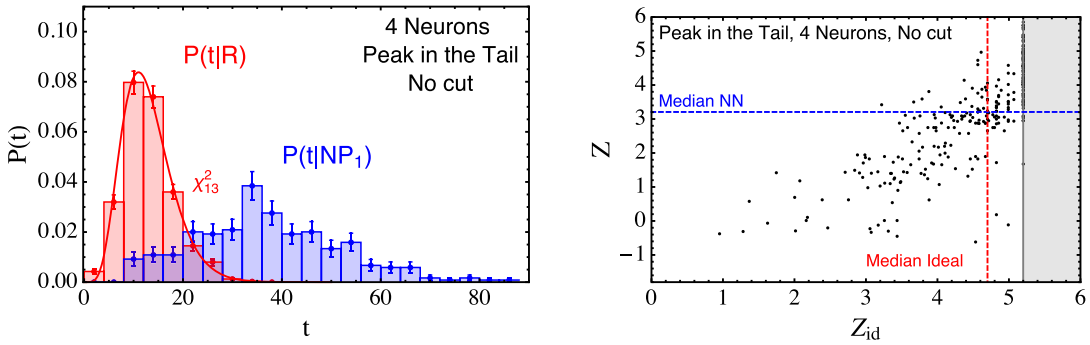

FIG. 4. Left panel: Test statistic distribution in the NP_1 new physics model $P(t|\mathrm{NP}_1)$, compared with the reference one $P(t|\mathrm{R})$. The two models are defined in Eqs. (13) and (14), respectively, and shown in Fig. 5. The larger values of t in $P(t|\mathrm{NP}_1)$ compared to $P(t|\mathrm{R})$ signal that our algorithm is sensitive to this new physics scenario. These two distributions are used to obtain the Z score on the y axis in the right panel. Right panel: Correlation between the significances (expressed in number of σ's) of our test and of the ideal test defined in Sec. II, for the NP_1 model. The gray shaded area corresponds to the region where the ideal significance cannot be computed with the number of toy data sets generated. We also show the median significance of our algorithm (Median NN) and the ideal one.

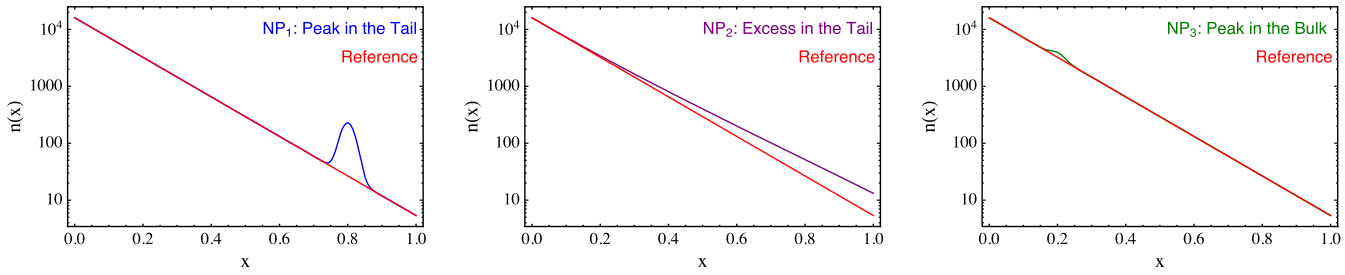

FIG. 5. The distributions of the three new physics models used in this work plus the reference one.

The comparative study of three new physics models carried out in this section provides a clear confirmation of the model-independent nature of our approach.

B. (In)sensitivity to cuts

The point is conveniently illustrated in the NP_1 example. Since the signal is sharply localized at $x = 0.8$, one might expect that restricting the analysis to events in the tail of the distribution, for instance to those with $x > 0.3$ or $x > 0.5$, will give us a better reach. This would have indeed been the case for the goodness-of-fit test. Our method is instead insensitive to the cut, as Fig. 7 shows.

The median significance is 3.1σ for both $x > 0.3$ and $x > 0.5$. Also the $Z_{\rm id}$-Z correlation plot that we do not show here is essentially identical to the one without a cut displayed in Fig. 4. These results have been obtained using the same procedure outlined in the previous section for the case without a cut on x. We employed the same learning rate, training algorithm, number of training rounds and network architecture (a single hidden layer with four neurons). The only change is in the number of expected events. However notice that we were not conceptually obliged to choose the same hyperparameters as in the no-cut case. In particular the smaller number of events might have suggested using a smaller network. It is encouraging that a selection cut does not improve the significance. If our method had been sensitive only in signal-enriched regions ($x > 0.5$ for example, where $S/B \approx 0.3$), we would have not solved the problems that plague the binned histogram test, discussed in Sec. II. Suppose, for concreteness, that we

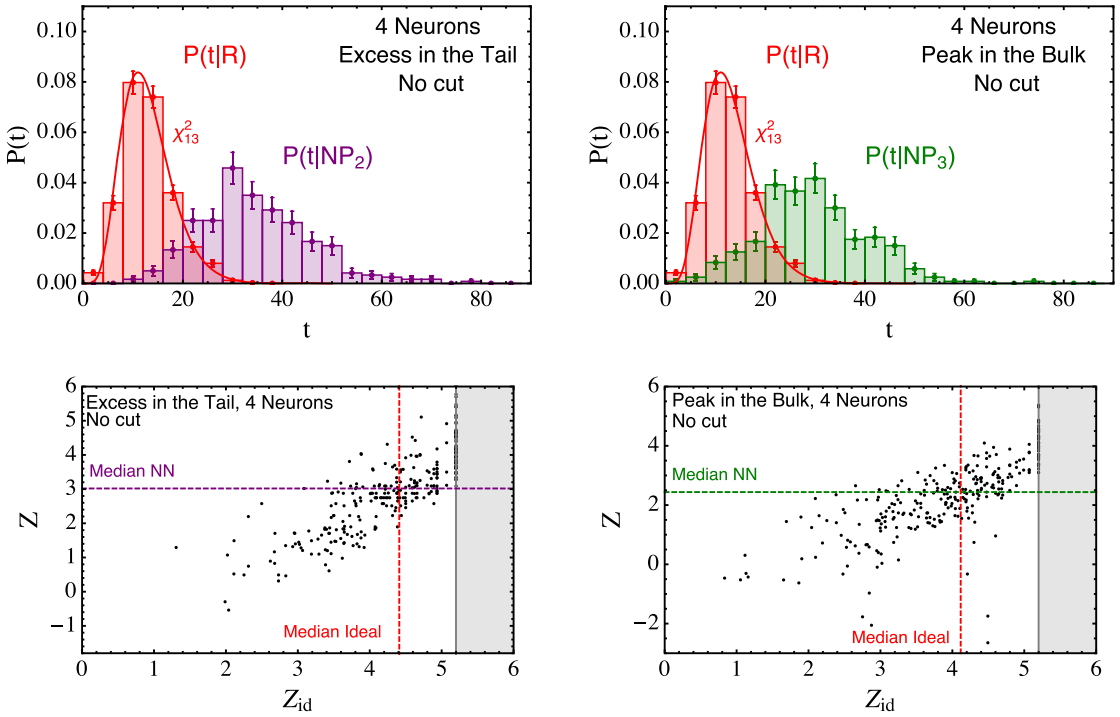

FIG. 6. *Top row*: Test statistic distribution in the NP_2 (left) and NP_3 (right) new physics models, compared with the reference one. The two models are defined in Eqs. (16) and (17). *Bottom row*: Correlation between the significances (expressed in number of σ's) of our test and of the ideal test defined in Sec. II, for the NP_2 (left column) and NP_3 (right column) new physics models. The gray shaded area corresponds to the region where the ideal significance can not be computed with the number of toy data sets generated. We also show the median significance of our algorithm (Median NN) and the ideal one.

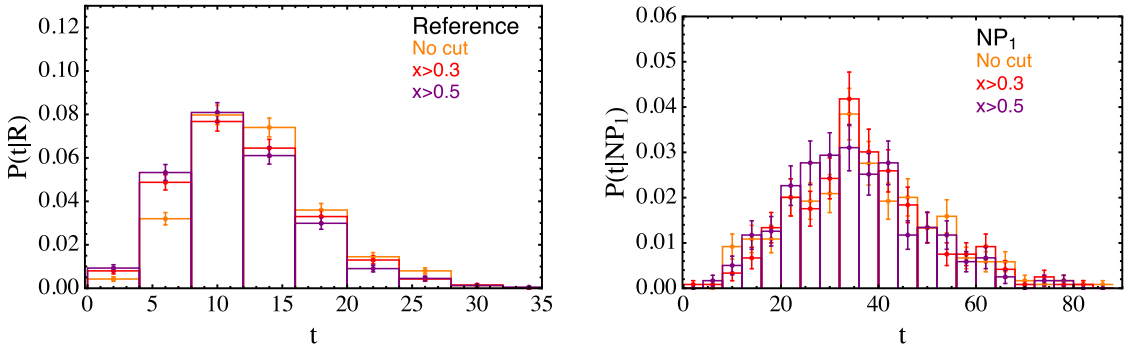

FIG. 7. Left panel: Test statistic distribution in the reference hypothesis, $P(t|\mathrm{R})$, for $x \geq 0, x > 0.3$ and $x > 0.5$. Right panel: Test statistic distribution in the new physics hypothesis NP_1 (narrow peak in the tail) for $x \geq 0, x > 0.3$ and $x > 0.5$. No substantial difference is observed in the distributions of the test statistic. As a consequence the expected reach is independent of the cut.

had analyzed data in the $x > 0.5$ search region, finding a considerable tension with the reference model. The immediate question, related to the look-elsewhere effect [1], would be whether adding data in the $x \in [0, 0.5]$ region would wash out the tension or not. We verified that in our examples this would not be the case, on average, even if new physics is only present at $x > 0.5$. Enlarging the search region to the full $x \in [0, 1]$ range would at most increase the tension, giving us sensitivity to the possible presence of new physics (such as for instance NP_3) that does not show up in the restricted data set.

C. Two dimensions

We now consider a two-dimensional random variable $x = (M, c)$, with $M \in [0, 1]$ and $c \in [-1, 1]$. The variable M is interpreted as the invariant mass, while c is the cosine of the scattering angle in the center of mass frame. These two variables conveniently characterize two-body final states in LHC events. The distributions of M are chosen among the ones that we previously introduced in the univariate examples. Namely, in the reference model M is exponentially distributed as in Eq. (13), while the putative new physics signal is the resonant peak in Eq. (14), duly combined with the background as in Eq. (15). The variable c is uniformly distributed both in the reference and in the new physics model; hence it possesses no discriminating power. This setup makes the comparison between 1D and 2D performances particularly meaningful and straightforward. The results obtained in the previous section can indeed be regarded as those that we have if the two-dimensional data set is analyzed with the prior bias that M is the only relevant variable. The present section instead discusses what we can get without this prior.

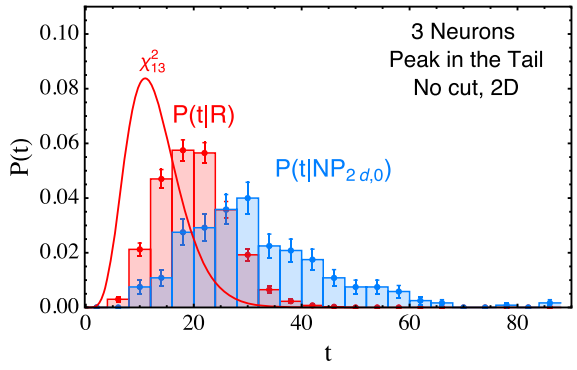

FIG. 8. Test statistic distribution in the $\mathrm{NP}_{2d,0}$ new physics model, compared with the reference one. We expect 2010 events in the new physics model as in the one-dimensional case.

The test statistic distributions are reported in Fig. 8. The results are obtained with a (2,3,1) network, trained with the same initial learning rate, training algorithm and training rounds as before. A considerable loss in sensitivity is observed in comparison with the 1D case in Fig. 3. The significance rarely reaches 3σ, and the median is 1.4σ. The correlation between Z and Z_{id} is less sharp, and large-Z_{id} samples often end up having low significance. This results from the combination of two distinct effects. The first one is that the values of t resulting from the neural network training on new physics samples are significantly smaller, and the second is that t is larger on the reference samples. Let us discuss the two effects separately.

The new physics median t is now 29, while it was 36 in 1D. This result might seem inconsistent, in light of the fact that the 2D network for M and c contains configurations, obtained by setting to zero all the weights for c, that are fully equivalent to a 1D network for M. However the 1D network obtained in this way has a (1,3,1) architecture, while a (1,4,1) network is employed in Fig. 3. A (1,3,1) network in 1D, discussed in the next section, indeed produces a median new physics t of 31, very close to the 2D one.[8] Therefore the new physics median t we find in 2D is not in sharp contradiction with 1D results. Still it is somewhat surprising that it is not larger than the 1D one

[8]In one dimension the smaller new physics t for the (1,3,1) network does not result in a degradation of the sensitivity because the reference model t distribution is also shifted to lower values, as discussed in the next subsection.

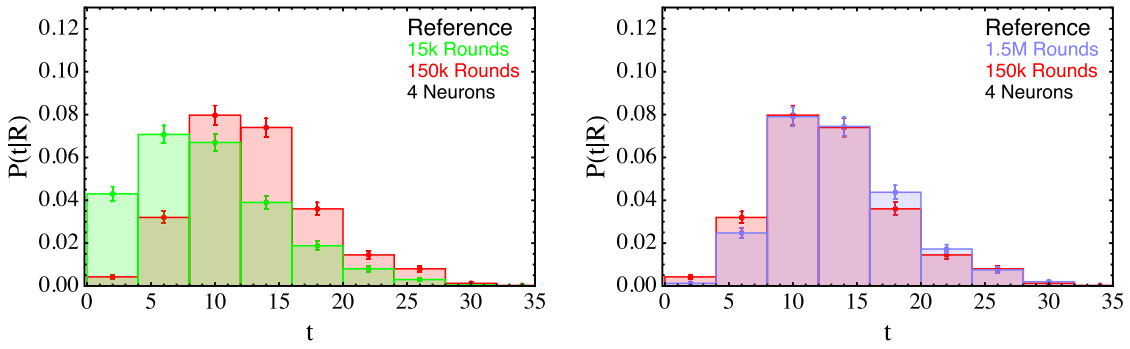

FIG. 9. Test statistic distribution in the reference hypothesis, $P(t|\mathrm{R})$, for networks with one hidden layer and four neurons. Left panel: 15×10^3 training rounds compared to 150×10^3. Right panel: 1.5×10^6 training rounds compared to 150×10^3.

because the weights associated to c should in principle allow one to find a deeper minimum for the loss function. This is what happens on reference model samples, whose 2D distribution is shifted to a much higher value than those in 1D for the (1,3,1) network (see Figs. 8 and 10).

The reference model t distribution is not only shifted with respect to the (1,3,1) network, which follows a χ^2 with nine d.o.f., but also with respect to the χ^2_{13}, in spite of the fact that the (2,3,1) network that we employed has 13 free parameters. We further elaborate on this point in Sec. IV D and in the conclusions.

The result indicates that improvements in the implementation of our method can be made before considering applications to multivariate data sets. There are many possible directions of investigation in terms of training algorithm and network architecture that we believe would improve the sensitivity in higher dimensions. We discuss them in the conclusions. However even with this loss in sensitivity, our method should still be explored as a viable alternative to binned histogram model-independent searches which are dramatically affected by the curse of dimensionality.

Furthermore the concrete impact of the loss in significance that we observe should not be overemphasized. Even if no significant tension is typically found in the 2D data sets under consideration, the signal could still be discovered by running the experiment longer and collecting more events. With twice more luminosity, i.e., $B = 4000$ and $S = 20$, we obtain a median significance of 2.3σ.

D. Dependence on hyperparameters

The aim of this section is to illustrate how the performances depend on the algorithm hyperparameters such as the initial learning rate, the number of training rounds and the architecture of the neural network.

Our method is founded on maximizing a likelihood function proportional to minus our loss function. Therefore the parameters of the training algorithm should be selected as those that produce the smallest loss and, in turn, the largest t in Eq. (10). We verified that lowering the learning rate below our benchmark value of 10^{-3} does not increase t. For higher values the loss oscillates as training proceeds and it does not converge. Similarly we verified that 10 times more training rounds than the 150 000 benchmark do not change the performances. Less training instead would be insufficient. This is shown in Fig. 9 for reference-distributed data. The same is found with new physics samples.

The situation is more interesting if we vary the network architecture. In the left panel of Fig. 10 we show how the test statistic distribution in the reference hypothesis changes with the number of neurons, while keeping the number of training rounds fixed at 150 000. As we increase the free parameters in the network, t increases. This has to be expected in light of the well-known result by Wald and Wilks [72,73] (see also [70] for a more modern discussion), according to which the maximum log-likelihood ratio test statistics is distributed in the asymptotic limit as a χ^2 with a number of d.o.f. which is equal to the number of free parameters in the maximum likelihood fit.[9] In our case the free parameters [i.e., $\mathbf{w}$ in Eq. (10)] are 10 for the (1,3,1) network, 13 for the (1,4,1) network and 31 in the (1,10,1) case. The (1,3,1) and (1,4,1) distributions follow the asymptotic formula with the corresponding number of parameters, while the (1,10,1) distribution is slightly below the expectation. However this is most likely due to insufficient training. With 1.5 million training rounds the (1,10,1) distribution tends to align with the χ^2_{31}, as shown in the right panel of Fig. 10.[10] The limited computing power at our disposal and the need to perform the training thousands of times on toy data sets did not allow us to check if an even longer training would take the (1,10,1) distribution even closer to χ^2_{31}.

It should be noticed that the asymptotic formulas only hold in the formal limit of infinite statistics, and there are no

[9]We are of course referring to the case in which the data are distributed according to the hypothesis that is being tested, i.e., the reference hypothesis in the present case.

[10]More training rounds do not change the distribution for smaller networks, as previously mentioned.

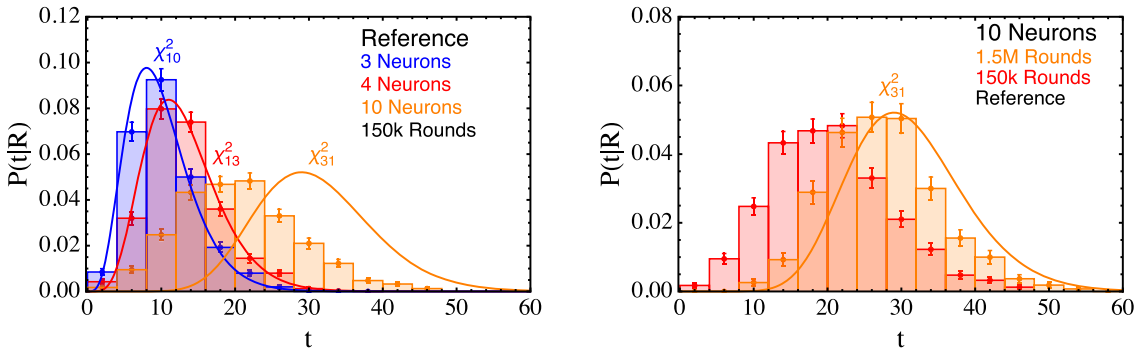

FIG. 10. Left panel: Test statistic distribution in the reference hypothesis, $P(t|\mathrm{R})$, for networks with one hidden layer and three, four or ten neurons, compared to the χ^2 with the same number of d.o.f. as the network. The training parameters are the same for all architectures (15 000 training rounds, 0.001 initial learning rate, RMSprop algorithm). Right panel: Test statistic distribution in the reference hypothesis for the network architecture with ten neurons. We compare the result with 150 thousand and 1.5 million training rounds. The figure shows how our networks reproduce the asymptotic formulas for the test statistic expected from the theorems in [72,73]. However larger networks require more training rounds.

sharp criteria to establish how many events are concretely needed for them to apply. Therefore the agreement we observe is not a consistency check. It simply means that the statistics in our 1D example is sufficient, at least for networks with up to ten neurons, to reproduce the asymptotic distribution. It is legitimate to expect departures from the χ^2 for much larger networks. However we could not verify this fact because the required training time increases with the network capacity, as we have seen. Departures from the χ^2 formula were instead found in the 2D example; see for instance Fig. 8 and Sec. IV C. We discuss in the conclusions why it would be important to develop an understanding of this difference between the 1D and the 2D examples.

More concretely, we are interested to know how the sensitivity of the test depends on the neural network architecture. We find that t increases with the network capacity also for new physics generated samples. The median t in the data samples is 31 for the (1,3,1) network, 36 for (1,4,1) and 56 for (1,10,1). This compensates for the growth of t in the reference model, making the significance roughly invariant. We find a median significance of 3.2σ, 3.1σ and 3σ for the three-, four- and ten-neuron networks, respectively. Notice however that 1.5 million training rounds have to be employed in the ten-neuron case, making the algorithm 10 times slower. With 150 000 rounds we would have obtained a slightly lower significance of 2.7σ.

V. ALTERNATIVE LOSS FUNCTIONS

In Secs. II and III we constructed our algorithm as a straightforward application of the maximum likelihood method. Here we describe an alternative derivation, slightly less direct and conceptually rewarding, which however offers more freedom in the implementation. In particular, it allows us to employ different loss functions than the one in Eq. (11). The starting point is the definition of t in Eq. (3), which we rewrite below for convenience:

$$t(\mathcal{D}) = 2\log\left[\frac{e^{-N(\hat{\mathbf{w}})}}{e^{-N(\mathrm{R})}}\prod_{x\in\mathcal{D}}\frac{n(x|\hat{\mathbf{w}})}{n(x|\mathrm{R})}\right]. \tag{18}$$

This equation instructs us to construct the test statistic as the log-ratio between the reference distribution and the "best-fit" distribution $n(x|\hat{\mathbf{w}})$, obtained from the data set under consideration. In Eq. (3) we are using as best-fit distribution the one that maximizes the likelihood (this is why we could add the second equality and express t as the minimum of the likelihood ratio). However Eq. (18) still defines a viable test statistic even if we employ a different method to estimate $n(x|\hat{\mathbf{w}})$.

Neural network estimators of $n(x|\hat{\mathbf{w}})$, or equivalently of $f(x;\hat{\mathbf{w}})$, can be obtained using different loss functions, the one in Eq. (11) being only one of many possibilities. The loss function that is most widely employed in classification problems is the so-called "cross-entropy"

$$\begin{aligned} L[f] &= \sum_{(x,y)}\left[y\log[1+e^{-f(x)}] + (1-y)\frac{N(\mathrm{R})}{\mathcal{N}_{\mathcal{R}}}\log[1+e^{f(x)}]\right] \\ &= \sum_{x\in\mathcal{D}}\log[1+e^{-f(x)}] + \frac{N(\mathrm{R})}{\mathcal{N}_{\mathcal{R}}}\sum_{x\in\mathcal{R}}\log[1+e^{f(x)}]. \end{aligned} \tag{19}$$

The reason why this is a viable choice can be easily understood as follows. In the asymptotic limit, i.e., when the data and the reference sets are large, the sums in Eq. (19) approach expectation values over the variable x. The distribution of the reference sample $\mathcal{R}$ is $n(x|\mathrm{R})$ by construction. The data sample $\mathcal{D}$ is instead distributed according to the "true" data distribution $n(x|\mathrm{T})$, which is precisely the one we would like to estimate. Equation (19) thus approaches the functional

$$L[f] \simeq \int dx n(x|\mathrm{T}) \log\left[1 + e^{-f(x)}\right] + \int dx n(x|\mathrm{R}) \log\left[1 + e^{f(x)}\right]. \quad (20)$$

Let us now take the limit in which the neural network is very large, such that $f(x, \mathbf{w})$ effectively spans the whole set of infinitely differentiable functions of x. In this limit the minimum of $L[f]$ is where the functional derivative $\delta L[f]/\delta f$ vanishes. Therefore the neural network trained with the loss function in Eq. (19) is approximately

$$f(x, \hat{\mathbf{w}}) \simeq \log\left[\frac{n(x|\mathrm{T})}{n(x|\mathrm{R})}\right]. \quad (21)$$

Since $f(x, \hat{\mathbf{w}})$ provides an approximation of the true data distribution, it can be meaningfully used to construct the test statistic. Notice that now t, unlike in the maximum likelihood approach (10), cannot be directly obtained from the value of the loss function at the end of training. On the contrary it must be evaluated from the definition in Eq. (18), using the trained neural network $f(x, \hat{\mathbf{w}})$ and evaluating separately the integral of Eq. (4). This is done with the Monte Carlo method

$$N(\mathbf{w}) = \frac{N(\mathrm{R})}{\mathcal{N}_{\mathcal{R}}} \sum_{x \in \mathcal{R}} e^{f(x, \hat{\mathbf{w}})}, \quad (22)$$

using the same reference sample that is employed for training.

Similar considerations hold for other loss functions such as the square loss or, of course, the maximum likelihood loss in Eq. (11). All of them approach, in the asymptotic limit, integral functionals whose minima give Eq. (21). Choosing one or the other is from this viewpoint merely a matter of technical convenience. We explored quite extensively the possibility of using the cross-entropy loss. This was actually our first attempt, which we eventually abandoned in favor of maximum likelihood, that was found to have better performances in all the examples we studied. At the technical level the advantage of maximum likelihood is that the test statistic is directly related with the minimum of the loss function. We have seen that this is not the case for other choices of the loss function; hence there is a much less direct connection between t and the quantity that is minimized by the training algorithm.

Maximum likelihood is normally considered to be the optimal hypothesis test, in accordance with our findings. However it should be kept in mind that for composite alternative hypotheses there is no rigorous notion of optimal test [65].

In spite of the fact that maximum likelihood was eventually found to be more effective, the possibility of employing other loss functions should be kept in mind for further evolutions of our algorithm, or for different applications. For instance, we mentioned that another possible application of our method could be the comparison between two samples obtained with different Monte Carlo generators. Since in this case there is no sharp notion of which one is the "data" and which one is the "reference" sample, one could argue in favor of a more symmetric loss function such as the cross-entropy or square loss. This is left to future work.

VI. CONCLUSIONS AND OUTLOOK

We studied the possibility of using neural networks to identify data departures from the prediction of a given reference model, making effectively no assumption on the alternative model that is responsible for the discrepancy. A concrete implementation of the idea was presented, in the form of an algorithm that straightforwardly follows from the maximum likelihood hypothesis test. The inputs of the algorithm are the data collected by an experiment and a reference sample that follows the reference model distribution. The reference data set can be obtained from a Monte Carlo event generator or from data in a control region. Its double role is to replace the analytical knowledge of the reference model distribution, which is typically not available, and to turn likelihood maximization into a supervised training process. The output of the algorithm is the ratio between the best-fit data distribution and the reference one and a test statistic variable t. The former can be used to select data that display the highest level of discrepancy with the reference model. The latter measures the disagreement between the reference model and the data and it can be used for a hypothesis test.

We performed simple numerical experiments to assess the virtues of our construction and its limitations. We confirmed the model-independent nature of our method, by showing that it has good sensitivity to different hypothetical new physics signals. We also verified that our method does not suffer from the presence of data that agree well with the reference model prediction, even if those constitute the vast majority of the sample. For the applications that we have in mind, as explained in the introduction and in Sec. II B, this is an essential property. Finally we found that the sensitivity does not depend much on the capacity of the neural network. The results above are obtained in a few simple, one-dimensional, examples. A more extensive investigation would be useful to put them on firmer ground.

We also quantified the sensitivity degradation due to including in the network input an additional variable that does not possess discriminating power between the reference and the new physics models. Some amount of degradation is unavoidable; however the one we observed does not reflect the full potential of our approach. On the other hand the sensitivity scales well with the statistics, by doubling the number of events we recover a sensitivity that is comparable to the one-dimensional case. Even at a fixed number of events we are confident that the situation can be

improved by refining our approach. This belief is motivated by the fact that the sensitivity loss in two dimensions comes from a significant departure, towards larger values, of the reference model t distribution with respect to the χ^2 prediction. We do not have a complete understanding of this phenomenon, but we conjecture that it is due to overfitting and to a nonoptimal choice of the neural network architecture. Overfitting could be the explanation because it produces bumps and other sharp features that contribute significantly to t, which are due to a few events that happen to be concentrated in some region of the phase space. Since they result from few events, these contributions to t can violate the asymptotic formula. The behavior is observed in two dimensions and not in one because two-dimensional data are much more sparse and, hence, easier to overfit. If rather than a fully connected (2,3,1) network we had employed an architecture where the variable c has less links than the variable M, the performances on the example discussed in Sec. IV C would have clearly been better. One might consider the limiting situation where all weights that connect c to the network are set to zero, effectively going back to the one-dimensional (1,3,1) network for which good performances were observed in Sec. IV D. At present it is unclear that this observation could be turned into a systematic optimization strategy. However we notice possible connections with the problem of identifying and eliminating the redundant parameters of a neural network, which goes under the name of "compression" in the machine learning literature [74].

Another direction of investigation is related with the alternative viewpoint on our approach that we discussed in Sec. V. What we are doing is learning from the data a likelihood ratio. We then use it to construct the test statistic. Whether or not the likelihood ratio is learned using the maximum likelihood loss function is irrelevant from this viewpoint. This suggests that we should look for synergies with recent works [43–45,75] where the problem of approximating likelihood ratios with neural networks has been studied. These studies could also help to model the systematic uncertainties of the reference Monte Carlo, through the formalism of nuisance parameters. We argued in Sec. III that the problem of systematics is orthogonal to the one that we are addressing and that it could be solved with standard tools. However studying its interplay with what we are doing would clearly be an important step.

At the purely computational level, the limiting factor of our algorithm is the training time. This can be considerable because we have employed a large number of reference data for training, typically 100 times the actual data. However one could try to employ the reference sample more efficiently. When we write $N(\mathbf{w})$ as in Eq. (4) we are effectively using the most naive Monte Carlo integration strategy; more refined techniques might give the same accuracy with much smaller reference samples. For instance one might employ weighted events, obtained by binning the large original reference sample. If the binning is compatible with the resolution on x, and in turn with the weight clipping of the neural network, Eq. (9) could be evaluated accurately using hundreds of reference events rather than hundreds of thousands. Clearly the loss function in Eq. (11) should be updated accordingly.

In this paper we exclusively discussed our method as a possible approach to model-independent new physics searches. However other applications could be envisaged. The first one is constructing an automated tool that compares the predictions of different Monte Carlo generators, using one of the two generators as "data" and the other as "reference." This might allow one to identify subtle discrepancies that might instead escape ordinary comparisons based on the inspection of selected variables. Monte Carlo generator comparison is much easier to implement than model-independent new physics searches because the data sample size is easier to increase. One might also consider our approach for data validation algorithms. The goal there is to establish if raw data produced during a certain, relatively short, period of time were collected under appropriate conditions, or if instead a contingent problem occurred in the data acquisition system. One should thus compare them with previously collected data, which might be used as the reference sample. This should be relatively easy to achieve because the data are abundant and because the reference sample is perfect by definition. Hence one would not need to worry about systematic uncertainties in the reference. We believe that these directions deserve further study.

ACKNOWLEDGMENTS

We would like to thank M. Pierini and M. Zanetti for collaboration during the early stages of this work. We also thank N. Arkani-Hamed, L. Biggio, V. Hirschi, M. Papucci, L. Rosasco and N. Toro for useful discussions. We would also like to thank T. Cohen for very useful comments on the manuscript. R. T. D. is supported by the U.S. Department of Energy under Contract No. DE-AC02-76SF00515.

APPENDIX: A SHORT INTRODUCTION TO NEURAL NETWORKS

As mentioned in Sec. II, a neural network is a set of functions. In our notation each architecture corresponds to a family of real functions $\mathcal{F}_{\vec{a}} = \{f_{\vec{a}}(x;\mathbf{w}),\ \forall\ \mathbf{w}\}$ of the d-dimensional variable x, labeled by a vector $\vec{a}$ of integers that specifies the neural network. The functions depend on N_{par} real parameters $\mathbf{w}$, generically called "weights" in what follows.

This family of functions, i.e., the neural network, is constructed as the composition of elementary blocks, called layers. In our notation, which follows the one of *Mathematica* [76], layers can be either of the elementwise or of the linear type. An elementwise layer applies a scalar

function to all the elements of the input vector, producing an output with the same dimensionality as the input. In our implementation all elementwise layers (i.e., all our activation functions [61,62]) are logistic sigmoids:

$$\sigma(z) = \frac{1}{1+e^{-z}}. \tag{A1}$$

As the name suggests, a linear layer performs a linear transformation and the dimensionality of its output (d_{O}) can be different from the one of the input (d_{I}). It can be represented as

$$[\lambda_{d_{\mathrm{O}},d_{\mathrm{I}}}(\vec{z})]_{\alpha_{\mathrm{O}}} = \sum_{\alpha_I=1}^{d_I} w_{\alpha_{\mathrm{O}}}^{\alpha_{\mathrm{I}}} z_{\alpha_{\mathrm{I}}} + \bar{w}_{\alpha_{\mathrm{O}}}, \tag{A2}$$

where α_{O} runs from 1 to d_{O}. The free parameters of a linear layer are the d_{O} times d_{I} entries of the $w_{\alpha_{\mathrm{O}}}^{\alpha_{\mathrm{I}}}$ matrix, plus the d_{O} shifts $\bar{w}_{\alpha_{\mathrm{O}}}$, for a total of $d_{\mathrm{O}}(d_{\mathrm{I}}+1)$ parameters. We denote all of them as weights in spite of the fact that the $\bar{w}$'s are often called "biases" in the machine learning literature.

A neural network is the composition of layers, alternating linear and elementwise ones:

$$f_{\vec{a}}(\cdot;\mathbf{w}) = \lambda_{a_{\mathrm{L}}=1,a_{\mathrm{L}-1}} \circ \sigma \circ \cdots \circ \sigma \circ \lambda_{a_2,a_1} \circ \sigma \circ \lambda_{a_1,a_0=d}. \tag{A3}$$

If the network is fully connected, i.e., the dimension of the output of layer $n-1$ equals that of the input of layer n, for every layer, then the total number of free parameters that the network depends on is

$$N_{\mathrm{par}}(\vec{a}) = \sum_{n=1}^{\mathrm{L}} a_n(a_{n-1}+1). \tag{A4}$$

The neural network function is applied to the variable x; hence the input of the first linear layer has dimensionality $a_0 = d$. The neural network output that we are interested in must be a real number; hence $a_{\mathrm{L}} = 1$. We are instead free to choose the remaining $\mathrm{L}-1$ entries of the $(\mathrm{L}+1)$-dimensional vector $\vec{a}$. Notice that $\mathbf{L}$ only counts the number of linear layers in the network. However we often refer to it as the number of layers, matching in this way the more standard terminology in which one layer is the composition of a linear transformation with σ. For instance, a two-layer neural network acting on a one-dimensional input variable x is represented by the vector $\vec{a} = (1, N_{\mathrm{neu}}, 1)$, where N_{neu} is the number of neurons.

In Eq. (A2) each neuron corresponds to a different value of α_O. So $\vec{a} = (1, N_{\mathrm{neu}}, 1)$ depends on $3N_{\mathrm{neu}}+1$ free parameters and its explicit functional form is

$$f_{(1,N_{\mathrm{neu}},1)}(x;\mathbf{w}) = \sum_{\alpha=1}^{N_{\mathrm{neu}}} (w_{(2)})^{\alpha} \sigma[(w_{(1)})_{\alpha} x + (\bar{w}_{(1)})_{\alpha}] + \bar{w}_{(2)}. \tag{A5}$$

For the applications considered in this paper we have employed simple networks of this class. However we have tested also deeper networks ($\mathrm{L} > 2$) for $d > 1$ finding comparable performances.

Once we have built the network, we need to train it. This is not different from fitting free parameters $\mathbf{w}$ given experimental observations. In analogy with maximum likelihood parameter estimation, we write down a loss function that at the minimum gives estimators of the values of $\mathbf{w}$ that best describe the data. Then we need to find the minimum.

The choice of loss function is determined by the specific problem at hand. In Sec. III we have already discussed what we consider the most motivated construction for our model-independent searches and in Sec. V we showed a variation based on more standard classification problems. Here we illustrate the point with a simpler example for the readers that are not familiar with the subject. For concreteness we discuss what one would do for supervised learning and refer the reader interested in semisupervised, unsupervised and reinforcement learning to [61,62,77,78].

Imagine that you have two sets of pictures one of cats and one of dogs. You would like the network to output 1 if given a cat and 0 for a dog. In this case the input x can be an array of numbers, each representing a different pixel of the picture. Then an obvious choice for the loss function would be

$$L[f] = \sum_{x\in\mathrm{cats}} [1 - f_{\vec{a}}(x|\mathbf{w})]^2 + \sum_{x\in\mathrm{dogs}} [f_{\vec{a}}(x|\mathbf{w})]^2. \tag{A6}$$

At the minimum of L, $f_{\vec{a}}(x_{\mathrm{cat}}|\hat{\mathbf{w}}) = 1$ and $f_{\vec{a}}(x_{\mathrm{dog}}|\hat{\mathbf{w}}) = 0$. It is very easy to prove it, by taking a functional derivative of L with respect to f. What is actually implemented in a computer consists in taking the derivatives of L with respect to the weights going backwards from the last layer.

Note that the form of the loss function in (A6) is just illustrative. As we have also mentioned in the main body of the text, in practical applications the cross-entropy, the Kullback-Leibler divergence and their variations are more widely used. One quality that they have over the χ^2 used in (11) is that their logarithms cancel the exponential saturation of sigmoids and hyperbolic tangents at least for the last layer, making the derivatives larger and the minimization process faster for certain values of the input.

Since the loss functions obtained by nesting layers are in general nonconvex, there are no algorithms that are guaranteed to find a global minimum. The prevailing approach consists in finding a "good enough" local minimum by using stochastic gradient descent. Gradient descent simply consists in taking a derivative of the loss function and updating the weights by moving them a small amount ϵ in the direction in which the derivative decreases. This

technique was proposed by Cauchy in 1847 [79]. The parameter ϵ is called the learning rate. It can be fixed *a priori* or changed adaptively during training. Since computing the derivative over the entire training sample is usually computationally unfeasible, it is typically computed on a subsample chosen at random. This is what goes under the name of stochastic gradient descent [62,77]. The RMSprop algorithm [71] that we employ is based on stochastic gradient descent.

The process of evaluating L on a subset of the cats and dogs sample, taking its derivatives and updating the values of the weights is known as training and the sample used for the process is known as the training sample. This comes in as many repetitions as it takes to obtain an acceptable degree of accuracy. The accuracy of classifiers, as the one in this simple example, can be tested on a separate sample, (you guessed it) the testing sample. In the applications discussed in the paper, where we are not solving a classification problem, we can perform a different test, by comparing the neural network estimation of the data distribution with its true functional form.

It can be proven that a function built following the procedure outlined at the beginning of this section can approximate with arbitrary accuracy any continuous function in a compact domain of $\mathbb{R}^N$. For a more precise statement of the relevant theorems we refer to [63,80–82]. Here we would like to present a heuristic argument that will also make clear why neural networks provide a good parametrization for the problem described in this work.

Take two neurons with a logistic sigmoid activation function and send their output to a third one. For simplicity consider a one-dimensional input for the first layer. The function that describes this small neural network is

$$f_{(1,2,1)}(x) = w_1'\sigma(z_1(x)) + w_2'\sigma(z_2(x)) + b',$$
$$z_i(x) = w_i x + b_i, \tag{A7}$$

where $i = 1, 2$ labels the two initial neurons. For $w_1' = -w_2' = w'$ and $b' = 0$ we have

$$f_{(1,2,1)}(x) = w'[\sigma(w_1 x + b_1) - \sigma(w_2 x + b_2)]. \tag{A8}$$

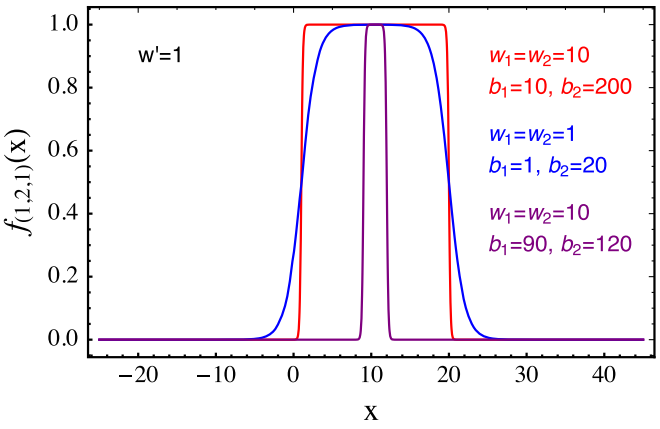

FIG. 11. Illustration of how three neurons with logistic sigmoid activation functions can reproduce a rectangular function or a smooth peak. The parameters in the legend of the plot are defined in Eqs. (A7) and (A8).

This is plotted as a function of x in Fig. 11. It is approximately zero for $x \gtrsim -b_2/w_2$ and $x \lesssim -b_1/w_1$ and roughly constant and equal to w' otherwise.

As illustrated in Fig. 11, by increasing w_1 and w_2 we can make the transition between zero and w' arbitrarily sharp. By adjusting b_1 and b_2 we can make the domain over which $f_{(1,2,1)}(x)$ is nonzero as narrow as we want. So we can make this three-neuron unit generate a smooth peak, a broad plateau or a rectangular function. By combining many of these units we can approximate any continuous function as a juxtaposition of rectangular functions. In higher dimensions we can repeat this argument by adding two more neurons for each new direction. We can send all their outputs into a single final neuron and construct a multidimensional rectangular function in the same way.

As discussed in Sec. II this also shows why neural networks are promising candidates for new physics searches. Even if we do not know *a priori* the type of signal that we are looking for, a network with very few parameters can reproduce an arbitrarily sharp feature, remaining smooth in its absence. Fewer free parameters mean a smaller look-elsewhere effect and a larger sensitivity.

[1] Particle Data Group, Review of particle physics, Chin. Phys. C **40**, 100001 (2016).
[2] G. Choudalakis, On hypothesis testing, trials factor, hypertests and the BumpHunter, in *Proceedings of the PHYSTAT 2011 Workshop on Statistical Issues Related to Discovery Claims in Search Experiments and Unfolding, 2011* (CERN, Geneva, Switzerland, 2011).
[3] B. Abbott *et al.* (D0 Collaboration), Search for new physics in $e\mu X$ data at D0 using Sherlock: A quasi model independent search strategy for new physics, Phys. Rev. D **62**, 092004 (2000).
[4] V. M. Abazov *et al.* (D0 Collaboration), A quasi model independent search for new physics at large transverse momentum, Phys. Rev. D **64**, 012004 (2001).

[5] A. Aktas *et al.* (H1 Collaboration), A general search for new phenomena in ep scattering at HERA, Phys. Lett. B **602**, 14 (2004).
[6] F. D. Aaron *et al.* (H1 Collaboration), A general search for new phenomena at HERA, Phys. Lett. B **674**, 257 (2009).
[7] P. Asadi, M. R. Buckley, A. DiFranzo, A. Monteux, and D. Shih, Digging deeper for new physics in the LHC data, J. High Energy Phys. 11 (2017) 194.
[8] T. Aaltonen *et al.* (CDF Collaboration), Model-independent and quasi-model-independent search for new physics at CDF, Phys. Rev. D **78**, 012002 (2008).
[9] T. Aaltonen *et al.* (CDF Collaboration), Global search for new physics with 2.0 fb^{-1} at CDF, Phys. Rev. D **79**, 011101 (2009).
[10] CMS Collaboration, MUSIC—An automated scan for deviations between data and Monte Carlo simulation, CERN, Report No. CMS-PAS-EXO-08-005.
[11] CMS Collaboration, Model unspecific search for new physics in pp collisions at $\sqrt{(s)} = 7$ TeV, Report No. CMS-PAS-EXO-10-021.
[12] ATLAS Collaboration, A model independent general search for new phenomena with the ATLAS detector at $\sqrt{s} =$ 13 TeV, Report No. ATLAS-CONF-2017-001.
[13] ATLAS Collaboration, A general search for new phenomena with the ATLAS detector in pp collisions at $\sqrt{(s)} =$ 7 TeV, Report No. ATLAS-CONF-2012-107.
[14] ATLAS Collaboration, A general search for new phenomena with the ATLAS detector in pp collisions at $\sqrt{s} = 8$ TeV, Report No. ATLAS-CONF-2014-006.
[15] L. de Oliveira, M. Kagan, L. Mackey, B. Nachman, and A. Schwartzman, Jet-images—Deep learning edition, J. High Energy Phys. 07 (2016) 069.
[16] A. Schwartzman, M. Kagan, L. Mackey, B. Nachman, and L. De Oliveira, Image processing, computer vision, and deep learning: New approaches to the analysis and physics interpretation of LHC events, J. Phys. Conf. Ser. **762**, 012035 (2016).
[17] M. Kagan, L. d. Oliveira, L. Mackey, B. Nachman, and A. Schwartzman, Boosted jet tagging with jet-images and deep neural networks, EPJ Web Conf. **127**, 00009 (2016).
[18] A. J. Larkoski, I. Moult, and B. Nachman, Jet substructure at the large hadron collider: A review of recent advances in theory and machine learning, arXiv:1709.04464.
[19] G. Louppe, K. Cho, C. Becot, and K. Cranmer, QCD-aware recursive neural networks for jet physics, arXiv:1702.00748.
[20] C. Shimmin, P. Sadowski, P. Baldi, E. Weik, D. Whiteson, E. Goul, and A. SÃÿgaard, Decorrelated jet substructure tagging using adversarial neural networks, Phys. Rev. D **96**, 074034 (2017).
[21] P. Baldi, K. Bauer, C. Eng, P. Sadowski, and D. Whiteson, Jet substructure classification in high-energy physics with deep neural networks, Phys. Rev. D **93**, 094034 (2016).
[22] D. Guest, J. Collado, P. Baldi, S.-C. Hsu, G. Urban, and D. Whiteson, Jet flavor classification in high-energy physics with deep neural networks, Phys. Rev. D **94**, 112002 (2016).
[23] L. G. Almeida, M. BackoviÄĞ, M. Cliche, S. J. Lee, and M. Perelstein, Playing tag with ANN: Boosted top identification with pattern recognition, J. High Energy Phys. 07 (2015) 086.
[24] J. Barnard, E. N. Dawe, M. J. Dolan, and N. Rajcic, Parton shower uncertainties in jet substructure analyses with deep neural networks, Phys. Rev. D **95**, 014018 (2017).
[25] G. Kasieczka, T. Plehn, M. Russell, and T. Schell, Deep-learning top taggers or the end of QCD?, J. High Energy Phys. 05 (2017) 006.
[26] A. Butter, G. Kasieczka, T. Plehn, and M. Russell, Deep-learned top tagging with a Lorentz layer, SciPost Phys. **5**, 028 (2018).
[27] K. Datta and A. Larkoski, How much information is in a jet?, J. High Energy Phys. 06 (2017) 073.
[28] K. Datta and A. J. Larkoski, Novel jet observables from machine learning, J. High Energy Phys. 03 (2018) 086.
[29] K. Fraser and M. D. Schwartz, Jet charge and machine learning, J. High Energy Phys. 10 (2018) 093.
[30] A. Andreassen, I. Feige, C. Frye, and M. D. Schwartz, JUNIPR: A framework for unsupervised machine learning in particle physics, arXiv:1804.09720.
[31] S. Macaluso and D. Shih, Pulling out all the tops with computer vision and deep learning, J. High Energy Phys. 10 (2018) 121.
[32] ATLAS Collaboration, Performance of top quark and W boson tagging in run 2 with ATLAS, Report No. ATLAS-CONF-2017-064.
[33] CMS Collaboration, CMS phase 1 heavy flavour identification performance and developments, Report No. CMS-DP-2017-013, https://cds.cern.ch/record/2263802.
[34] ATLAS Collaboration, Optimisation and performance studies of the ATLAS *b*-tagging algorithms for the 2017-18 LHC run, Technical Report No. ATL-PHYS-PUB-2017-013, CERN, Geneva, 2017.
[35] ATLAS Collaboration, Identification of hadronically-decaying W bosons and top quarks using high-level features as input to boosted decision trees and deep neural networks in ATLAS at $\sqrt{s} = 13$ TeV, Technical Report No. ATL-PHYS-PUB-2017-004, CERN, Geneva, 2017.
[36] CMS Collaboration, Heavy flavor identification at CMS with deep neural networks, https://cds.cern.ch/record/2255736.
[37] ATLAS Collaboration, Identification of jets containing *b*-hadrons with recurrent neural networks at the ATLAS experiment, Technical Report No. ATL-PHYS-PUB-2017-003, CERN, Geneva, 2017.
[38] ATLAS Collaboration, Quark versus gluon jet tagging using jet images with the ATLAS detector, Technical Report No. ATL-PHYS-PUB-2017-017, CERN, Geneva, 2017.
[39] CMS Collaboration, New developments for jet substructure reconstruction in CMS, https://cds.cern.ch/record/2275226.
[40] P. Baldi, K. Cranmer, T. Faucett, P. Sadowski, and D. Whiteson, Parameterized neural networks for high-energy physics, Eur. Phys. J. C **76**, 235 (2016).
[41] S. Chang, T. Cohen, and B. Ostdiek, What is the machine learning?, Phys. Rev. D **97**, 056009 (2018).
[42] T. Cohen, M. Freytsis, and B. Ostdiek, (Machine) learning to do more with less, J. High Energy Phys. 02 (2018) 034.
[43] J. Brehmer, K. Cranmer, G. Louppe, and J. Pavez, A guide to constraining effective field theories with machine learning, Phys. Rev. D **98**, 052004 (2018).

[44] J. Brehmer, K. Cranmer, G. Louppe, and J. Pavez, Constraining Effective Field Theories with Machine Learning, Phys. Rev. Lett. **121**, 111801 (2018).
[45] J. Brehmer, G. Louppe, J. Pavez, and K. Cranmer, Mining gold from implicit models to improve likelihood-free inference, arXiv:1805.12244.
[46] T. Roxlo and M. Reece, Opening the black box of neural nets: Case studies in stop/top discrimination, arXiv:1804.09278.
[47] J. H. Collins, K. Howe, and B. Nachman, CWoLa hunting: Extending the bump hunt with machine learning, arXiv:1805.02664.
[48] M. Paganini, L. de Oliveira, and B. Nachman, CaloGAN: Simulating 3D high energy particle showers in multilayer electromagnetic calorimeters with generative adversarial networks, Phys. Rev. D **97**, 014021 (2018).
[49] L. de Oliveira, M. Paganini, and B. Nachman, Controlling physical attributes in GAN-accelerated simulation of electromagnetic calorimeters, J. Phys. Conf. Ser. **1085**, 042017 (2018).
[50] M. Paganini, L. de Oliveira, and B. Nachman, Accelerating Science with Generative Adversarial Networks: An Application to 3D Particle Showers in Multilayer Calorimeters, Phys. Rev. Lett. **120**, 042003 (2018).
[51] R. D. Ball *et al.* (NNPDF Collaboration), Parton distributions for the LHC run II, J. High Energy Phys. 04 (2015) 040.
[52] S. Forte, L. Garrido, J. I. Latorre, and A. Piccione, Neural network parametrization of deep inelastic structure functions, J. High Energy Phys. 05 (2002) 062.
[53] G. Cybenko, Approximation by superpositions of a sigmoidal function, Math. Control Signals Syst. **2**, 303 (1989).
[54] V. Y. Kreinovich, Arbitrary nonlinearity is sufficient to represent all functions by neural networks: A theorem, Neural Netw. **4**, 381 (1991).
[55] R. Hecht-Nielsen, Neural networks for perception, Vol. **2**, http://dl.acm.org/citation.cfm?id=140639.140643.
[56] K. Hornik, M. Stinchcombe, and H. White, Multilayer feedforward networks are universal approximators, Neural Netw. **2**, 359 (1989).
[57] S. Liang and R. Srikant, Why deep neural networks for function approximation?, arXiv:1610.04161.
[58] T. A. Poggio, H. Mhaskar, L. Rosasco, B. Miranda, and Q. Liao, Why and when can deep—but not shallow—networks avoid the curse of dimensionality: A review, arXiv:1611.00740.
[59] F. R. Bach, Breaking the curse of dimensionality with convex neural networks, arXiv:1412.8690.
[60] H. Montanelli and Q. Du, Deep ReLU networks lessen the curse of dimensionality, arXiv:1712.08688.
[61] C. M. Bishop, *Pattern Recognition and Machine Learning* (Springer, New York, 2006).
[62] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (MIT, Cambridge, MA, 2016).
[63] S. Haykin, *Neural Networks: A Comprehensive Foundation* (Prentice-Hall, Englewood Cliffs, NJ, 1998), Vol. 2.
[64] M. Kuusela, T. Vatanen, E. Malmi, T. Raiko, T. Aaltonen, and Y. Nagai, Semi-supervised anomaly detection—Towards model-independent searches of new physics, J. Phys. Conf. Ser. **368**, 012032 (2012).
[65] J. Neyman and E. S. Pearson, On the problem of the most efficient tests of statistical hypotheses, Phil. Trans. R. Soc. A **231**, 289 (1933).
[66] G. Cowan, *Statistical Data Analysis* (Clarendon, Oxford, 1998).
[67] G. Cowan, Lecture at the 2017 GGI school on the theory of fundamental interactions, https://www.youtube.com/watch?v=Y23Kxg61scc&list=PLDxsZU4NC6Z5DFFFx2bj03phoZpP1qrp2&index=2.
[68] http://neuralnetworksanddeeplearning.com/chap4.html.
[69] W. A. Rolke, A. M. Lopez, and J. Conrad, Limits and confidence intervals in the presence of nuisance parameters, Nucl. Instrum. Methods Phys. Res., Sect. A **551**, 493 (2005).
[70] G. Cowan, K. Cranmer, E. Gross, and O. Vitells, Asymptotic formulae for likelihood-based tests of new physics, Eur. Phys. J. C **71**, 1 (2011); Erratum, Eur. Phys. J. C **73**, 2501(E) (2013).
[71] http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf.
[72] S. S. Wilks, The large-sample distribution of the likelihood ratio for testing composite hypotheses, Ann. Math. Stat. **9**, 60 (1938).
[73] A. Wald, Tests of statistical hypotheses concerning several parameters when the number of observations is large, Trans. Am. Math. Soc. **54**, 426 (1943).
[74] S. Han, H. Mao, and W. J. Dally, Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding, arXiv:1510.00149.
[75] K. Cranmer, J. Pavez, and G. Louppe, Approximating likelihood ratios with calibrated discriminative classifiers, arXiv:1506.02169.
[76] https://www.wolfram.com/mathematica/.
[77] M. Z. Alom, T. M. Taha, C. Yakopcic, S. Westberg, M. Hasan, B. C. Van Esesn, A. A. S. Awwal, and V. K. Asari, The history began from AlexNet: A comprehensive survey on deep learning approaches, arXiv:1803.01164.
[78] T. Hastie, R. Tibshirani, and J. Friedman, The elements of statistical learning, arXiv:1011.1669v3.
[79] M. Augustine Cauchy, Méthode générale pour la résolution des systèmes d'équations simultanées, Comptes Rendus Hebd. Séances Acad. Sci. **25**, 536 (1847).
[80] G. Cybenko, Approximation by superpositions of a sigmoidal function, Math. Control Signals Syst. **2**, 303 (1989).
[81] K. Hornik, Approximation capabilities of multilayer feedforward networks, Neural Netw. **4**, 251 (1991).
[82] V. Y. Kreinovich, Arbitrary nonlinearity is sufficient to represent all functions by neural networks: A theorem, Neural Netw. **4**, 381 (1991).