Adversarial Training against Systematic Uncertainty

Gilles Louppe,[1] **Michael Kagan**,[2] and **Kyle Cranmer**[1]
[1] *New York University*
[2] *SLAC National Accelerator Laboratory*

Nice and cozy abstract goes here...

I. INTRODUCTION

The discovery of new particles or physical phenomena in high energy physics experiments, like those at the Large Hadron Collider, requires the observation of statistically significant deviations from the predictions of the Standard Model, the current model of the known fundamental particles and their interactions. Typically, discovery requires satisfying the *5-sigma rule*, whereby the deviation must be at least five standard deviations from predictions, i.e. having a p-value of $p \lesssim 3 \times 10^{-7}$. The evaluation of the statistical significance must not only account for statistical uncertainties due to random fluctuations, but also for systematic uncertainties that represent our incomplete knowledge of the theory of particle interactions or of the properties of the experimental detection apparatus. Systematic uncertainties can alter the expected rate of signal and background classes, e.g. their priors, as well as the distributions of features. As such, the signal class true-positive rate and the background class false-positive rate of a classifier can change with variations in the systematic uncertainty thereby decreasing the sensitivity of an analysis. In this paper, we present a new strategy for training a classifier, based on adversarial training [10], which aims at building a classifier whose decision function is invariant under variations due to systematic uncertainties. In this way, uncertainties on predictions from simulations will have a reduced impact when computing the significance of potential deviations after application of the discriminant model and thereby increase the sensitivity of the analysis to signals from potential new physics sources. [GL: Shall we be more explicit somewhere and put a formula underlining the effect of having a pivotal classifier on the resulting sensitivity? (versus a classifier whose decision function depends on the nuisance.)]

Given data X and associated labels Y taking values $y \in \mathcal{Y} = \{s, b_{i=1\dots k}\}$ where s is the signal class and $b_{i=1\dots k}$ are the background classes, the probability density of the data can be written as a mixture

$$p(X) = \pi_s p(X|Y=s) + \sum_{i=1}^{k} \pi_{b_i} p(X|Y=b_i), \quad (1)$$

where π_y represents the prior for the signal or a given background, and $p(X|Y)$ is the conditional distribution of the data for the signal or a given background. Systematic uncertainties can affect this distribution in the following ways [GL: add references]:

- Uncertainties on the priors π_y. These arise from uncertainties on the rates we expect a given process to occur as predicted from theoretical calculations or from alternative measurements of the rate of a given class.

- Uncertainties in the simulators of the physical interactions, giving rise to uncertainties on the distributions $p(X|Y)$. These uncertainties are often determined by comparing the predictions of different simulations using differing models of the same physical process.

- Uncertainties on the detection apparatus and its effect on the measurement of particle properties, giving rise to uncertainties on the distributions $p(X|Y)$. For instance, this includes uncertainties on our knowledge of the mean and variance of the measured energy distribution of a particle.

In this paper, we will not address uncertainties of the first type mentioned above (uncertainties on the priors). Rather, we will focus on mitigating the effect of systematic uncertainties that alter the conditional distributions $p(X|Y)$.

Systematic uncertainties are parameterized by nuisance parameters in the statistical analysis of data [GL: add references]. Nuisance parameter often, though not always, have a suitable prior coming from alternative measurements in data. For instance, the average measured energy of an electron in simulation may be compared with those in data using a relatively pure sample of Z boson decays to electrons. The average measured electron energy can not be measured without uncertainty, and the measured uncertainty can be used in other analyses to estimate the uncertainty deriving from our imperfect knowledge of the electron's energy. A nuisance parameter is assigned to the electron energy uncertainty, which controls the variation of the electron energy in the statistical analysis of data. Nuisance parameters are typically, though not always, constrained by a suitable prior, e.g. a Gaussian distribution with a standard deviation equal to the measured uncertainty. [GL: ... and accordingly, we want to build a classifier independent on the nuisance parameter, as a way to be independent of the corresponding systematics.]

[GL: Add paper outline.]

II. PROBLEM STATEMENT

Let assume a probability space $(\Omega, \mathcal{F}, P)$, where Ω is a sample space, $\mathcal{F}$ is a set of events and P is a probability measure. Let consider the multivariate random variables $X_z : \Omega \mapsto \mathbb{R}^p$ and $Y : \Omega \mapsto \mathcal{Y}$, where X_z denotes a dependence of the functional X on a nuisance parameter Z whose values $z \in \mathcal{Z}$ define a parameterized family of its systematic uncertainties. That is, X_z and Y induce together a joint probability distribution $p(X, Y|z)$, where the conditional on z denotes X_z. For training, let further assume a finite set $\{x_i, y_i, z_i\}_{i=1}^N$ of realizations $X_{z_i}(\omega_i), Y(\omega_i)$, for $\omega_i \in \Omega$ and known values z_i of the nuisance parameter. Our goal is to learn a function $f(\cdot; \theta_f) : \mathbb{R}^p \mapsto \mathcal{Y}$ of parameters θ_f (e.g., a neural network-based classifier if $\mathcal{Y}$ is a finite set of classes) and minimizing a loss $\mathcal{L}_f(\theta_f)$ (e.g., the cross-entropy). In addition, we require that $f(X_z; \theta_f)$ should be robust to the value z of the nuisance parameter – which remains unknown at test time. More specifically, we aim at building f such that in the ideal case

$$f(X_z(\omega); \theta_f) = f(X_{z'}(\omega); \theta_f) \tag{2}$$

for all samples $\omega \in \Omega$ and all z, z' pairs of values of the nuisance parameter.

Since we do not have training tuples $(X_z(\omega), X_{z'}(\omega))$ (for the same unknown ω), we propose instead to solve the closely related problem of finding a predictive function f such that

$$\begin{aligned} &P(\{\omega | f(X_z(\omega); \theta_f) = y\}) \\ &\quad = P(\{\omega' | f(X_{z'}(\omega'); \theta_f) = y\}) \end{aligned} \tag{3}$$

for all $y \in \mathcal{Y}$. In words, we are looking for a predictive function f which is a pivotal quantity [6] with respect to the nuisance parameter. That is, such that the distribution of $f(X_z; \theta_f)$ is invariant with respect to the value z of the nuisance. Note that a function f for which Eqn. 2 is true necessarily satisfies Eqn. 3. In general, the converse is however not true, since the sets of samples $\{\omega | f(X_z(\omega); \theta_f) = y\}$ and $\{\omega' | f(X_{z'}(\omega'); \theta_f) = y\}$ do not need to be the same for the equality to hold. In order to simplify notations, and as only Eqn. 3 is of direct interest in this work, we denote from here on the pivotal quantity criterion as

$$p(f(X; \theta_f)|z) = p(f(X; \theta_f)|z') \tag{4}$$

for all $z, z' \in \mathcal{Z}$ and all values of $f(X; \theta_f)$.

III. METHOD

Adversarial training was first proposed by [10] as a way to build a generative model capable of producing samples from random noise $z \sim p_Z$. More specifically, the authors pit a generative model $g : \mathbb{R} \mapsto \mathbb{R}^p$ against an adversary classifier $d : \mathbb{R}^p \mapsto \{0, 1\}$ whose antagonistic objective is to recognize real data X from generated data $g(Z)$. Both models g and d are trained simultaneously, in such a way that g learns to produce samples that are difficult to identify by d, while d incrementally adapts to changes in g. At the equilibrium, g models a distribution whose samples can be identified by d only by chance. That is, assuming enough capacity in d and g, the distribution $p_{g(Z)}$ eventually converges towards the real distribution p_X.

In this work, we repurpose adversarial training as a means to constraint the predictive model f in order to satisfy Eqn. 4. As illustrated in Figure 1, we pit f against an adversary model $r := p_{\theta_r}(z|f(X; \theta_f))$ of parameters θ_r and associated loss $\mathcal{L}_r(\theta_f, \theta_r)$. This model takes as input realizations of $f(X; \theta_f)$, for the current value θ_f of f parameters, and produces as output a function $p_{\theta_r}(z|f(X; \theta_f))$ modeling the posterior probability density that z parameterizes the sample X observed through $f(\cdot; \theta_f)$. Intuitively, if $p(f(X; \theta_f)|z)$ varies with z, then the corresponding correlation can be captured by r. By contrast, if $p(f(X; \theta_f)|z)$ is invariant with z, as we require, then r should perform poorly and be close to random guessing. Training f such that it additionally minimizes the performance of r therefore acts as a regularization towards Eqn. 4.

If Z takes discrete values, then p_{θ_r} can be represented e.g. as a probabilistic classifier $\mathbb{R} \mapsto \mathbb{R}^{|\mathcal{Z}|}$ whose output j (for $j = 1, \ldots, |\mathcal{Z}|$) is the estimated probability mass $p_{\theta_r}(z_j|f(X; \theta_f))$. Similarly, if Z takes continuous values and if we assume some parameteric distribution $\mathcal{P}$ for $Z|f(X; \theta_f)$ (e.g., a mixture of gaussians over a bounded support, as modeled with a mixture density network [3]), then p_{θ_r} can be represented e.g. as network whose output j is the estimated value of the corresponding parameter θ^j of that distribution (e.g., the mean, variance and mixing coefficients of its components). As in [3, 13], the estimated probability density $p_{\theta_r}(z|f(X; \theta_f))$ can then be evaluated for any $z \in \mathcal{Z}$. As further explained in the next section, let us note that the adversary r may take any form, i.e. it does need to be a neural network, as long as it exposes a differentiable function $p_{\theta_r}(z|f(X; \theta_f))$ of sufficient capacity to represent the true distribution within its bounded support.

As for generative adversarial networks, we propose to train f and r simultaneously, which we carry out by considering the value function

$$E(\theta_f, \theta_r) = \mathcal{L}_f(\theta_f) - \mathcal{L}_r(\theta_f, \theta_r) \tag{5}$$

that we optimize by finding the saddle point $(\hat{\theta}_f, \hat{\theta}_r)$ such that

$$\hat{\theta}_f = \arg\min_{\theta_f} E(\theta_f, \hat{\theta}_r), \tag{6}$$

$$\hat{\theta}_r = \arg\max_{\theta_r} E(\hat{\theta}_f, \theta_r). \tag{7}$$

Without loss of generality, the adversarial training procedure to obtain $(\hat{\theta}_f, \hat{\theta}_r)$ is formally presented in Algorithm 1 in the case of a binary classifier $f : \mathbb{R}^p \mapsto [0, 1]$

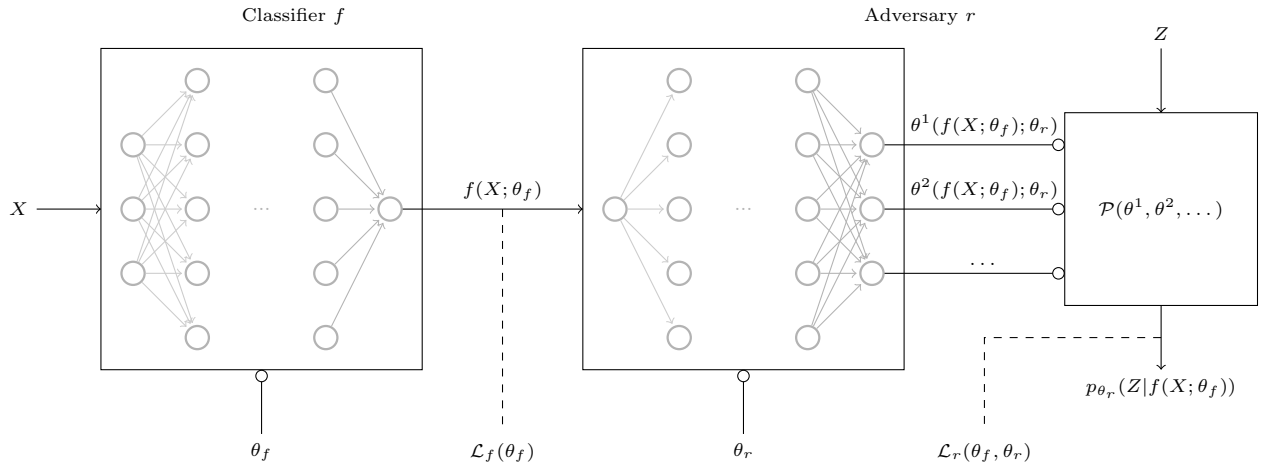

FIG. 1. Architecture for the adversarial training of a binary classifier f against a nuisance parameter Z. The adversary r models the distribution $p(Z|f(X;\theta_f))$ of the nuisance as observed only through the output $f(X;\theta_f)$ of the classifier. By maximizing the antagonistic objective $\mathcal{L}_r(\theta_f,\theta_r)$ (as part of minimizing $\mathcal{L}_f(\theta_f)-\lambda\mathcal{L}_r(\theta_f,\theta_r)$), the classifier f forces $p(Z|f(X;\theta_f)$ to become uniform, which happens when $f(X;\theta_f)$ is independent of the nuisance parameter Z and therefore pivotal.

modeling $p(Y=1|X)$. For reasons further explained in Section IV, $\mathcal{L}_f$ and $\mathcal{L}_r$ are respectively set to the expected value of the negative log-likelihood of $Y|X$ under f and of $Z|f(X;\theta_f)$ under r:

$$\mathcal{L}_f(\theta_f) = \mathbb{E}_X\mathbb{E}_{Y|X}[-\log p_{\theta_f}(Y|X)], \tag{8}$$
$$\mathcal{L}_r(\theta_f,\theta_r) = \mathbb{E}_X\mathbb{E}_{Z|f(X;\theta_f)}[-\log p_{\theta_r}(Z|f(X;\theta_f))]. \tag{9}$$

The optimization algorithm consists in using stochastic gradient descent alternatively for solving Eqn. 6 and 7.

IV. THEORETICAL RESULTS

In this section, we show that in the setting of Algorithm 1 where $\mathcal{L}_f$ and $\mathcal{L}_r$ are respectively set to expected value of the negative log-likelihood of $Y|X$ under f and of $Z|f(X;\theta_f)$ under r, the procedure converges to a classifier f which is a pivotal quantity in the sense of Eqn. 4.

In this setting, the nuisance parameter Z is considered as a random variable of bounded support, for which we require the uniform prior $p(z)$ (for $z\in\mathcal{Z}$). Importantly, classification of Y with respect to X is therefore considered in the context where Z is marginalized out, which means that the classifier minimizing $\mathcal{L}_f$ is optimal with respect to $Y|X$, but not necessarily with $Y|X,Z$. Results hold for a nuisance parameter Z taking either categorical values or continuous values within a bounded support. By abuse of notation, $H(p_Z)$ denotes the differential entropy in this latter case. Finally, propositions below are derived in a non-parametric setting, by assuming that both f and r have enough capacity.

Proposition 1. *Let $\hat{\theta}_r = \arg\max_{\theta_r} E(\theta_f,\theta_r)$ for fixed θ_f. If $p_{\hat{\theta}_r}(z|f(X;\theta_f)) = p(z)$ for all $z\in\mathcal{Z}$ and all values of $f(X;\theta_f)$, then f is a pivotal quantity.*

Proof. Since we assume enough capacity for r, the optimal parameters

$$\hat{\theta}_r = \arg\max_{\theta_r} E(\theta_f,\theta_r) = \arg\min_{\theta_r} \mathcal{L}_r(\theta_f,\theta_r)$$

are such that $p_{\hat{\theta}_r}(z|f(X;\theta_f)) = p(z|f(X;\theta_f))$. By assumption, $p_{\hat{\theta}_r}(z|f(X;\theta_f)) = p(z)$, and therefore $p(z|f(X;\theta_f)) = p(z)$. Using the Bayes' rule, we write

$$\begin{aligned} p(f(X;\theta_f)|z) &= \frac{p(z|f(X;\theta_f))p(f(X;\theta_f))}{p(z)} \\ &= p(f(X;\theta_f)), \end{aligned}$$

which holds for all $z\in\mathcal{Z}$ and all values of $f(X;\theta_f)$ and implies that f is a pivotal quantity. □

Proposition 2. *If there exists a saddle point $(\hat{\theta}_f,\hat{\theta}_r)$ for Eqn. 6 and 7 such that $E(\hat{\theta}_f,\hat{\theta}_r) = \mathbb{E}_X[H(p_{Y|X}) - H(p_Z)]$, then $f(\cdot;\hat{\theta}_f)$ is both an optimal classifier and a pivotal quantity.*

Proof. For fixed θ_f, the adversary r is optimal at $\hat{\theta}_r = \arg\max_{\theta_r} E(\theta_f,\theta_r) = \arg\min_{\theta_r}\mathcal{L}_r(\theta_f,\theta_r)$, in which case $p_{\hat{\theta}_r}(z|f(X;\theta_f)) = p(z|f(X;\theta_f))$ and $\mathcal{L}_r$ reduces to the expected entropy $\mathbb{E}_X[H(p_{Z|f(X;\theta_f)})]$ of the conditional distribution of the nuisance. The value function E can therefore be rewritten as

$$E'(\theta_f) = \mathcal{L}_f(\theta_f) - \mathbb{E}_X[H(p_{Z|f(X;\theta_f)})].$$

In particular, we have the lower bound

$$\mathbb{E}_X[H(p_{Y|X}) - H(p_Z)] \leq \mathcal{L}_f(\theta_f) - \mathbb{E}_X[H(p_{Z|f(X;\theta_f)})]$$

where the equality holds at $\hat{\theta}_f = \arg\min_{\theta_f} E'(\theta_f)$ only when

Algorithm 1 Adversarial training of a classifier f against an adversary r.

Inputs: training data $\{x_i, y_i, z_i\}_{i=1}^N$;
Outputs: $\hat{\theta}_f, \hat{\theta}_r$;
Hyper-parameters: Number T of training iterations, Number K of gradient steps to update r.

for $t = 1$ to T **do**
  for $k = 1$ to K **do** ▷ Update r
    Sample minibatch $\{x_m, z_m\}_{m=1}^M$ of size M;
    With θ_f fixed, update r by ascending its stochastic gradient $\nabla_{\theta_r} E(\theta_f, \theta_r) :=$

$$\nabla_{\theta_r} \sum_{m=1}^{M} \log p_{\theta_r}(z_m | f(x_m; \theta_f));$$

  end for
  Sample minibatch $\{x_m, y_m, z_m\}_{m=1}^M$ of size M; ▷ Update f
  With θ_r fixed, update f by descending its stochastic gradient $\nabla_{\theta_f} E(\theta_f, \theta_r) :=$

$$\nabla_{\theta_f} \sum_{m=1}^{M} \left[-\log p_{\theta_f}(y_m | x_m) + \log p_{\theta_r}(z_m | f(x_m; \theta_f)) \right],$$

  where $p_{\theta_f}(y_m|x_m)$ denotes $1(y_m = 0)(1 - f(x_m; \theta_f)) + 1(y_m = 1) f(x_m; \theta_f))$;
end for

- $\hat{\theta}_f$ minimizes the negative log-likelihood $\mathcal{L}_f$ of $X|Y$, which happens when $\hat{\theta}_f$ are the parameters of an optimal classifier and in which case $\mathcal{L}_f$ reduces to its minimum value $\mathbb{E}_X[H(p_{Y|X})]$,
- $\hat{\theta}_f$ maximizes the expected entropy $\mathbb{E}_X[H(p_{Z|f(X;\theta_f)})]$, which happens when all outcomes of $Z|f(X; \hat{\theta}_f)$ are equally likely. In other words, $p(z|f(X; \hat{\theta}_f)) = p(z)$ for all $z \in \mathcal{Z}$ and all values of $f(X; \theta_f)$ since we require a uniform prior by construction. Note that in the continuous case, the supremum of the differential entropy over continuous distributions on the same bounded support is also realized by the uniform distribution over that support.

When the lower bound is active, we have $p_{\theta_r}(z|f(X;\theta_f)) = p(z)$ because of the second condition, which induces that the optimal classifier $f(\cdot; \hat{\theta}_f)$ is also a pivotal quantity as a consequence of Proposition 1.

□

Proposition 2 suggests that if at each step of Algorithm 1 the adversary r is allowed to reach its optimum given f (e.g., by setting K sufficiently high) and if f is updated to improve $\mathcal{L}_f(\theta_f) - \mathbb{E}_X[H(p_{Z|f(X;\theta_f)})]$, then f should converge to a classifier which is both optimal and pivotal, provided such a classifier exists. On many problems of interest though, such a classifier may not exist because the nuisance parameter directly shapes the decision boundary, in which cases the lower bound $\mathbb{E}_X[H(p_{Y|X}) - H(p_Z)] < \mathcal{L}_f(\theta_f) - \mathbb{E}_X[H(p_{Z|f(X;\theta_f)})]$ is strict: f can either be an optimal classifier or a pivotal quantity, but not both simultaneously. In this situation, it is natural to rewrite the value function E as

$$E_\lambda(\theta_f, \theta_r) = \mathcal{L}_f(\theta_f) - \lambda \mathcal{L}_r(\theta_f, \theta_r), \tag{10}$$

where $\lambda \geq 0$ is a hyper-parameter controlling the trade-off between the performance of f and its independence with respect to the nuisance parameter. Setting λ to a large value will preferably enforces f to be pivotal while setting λ close to 0 will rather constraint f to be optimal.

Interestingly, let us emphasize that these results hold using only the (1D) output of f (in the case of binary classification) as input to the adversary. We could similarly enforce an intermediate representation of the data to be pivotal, e.g. as in [8], but this is in fact not necessary.

V. EXPERIMENTS

A. Toy example

As a guiding toy example, let us consider the binary classification of 2D data drawn from multivariate gaussians with equal priors, such that

$$x \sim \mathcal{N}(\mu = (0, 0), \sigma = 2 \times \mathbb{1}) \text{ when } Y = 0, \tag{11}$$
$$x \sim \mathcal{N}(\mu = (1, -1 + z), \sigma = 2 \times \mathbb{1}) \text{ when } Y = 1. \tag{12}$$

The continuous nuisance parameter Z represents in this case our uncertainty about the exact location of the mean of the second gaussian. Our goal is to build a classifier $f(\cdot; \theta_f)$ for predicting Y given X, but such that the probability distribution of $f(X; \theta_f)$ is invariant with respect to the nuisance parameter Z.

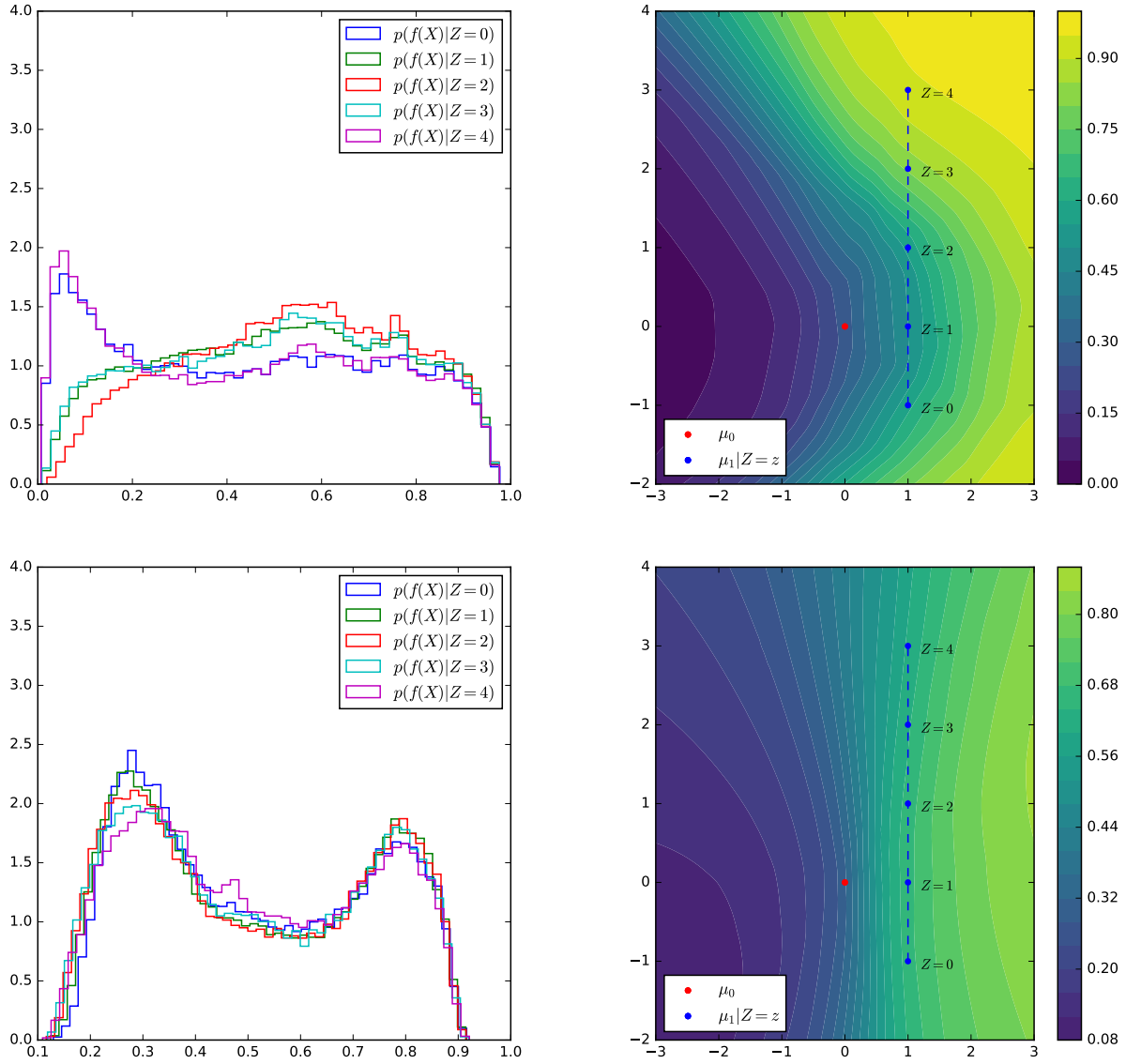

FIG. 2. Toy example. (Upper left) Conditional probability densities of the decision scores for $Z = 0, 1, \ldots, 4$ when f is built without adversarial training. The resulting densities are clearly dependent on Z, indicating that f is not pivotal. (Upper right) The corresponding decision surface, highlighting the fact that samples are easier to classify for values of Z closer to 4, hence partially explaining the dependency. (Lower left) Conditional probability densities of the decision scores for $Z = 0, 1, \ldots, 4$ when f is built with adversarial training, as outlined in Section III. The resulting densities are now almost identical to each other, indicating only a small dependency on Z. (Lower right) The corresponding decision surface, illustrating how adversarial training bends the decision function to erase the dependency on Z.

Assuming a uniform prior over the interval $\mathcal{Z} = [0; 4]$, we start by generating training data $\{x_i, y_i, z_i\}_{i=1}^N$, from which we build a neural network classifier f minimizing $\mathcal{L}_f(\theta_f)$ without adversarial training. As shown in the upper plots of Figure 2, the resulting classifier is not pivotal, as the conditional probability densities of its decision scores $f(X; \theta_f)$ show large discrepancies between values z of the nuisance. While not shown here, a classifier trained only from data generated at the nominal value $Z = 0$ would also not be pivotal.

By contrast, when using Algorithm 1 for training f jointly with a mixture density network r modeling the conditional distribution of the nuisance as observed through f, the resulting classifier becomes effectively almost independent on Z, as shown in the lower plots of Figure 2. In particular, the conditional probability densities of the decision scores $f(X; \theta_f)$ are now very similar to each other, indicating only a small dependency on the nuisance, as theoretically expected. The dynamics of adversarial training is illustrated in Figure 3, where

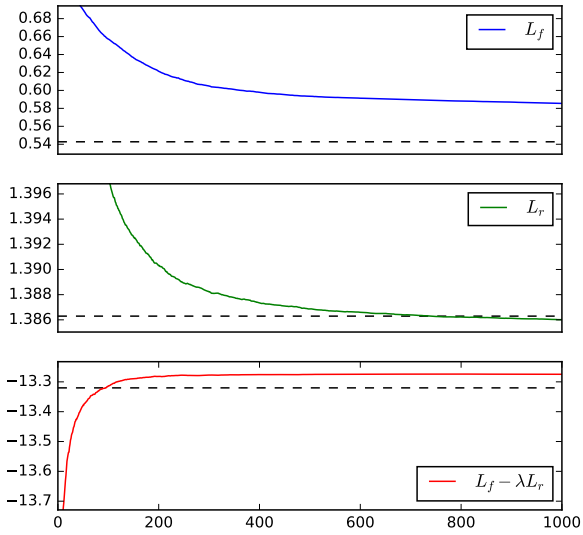

FIG. 3. Toy example. Training curves for $\mathcal{L}_f(\theta_f)$, $\mathcal{L}_r(\theta_f, \theta_r)$ and $\mathcal{L}_f(\theta_f) - \lambda\mathcal{L}_r(\theta_f, \theta_r)$. Adversarial training was performed for 1000 iterations, mini-batches of size $M = 128$, $K = 1$ and $\lambda = 10$.

the losses $\mathcal{L}_f$, $\mathcal{L}_r$ and $\mathcal{L}_f - \lambda\mathcal{L}_r$ are evaluated after each iteration of Algorithm 1. In the first iterations, we observe that f is worse than random because it rather tries to maximizes the adversarial loss $\mathcal{L}_r$ by producing decision scores $f(X; \theta_f)$ that are very unlikely under the current model p_{θ_r}, at the expense of making bad predictions with respect to the classification problem. As learning goes and p_{θ_r} becomes more accurate, minimizing E requires making predictions that are more accurate, hence decreasing $\mathcal{L}_f$, or that are less dependent on Z, hence shaping p_{θ_r} so that it becomes more uniform. Indeed, $\mathcal{L}_f$ eventually start decreasing, while remaining lower bounded by $\min_{\theta_f} \mathcal{L}_f(\theta_f)$ as approximated by the dashed line in the first plot. Similarly, $\mathcal{L}_r$ increasingly tends towards the differential entropy $H(p_Z) = \log(4)$ of the uniform distribution over $\mathcal{Z}$, as shown by the dashed line in the second plot. Finally, let us note that the ideal situation of a classifier that is both optimal and pivotal appears to be unreachable for this problem, as shown in third plot by the almost constant offset between $\mathcal{L}_f - \lambda\mathcal{L}_r$ and the dashed line approximating $\mathbb{E}_X[H(p_{Y|X}) - \lambda H(p_Z)]$.

[GL: Discuss practical considerations regarding the stability of training. A better alternative seems to update r until it is good enough, then update f, and update r again as soon as it becomes too bad. This is important e.g. for MDN to prevent weights to explode if the network is trained too much, and in which case gradients become useless.] [GL: Warn against setting λ too high, since it could simply force f to produce unlikely and very bad decisions.]

B. Physics example

VI. RELATED WORK

To account for systematic uncertainties, experimentalists in high energy physics typically take as fixed a classifier f built from training data for a nominal value z_0 of the nuisance parameter, and then propagate uncertainty [GL: add ref] by estimating $p(f(x)|z)$ with a parameterized calibration procedure. Clearly, this classifier is however not optimal for $z \neq z_0$. To overcome this issue, the classifier f is sometimes [GL: add refs] built instead on a mixture of training data generated from several nominal values $z_0, z_1, \ldots$ of the nuisance. While this certainly improves with respect to classification performance, there is however no guarantee that the resulting classifier is pivotal, as shown previously in Section V A. As an alternative, parameterized classifiers [2, 5] directly take (nuisance) parameters as additional input variables, hence ultimately providing the most statistically powerful approach for incorporating the effect of systematics on the underlying classification task. As argued in [12], such classifiers can however not be used on real data since the correct value z of the nuisance often remains unknown. This is typically not an issue in the context of parameter inference [5], where nuisance parameters are marginalized out, but otherwise often limits the range of their applications. In practice, parameterized classifiers are also computationally expensive to build and evaluate. In particular, calibrating their decision function, i.e. approximating $p(f(x, z)|z)$ as a continuous function of z, remains an open challenge. By contrast, constraining f to be pivotal yields a classifier which may not be optimal with respect to $Y|X, Z$, as discussed in Section IV, but that can otherwise be used in a wider range of applications, since knowing the correct value z of the nuisance is not necessary. Similarly, calibration needs to be carried out only once, since the dependence on the nuisance is now built-in.

In machine learning, learning a pivotal quantity can be related to the problem of domain adaptation [1, 4, 8, 9, 11, 14], where the goal is often stated as trying to learn a domain-invariant representation of the data. Likewise, our method also relates to the problem of enforcing fairness in classification [7, 15], which is stated as learning a classifier that is independent of some chosen attribute such as gender, color or age. For both families of methods, the problem can equivalently be stated as learning a classifier which is a pivotal quantity with respect to either the domain or the selected feature. In this context, [7, 8] are certainly among the closest to our work, in which domain invariance and fairness are enforced through an adversarial minimax setup composed of a classifier and an adversary discriminator. Following this line of work, our method can be regarded as a generalization that also supports the continuous case, which can be viewed as handling infinitely many domains, provided they can be continuously parameterized, or as enforcing fairness over

continuous attributes.

[GL: Check related work of cited references to see if important related work are missing.]

VII. CONCLUSIONS

ACKNOWLEDGMENTS

[1] Baktashmotlagh, M., Harandi, M., Lovell, B., and Salzmann, M. Unsupervised domain adaptation by domain invariant projection. In *Proceedings of the IEEE International Conference on Computer Vision* (2013), pp. 769–776.

[2] Baldi, P., Cranmer, K., Faucett, T., Sadowski, P., and Whiteson, D. Parameterized Machine Learning for High-Energy Physics. *arXiv preprint arXiv:1601.07913* (2016).

[3] Bishop, C. M. Mixture density networks.

[4] Blitzer, J., McDonald, R., and Pereira, F. Domain adaptation with structural correspondence learning. In *Proceedings of the 2006 conference on empirical methods in natural language processing* (2006), Association for Computational Linguistics, pp. 120–128.

[5] Cranmer, K., Pavez, J., and Louppe, G. Approximating likelihood ratios with calibrated discriminative classifiers.

[6] Degroot, M. H., and Schervish, M. J. *Probability and statistics*, 4 ed. 2010.

[7] Edwards, H., and Storkey, A. J. Censoring representations with an adversary.

[8] Ganin, Y., and Lempitsky, V. Unsupervised Domain Adaptation by Backpropagation. *ArXiv e-prints* (Sept. 2014).

[9] Gong, B., Grauman, K., and Sha, F. Connecting the dots with landmarks: Discriminatively learning domain-invariant features for unsupervised domain adaptation. In *Proceedings of The 30th International Conference on Machine Learning* (2013), pp. 222–230.

[10] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *Advances in Neural Information Processing Systems* (2014), pp. 2672–2680.

[11] Gopalan, R., Li, R., and Chellappa, R. Domain adaptation for object recognition: An unsupervised approach. In *Computer Vision (ICCV), 2011 IEEE International Conference on* (2011), IEEE, pp. 999–1006.

[12] Neal, R. M. Computing likelihood functions for high-energy physics experiments when distributions are defined by simulators with nuisance parameters. In *Proceedings of PhyStat2007, CERN-2008-001* (2007), pp. 111–118.

[13] Nix, D. A., and Weigend, A. S. Estimating the mean and variance of the target probability distribution. In *Neural Networks, 1994. IEEE World Congress on Computational Intelligence., 1994 IEEE International Conference on* (1994), vol. 1, IEEE, pp. 55–60.

[14] Pan, S. J., Tsang, I. W., Kwok, J. T., and Yang, Q. Domain adaptation via transfer component analysis. *Neural Networks, IEEE Transactions on 22*, 2 (2011), 199–210.

[15] Zemel, R. S., Wu, Y., Swersky, K., Pitassi, T., and Dwork, C. Learning fair representations.