Learning to Pivot with Adversarial Networks

Gilles Louppe,[1] **Michael Kagan**,[2] and **Kyle Cranmer**[1]

[1] *New York University*
[2] *SLAC National Accelerator Laboratory*

Many inference problems involve data generation processes that are not uniquely specified or are uncertain in some way. In a scientific context, the presence of several plausible data generation processes is often associated to the presence of systematic uncertainties. Robust inference is possible if it is based on a pivot – a quantity whose distribution is invariant to the unknown value of the (categorical or continuous) nuisance parameters that parametrize the family of data generation processes. In this work, we introduce a flexible adversarial training procedure to enforce the pivotal property on a predictive model, which allows one to tune the tradeoff between power and robustness. Under the assumption of existence, we derive theoretical results proving that the proposed procedure converges towards a predictive model that is both optimal and independent on the nuisance parameters, and demonstrate the effectiveness of this approach with a toy and physics examples.

I. INTRODUCTION

Machine learning techniques have been used to enhance a number of scientific disciplines, and they have the potential to transform even more of the scientific process. One of the challenges of applying machine learning techniques to scientific problems is the need to incorporate systematic uncertainties. The presence of systematic uncertainties affects both the robustness of inference and the metrics used to evaluate the performance of a particular data analysis strategy.

In this work, we focus on supervised learning techniques where the presence of systematic uncertainty can be associated to the fact that the data generation process is not uniquely specified. In other words, the lack of systematic uncertainties corresponds to the (rare) case that the process that generates training data is unique, fully specified, and an accurate representative of the real world data. By contrast, a common situation when systematic uncertainty is present is when the training data are not representative of the real data that a predictive model will be applied to in practice. Several techniques for domain adaptation have been developed to create predictive models that are more robust to this type of uncertainty. Another common situation is that there are several plausible data generation processes, specificied as a family of data generation processes parametrized by nuisance parameters, which can be categorical or continuous.

Assuming a probability model $p(X, Y, Z)$, where X are the data, Y are the target labels, and Z are the nuisance parameters, we consider the problem of learning a predictive model for Y conditional on the observed values of X that is robust to uncertainty in the unknown value of Z. More specifically, we introduce a learning procedure for enforcing the pivotal property on a predictive model by jointly training two neural networks in an adversarial fashion. We derive theoretical results proving that the proposed procedure converges towards a predictive model that is both optimal and independent on the nuisance parameters (if that models exists) or for which one can tune the trade-off between power and robustness. Finally, we demonstrate the effectiveness of this approach with a toy and physics examples.

The remainder of the paper is structured as follows. In Sec. II, we first formally define the problem of learning a pivotal predictive model. In sections III and IV, we describe how adversarial networks can be trained to obtain predictive models that are independent on nuisance parameters and show that the proposed procedure converges towards a model that is both optimal and robust. The effectiveness of the approach is then demonstrated empirically in sections V A and V B. In Sec. VI, we relate our work to close contributions from statistical inference, domain adaptation and fairness in classification. Finally, we gather our concluding remarks in Sec. VII.

II. PROBLEM STATEMENT

We begin with a family of data generation processes $p(X, Y, Z)$, where X are the data, Y are the target labels, and Z are the nuisance parameters that can be continuous or categorical. We assume training data $\{x_i, y_i, z_i\}_{i=1}^N$ for X conditional on both Y and Z. Our goal is to learn a predictive score function $f(\cdot; \theta_f) : \mathbb{R}^p \mapsto \mathcal{S}$ of parameters θ_f (e.g., a neural network-based probabilistic classifier) and minimizing a loss $\mathcal{L}_f(\theta_f)$ (e.g., the cross-entropy).

In addition, we would like that inference based on $f(X; \theta_f)$ should be robust to the value $z \in \mathcal{Z}$ of the nuisance parameter Z – which remains unknown at test time. A formal way of enforcing robustness is to require that the distribution of $f(X; \theta_f)$ conditional on Z (and possibly Y) be invariant to the value of the nuisance parameter Z. Thus, we wish find a predictive function f such that

$$p(f(X; \theta_f) = s|z) = p(f(X; \theta_f) = s|z') \qquad (1)$$

for all $z, z' \in \mathcal{Z}$ and all values $s \in \mathcal{S}$ of $f(X; \theta_f)$. In words, we are looking for a predictive function f which is a pivotal quantity [15] with respect to the nuisance parameter. That is, such that $f(X; \theta_f)$ and Z are independent random variables.

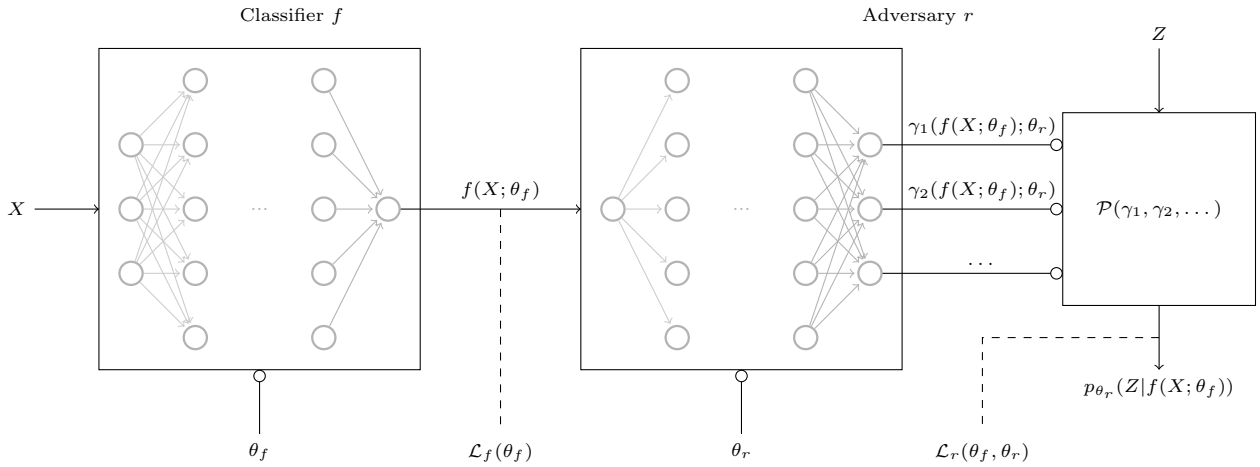

FIG. 1. Architecture for the adversarial training of a binary classifier f against a nuisance parameter Z. The adversary r models the distribution $p(z|f(X;\theta_f)=s)$ of the nuisance as observed only through the output $f(X;\theta_f)$ of the classifier. By maximizing the antagonistic objective $\mathcal{L}_r(\theta_f,\theta_r)$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

As stated in Eqn. 1, the pivotal quantity criterion is considered in the case where Y is marginalized out. In some situations however (see e.g., Sec. V B), class conditional independence of $f(X;\theta_f)$ on the nuisance Z is preferred, which can then be stated as requiring

$$p(f(X;\theta_f)=s|z,y)=p(f(X;\theta_f)=s|z',y) \qquad (2)$$

for one or several specified values $y\in\mathcal{Y}$ of Y.

III. METHOD

Joint training of adversarial networks was first proposed by [21] as a way to build a generative model capable of producing samples from random noise 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 of $g(Z)$ eventually converges towards the real distribution of X.

In this work, we repurpose adversarial networks as a means to constraint the predictive model f in order to satisfy Eqn. 1. As illustrated in Fig. 1, we pit f against an adversary model $r:=p_{\theta_r}(z|f(X;\theta_f)=s)$ of parameters θ_r and associated loss $\mathcal{L}_r(\theta_f,\theta_r)$. This model takes as input realizations s of $f(X;\theta_f)$, for the current value θ_f of f parameters, and produces as output a function $p_{\theta_r}(z|f(X;\theta_f)=s)$ modeling the posterior probability density that z parameterizes the sample observed as s. Intuitively, if $p(f(X;\theta_f)=s|z)$ varies with z, then the corresponding correlation can be captured by r. By contrast, if $p(f(X;\theta_f)=s|z)$ 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. 1.

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 $j=1,\dots,|\mathcal{Z}|$) is the estimated probability mass $p_{\theta_r}(z_j|f(X;\theta_f)=s)$. Similarly, if Z takes continuous values and if we assume some parameteric distribution $\mathcal{P}$ for $Z|f(X;\theta_f)=s$ (e.g., a mixture of gaussians, as modeled with a mixture density network [10]), then p_{θ_r} can be represented e.g. as network whose output j is the estimated value of the corresponding parameter γ_j of that distribution (e.g., the mean, variance and mixing coefficients of its components). As in [10, 26], the estimated probability density $p_{\theta_r}(z|f(X;\theta_f)=s)$ can then be evaluated for any $z\in\mathcal{Z}$ and any score $s\in\mathcal{S}$. 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 $p_{\theta_r}(z|f(X;\theta_f)=s)$ of sufficient capacity to represent the true distribution.

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

$$E(\theta_f,\theta_r)=\mathcal{L}_f(\theta_f)-\mathcal{L}_r(\theta_f,\theta_r) \qquad (3)$$

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

$$\hat{\theta}_f=\arg\min_{\theta_f}E(\theta_f,\hat{\theta}_r), \qquad (4)$$

$$\hat{\theta}_r=\arg\max_{\theta_r}E(\hat{\theta}_f,\theta_r). \qquad (5)$$

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 $\{x_m, z_m, s_m = f(x_m; \theta_f)\}_{m=1}^M$ of size M;
    With θ_f fixed, update r by ascending its stochastic gradient $\nabla_{\theta_r} E(\theta_f, \theta_r) :=$

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

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

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

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

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 $f : \mathbb{R}^p \mapsto [0, 1]$ 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:

$$\mathcal{L}_f(\theta_f) = \mathbb{E}_{x\sim X}\mathbb{E}_{y\sim Y|x}[-\log p_{\theta_f}(y|x)], \quad (6)$$

$$\mathcal{L}_r(\theta_f, \theta_r) = \mathbb{E}_{s\sim f(X;\theta_f)}\mathbb{E}_{z\sim Z|s}[-\log p_{\theta_r}(z|s)]. \quad (7)$$

The optimization algorithm consists in using stochastic gradient descent alternatively for solving Eqn. 4 and 5. Finally, in the case of a class conditional pivot, the settings are the same, except that the adversarial term $\mathcal{L}_r(\theta_f, \theta_r)$ is restricted to $Y = y$.

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. 1.

In this setting, the nuisance parameter Z is considered as a random variable of prior $p(Z)$, and our goal is to find a function $f(\cdot; \theta_f)$ such that $f(X; \theta_f)$ and Z are independent random variables. Importantly, classification of Y with respect to X is 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 $Y|X, Z$ (unless Z is made explicit and is included among the input variables in X). Results hold for a nuisance parameter Z taking either categorical or continuous values. By abuse of notation, $H(Z)$ denotes the differential entropy in this latter case. Finally, the proposition below is derived in a non-parametric setting, by assuming that both f and r have enough capacity.

Proposition 1. *If there exists a saddle point $(\hat{\theta}_f, \hat{\theta}_r)$ for Eqn. 4 and 5 such that $E(\hat{\theta}_f, \hat{\theta}_r) = H(Y|X) - H(Z)$, 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 $p_{\hat{\theta}_r}(z|f(X; \theta_f) = s) = p(z|f(X; \theta_f) = s)$ for all z and all s, and $\mathcal{L}_r$ reduces to the expected entropy $\mathbb{E}_{s\sim f(X;\theta_f)}[H(Z|f(X; \theta_f) = s)]$ of the conditional distribution of the nuisance. This expectation is nothing else than the conditional entropy of the random variables Z and $f(X; \theta_f)$ and can be written as $H(Z|f(X; \theta_f))$. Accordingly, the value function E can be restated as a function depending on θ_f only:

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

In particular, we have the lower bound

$$H(Y|X) - H(Z) \leq \mathcal{L}_f(\theta_f) - H(Z|f(X; \theta_f))$$

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

- $\hat{\theta}_f$ minimizes the negative log-likelihood of $Y|X$ under f, which happens when $\hat{\theta}_f$ are the parameters of an optimal classifier. In this case, $\mathcal{L}_f$ reduces to its minimum value $H(Y|X)$.

- $\hat{\theta}_f$ maximizes the conditional entropy $H(Z|f(X;\theta_f))$ since $H(Z|f(X;\theta)) \leq H(Z)$. Note that this latter inequality holds for both the discrete and the differential definitions of entropy.

When the lower bound is active, we have $H(Z|f(X;\theta_f)) = H(Z)$ because of the second condition, which happens exactly when Z and $f(X;\theta_f)$ are independent variables. In other words, the optimal classifier $f(\cdot;\hat{\theta}_f)$ is also a pivotal quantity.

□

Proposition 1 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 $\mathcal{L}_f(\theta_f) - H(Z|f(X;\theta_f))$ with sufficiently small steps, then f should converge to a classifier which is both optimal and pivotal, provided such a classifier exists. A formal proof of convergence of the alternating stochastic gradient descent procedure of Algorithm 1 in the case where a finite number K of steps is taken for r remains however to be proven.

On many practical problems, the assumption of existence of such a classifier may not hold because the nuisance parameter directly shapes the decision boundary. In this case, the lower bound

$$H(Y|X) - H(Z) < \mathcal{L}_f(\theta_f) - H(Z|f(X;\theta_f))$$

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

$$E_\lambda(\theta_f, \theta_r) = \mathcal{L}_f(\theta_f) - \lambda \mathcal{L}_r(\theta_f, \theta_r), \tag{8}$$

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 finally emphasize that these results hold using only the (1D) output s of $f(\cdot;\theta_f)$ (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 [19], 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

$$x \sim \mathcal{N}((0,0), \begin{bmatrix} 1 & -0.5 \\ -0.5 & 1 \end{bmatrix}) \qquad \text{when } Y = 0, \tag{9}$$

$$x \sim \mathcal{N}((1, 1+Z), \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}) \qquad \text{when } Y = 1. \tag{10}$$

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.

Assuming a gaussian prior $z \sim \mathcal{N}(0,1)$, we start by generating training data $\{x_i, y_i, z_i\}_{i=1}^N$, from which we train a neural network classifier f minimizing $\mathcal{L}_f(\theta_f)$ without considering its adversary r. The network architecture comprises 2 dense hidden layers of 20 nodes with ReLU activations, followed by a dense output layer with a single node with a sigmoid activation. As shown in the upper plots of Fig. 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.

Let us now consider the joint training of f against an adversary r implemented as a mixture density network modeling $Z|f(X;\theta_f)$ as a mixture of five gaussians. As for f, the network architecture comprises 2 dense hidden layers of 20 nodes with ReLU activations, but is followed by an output layer of 15 nodes corresponding to the means, standard deviations and mixture coefficients of the five gaussians. Output nodes for the mean values come with linear activations, output nodes for the standard deviations with exponential activations to ensure positivity, while output nodes for the mixture coefficients implement the softmax function to ensure positivity and normalization. When running Algorithm 1 as initialized with the classifier f obtained previously, adversarial training effectively reshapes the decision function so it that becomes almost independent on the nuisance parameter, as shown in the lower plots of Fig. 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 residual dependency on the nuisance, as theoretically expected. The dynamics of adversarial training is illustrated in Fig. 3, where 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 the global objective $\mathcal{L}_f - \lambda\mathcal{L}_r$ is minimized by making the classifier less accurate, hence the corresponding increase of $\mathcal{L}_f$, but which results in a classifier that is more pivotal, hence the corresponding increase of $\mathcal{L}_r$ and the total net benefit. As learning goes, minimizing E then requires making predictions that are more accurate, hence decreasing $\mathcal{L}_f$, or that are even less dependent on Z, hence shaping p_{θ_r} towards the prior $p(z)$. Indeed, $\mathcal{L}_f$ eventually starts to slightly decrease, while remaining lower bounded by $\min_{\theta_f} \mathcal{L}_f(\theta_f)$ as approximated by the dashed line in the first plot. Similarly, $\mathcal{L}_r$ tends towards the differential entropy $H(Z)$ of the prior (where $H(Z) = \log(\sigma\sqrt{2\pi e}) = 1.419$ in the case of a gaussian with unit variance), as shown by the dashed line in the second plot. Finally, let us note that the ideal situation

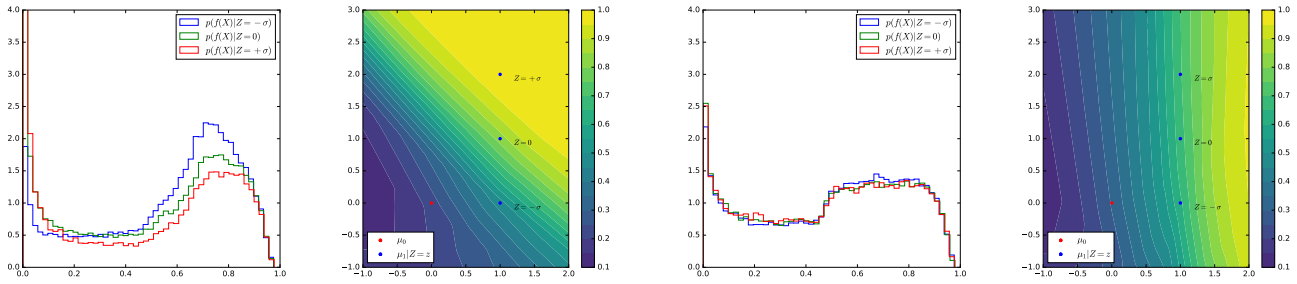

FIG. 2. Toy example. (Left) Conditional probability densities of the decision scores at $Z = -\sigma, 0, \sigma$ when f is built without adversarial training. The resulting densities are clearly dependent on the continuous parameter Z, indicating that f is not pivotal. (Middle left) The corresponding decision surface, highlighting the fact that samples are easier to classify for values of Z above to σ, hence partially explaining the dependency. (Middle right) Conditional probability densities of the decision scores at $Z = -\sigma, 0, \sigma$ 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. (Right) The corresponding decision surface, illustrating how adversarial training bends the decision function vertically to erase the dependency on Z.

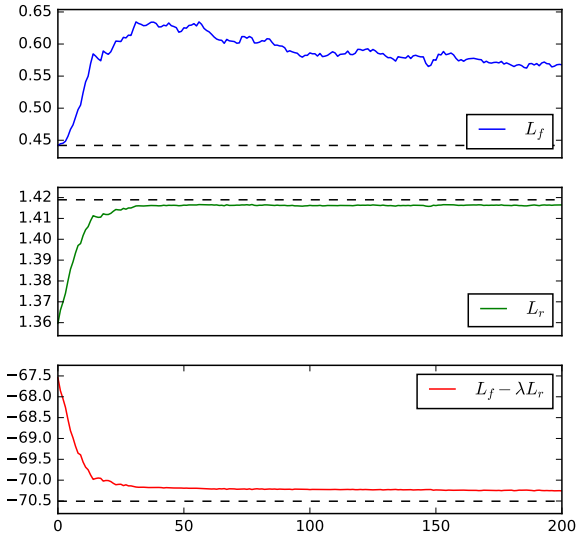

FIG. 3. Toy example. Training curves for $\mathcal{L}_f(\theta_f)$, $\mathcal{L}_r(\theta_f, \theta_r)$ and $\mathcal{L}_f(\theta_f) - \lambda\mathcal{L}_r(\theta_f, \theta_r)$. Adversarial training was performed for 200 iterations, mini-batches of size $M = 128$, $K = 500$ and $\lambda = 50$.

of a classifier that is both optimal and pivotal appears to be unreachable for this problem, as shown in the third plot by the offset between $\mathcal{L}_f - \lambda\mathcal{L}_r$ and the dashed line approximating $H(Y|X) - \lambda H(Z)$.

B. High energy physics example

Jets, or collimated sprays of particles produced by high energy quarks and gluons, are produced ubiquitously at high energy colliders like the LHC [17]. In a wide array of new physics models, new highly massive particles can decay through known heavy particles in the Standard Model (SM), like the W, Z, Higgs bosons, or top quarks which subsequently decay to multiple quarks. When these heavy SM particles have significant energy, the Lorentz boost causes their quark decay products to merge into a single jet which has a rich internal substructure (see e.g. [3, 4] and references therein). Distinguishing these boosted jets produced by heavy SM particles from vanilla quarks and gluons is a fundamental challenge in searching for signs of new high mass particles. Challenging in its own right, this classification problem is made all the more difficult by the presence of pileup, or multiple simultaneous proton-proton interactions occurring at the same time as the primary interaction. These pileup interactions produce additional particles that can contribute significant energies to jets unrelated to the underlying discriminating information. Thus pileup acts as a source of noise in the classification problem.

The classification challenge used here is common in jet substructure studies (see [5, 6, 13] and references therein): we aim to discriminate between jets produced by quark or gluon fragmentation (the background) and boosted W boson jets containing a collimated pair of quarks (the signal). We reuse the datasets used in reference [8]. Briefly summarizing, jets are built with the anti-k_T algorithm [12] with radius parameter $R = 1.2$, and trimmed [23] with subjets built with the k_T algorithm and parameter $f_{cut} = 0.2$. The features used in the classifiers are: trimmed jet invariant mass, N-subjettiness $\tau_{21}^{\beta=1}$ [28, 29], and the energy correlation functions [24] $C_2^{\beta=1}$, $C_2^{\beta=2}$, $D_2^{\beta=1}$, $D_2^{\beta=2}$.

As nuisance parameter, we consider events without pileup ($Z = 0$) and events with pileup ($Z = 1$), for which an average number of $\langle\mu\rangle = 50$ unrelated additional pileup interactions are overlaid. Our goal is to build an accurate classifier, for which we also want to minimize the effects due to the uncertainties on the nuisance. More specifically, we choose to recast the classification prob-

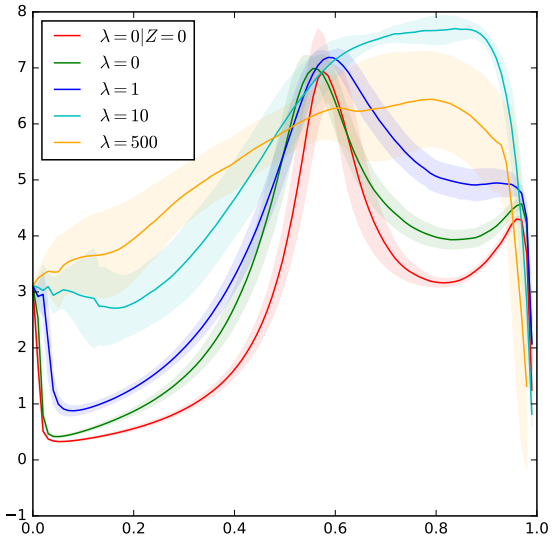

FIG. 4. Physics example. Approximate median significance as a function of the decision threshold on the output of f. As shown at $\lambda = 10$, trading classification accuracy for independence with respect to pileup results in a positive total net benefit in terms of statistical significance.

lem as a hypothesis test between signal+background (jets originating from W bosons or from quarks and gluons) and background only (jets originating from quarks and gluons only) and tune the decision threshold of our classifier by maximizing its approximate median significance (AMS), when uncertainties in the background are taken into account (see Eqn. 20 of [1]). Our motivation is that reducing the effects of uncertainties by requiring independence of Z with the classifier output $f(X;\theta_f)$ should allow for a larger maximum significance.

To minimize the effects of Z in the background events, we train a classifier using Algorithm 1 but consider the adversarial term $\mathcal{L}_r$ conditioned on $Y = 0$ only, as outlined in Sec. II. Both f and r are neural networks with 3 hidden layers of 64 nodes and ReLU activations, each terminated by a single final output node with a sigmoid activation. Experiments are performed on a subset of 150000 samples for training while AMS is evaluated on an independent test set of 5000000 samples. Both training and testing samples are weighted such that the signal-to-background ratio is 1/10, allowing us to probe the efficacy of the method proposed here in a background dominated environment. Results reported below are averages over 5 runs.

As Fig. 4 illustrates, without adversarial training (at $\lambda = 0|Z = 0$ when building a classifier at the nominal value $Z = 0$ only, or at $\lambda = 0$ when building a classifier on data sampled from $p(X, Y, Z)$), the maximum significance peaks at 7. By contrast, as the independence constraint is made stronger (for $\lambda > 0$) the AMS peak moves higher, with a maximum value around 7.8 for $\lambda = 10$. In other words, trading classification accuracy for independence with respect to pileup results in a positive total net benefit in terms of statistical significance. Setting λ too high however (e.g. $\lambda = 500$) results in a decrease of the maximum significance, by focusing the capacity of f too strongly on independence, at the expense of classification accuracy. As demonstrated in this example, controlling the classification versus pivot trade-off through λ therefore gives us a principled and effective approach for maximizing significance by desensitizing the classifier output $f(X;\theta_f)$ in the most beneficial way.

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 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 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 [9, 14] 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 [25], 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 [14], 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 [2, 7, 11, 19, 20, 22, 27], 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 [16, 18, 30, 31], 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, [16, 19] 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 and without making assumptions on the prior $p(Z)$, 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.

VII. CONCLUSIONS

In this work, we proposed a flexible learning procedure for building a predictive model that is independent on continuous or categorical nuisance parameters by jointly training two neural networks in an adversarial fashion. From a theoretical perspective, we showed that the approach leads to a predictive model that is both optimal and pivotal (if that models exists) or for which one can tune the trade-off between power and robustness. From an empirical point of view, we confirmed the effectiveness of our method on a toy and physics example.

[GL: Add future works.] [GL: Mention limitations of the current algorithm.]

ACKNOWLEDGMENTS

KC and GL are both supported through NSF ACI-1450310, additionally KC is supported through PHY-1505463 and PHY-1205376.

[1] Adam-Bourdarios, C., Cowan, G., Germain, C., Guyon, I., Kégl, B., and Rousseau, D. The higgs boson machine learning challenge. In *NIPS 2014 Workshop on High-energy Physics and Machine Learning* (2014), vol. 42, p. 37.

[2] Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., and Marchand, M. Domain-adversarial neural networks. *arXiv preprint arXiv:1412.4446* (2014).

[3] Altheimer, A., et al. Jet Substructure at the Tevatron and LHC: New results, new tools, new benchmarks. *J. Phys. G39* (2012), 063001.

[4] Altheimer, A., et al. Boosted objects and jet substructure at the LHC. Report of BOOST2012, held at IFIC Valencia, 23rd-27th of July 2012. *Eur. Phys. J. C74*, 3 (2014), 2792.

[5] ATLAS Collaboration. Performance of Boosted W Boson Identification with the ATLAS Detector. Tech. Rep. ATL-PHYS-PUB-2014-004, CERN, Geneva, Mar 2014.

[6] ATLAS Collaboration. Identification of boosted, hadronically-decaying W and Z bosons in $\sqrt{s} = 13$ TeV Monte Carlo Simulations for ATLAS. Tech. Rep. ATL-PHYS-PUB-2015-033, CERN, Geneva, Aug 2015.

[7] 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.

[8] Baldi, P., Bauer, K., Eng, C., Sadowski, P., and Whiteson, D. Jet substructure classification in high-energy physics with deep neural networks. *Physical Review D 93*, 9 (2016), 094034.

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

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

[11] 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.

[12] Cacciari, M., Salam, G. P., and Soyez, G. The Anti-k(t) jet clustering algorithm. *JHEP 0804* (2008), 063.

[13] CMS Collaboration. Identification techniques for highly boosted W bosons that decay into hadrons. *JHEP 12* (2014), 017.

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

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

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

[17] Evans, L., and Bryant, P. LHC Machine. *JINST 3* (2008), S08001.

[18] Feldman, M., Friedler, S. A., Moeller, J., Scheidegger, C., and Venkatasubramanian, S. Certifying and removing disparate impact. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2015), ACM, pp. 259–268.

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

[20] 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.

[21] 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.

[22] 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.

[23] Krohn, D., Thaler, J., and Wang, L.-T. Jet Trimming. *JHEP 1002* (2010), 084.

[24] Larkoski, A. J., Salam, G. P., and Thaler, J. Energy correlation functions for jet substructure. *Journal of High Energy Physics 2013*, 6 (2013), 108.

[25] 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.

[26] 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.

[27] 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.

[28] Thaler, J., and Van Tilburg, K. Identifying Boosted Objects with N-subjettiness. *JHEP 1103* (2011), 015.

[29] Thaler, J., and Van Tilburg, K. Maximizing boosted top identification by minimizing n-subjettiness. *Journal of High Energy Physics 2012*, 2 (2012), 93.

[30] Zafar, M. B., Valera, I., Rodriguez, M. G., and Gummadi, K. P. Fairness constraints: A mechanism for fair classification. *arXiv preprint arXiv:1507.05259* (2015).

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