Robust Neural Network for Causal Invariant Features Extraction

Shuxi Zeng[*]
zengshx777@gmail.com

Pengchuan Zhang[†]
penzhan@microsoft.com

Denis Charles[‡]
cdx@microsoft.com

Eren Manavoglu[§]
ermana@microsoft.com

Emre Kıcıman[¶]
emrek@microsoft.com

Abstract

Most machine learning approaches exploit correlational relationships in a training data set to predict a target variable. When these correlations are spurious or unreliable, this hampers the ability to generalize learned models to new environments. In contrast, models exploiting causal relationships between features and the outcome generalize better across environments. In this paper, we posit that these robust causal relationships can be identified by finding features that, when conditioned upon, render the outcome invariant across environments—that is, when the outcome is independent of the environment given a set of selected features with lower dimensions. We propose a neural network architecture for this task, comparing it with several existing approaches to exploit the causal invariant property, with a discussion on their motivations in a unified framework. Empirically, we perform a simulated experiment to demonstrate and compare the performance of the proposed method to the existing approaches. Finally, we measure its efficacy in a real world data set for advertisement click prediction.

1 Introduction

Machine learning and deep learning methods achieve great success in real applications. However, many of the existing methods solely focus on optimizing the training loss or accuracy, without regard to whether the relationships being exploited in a training data set are potentially unreliable correlational or more robust causal relationships. Thus, such methods suffer from the inherent biases such as confounding factors or selection bias. As a consequence, learned models encounter difficulties in transferring or adapting to a testing environment where the distribution shifts from the training set. Much research in the field of domain adaptation or transfer learning attempts to solve this problem by representing the features or learning an encoding of the raw features which are indistinguishable across different domains or data sources [1, 2]. However, the motivation for such feature extraction lacks theoretical justification and in some cases even becomes sub-optimal.

Motivated by recent work on invariant modeling and causal transfer learning [3, 4, 5], we paraphrase this problem causally and justify the robust features as being the direct causes to outcome. Furthermore, we enumerate four possible approaches to extract robust features and compare their performances on a colored MNIST dataset. Three of these approaches have been previously proposed

[*] Intern at Microsoft Research Ph.D. student at Department of Statistical Science, Duke University

[†] Researcher at Microsoft Research

[‡] Principal Applied Scientist at Bing Ads

[§] Partner Scientist at Bing Ads

[¶] Senior Principal Researcher at Microsoft Research

Preprint. Under review.

[1, 6, 4] while the last one is newly proposed by this paper. Also, we implement the proposed method to make robust predictions in one real online advertisement system for application.

2 Fundamental Motivation

One of the common failures of deep learning techniques—and machine learning methods more broadly—occurs when the model captures spurious correlations in the training data which to not hold in a test data set or deployed environment [7, 8, 9]. A simple example demonstrates this problem [4]. Consider Beery et al's example of training an image classifier to distinguish cows and camels [10] where, in the training data, the background of cows is green grass while that of camels is sandy desert. If this background provides a consist signal of the desired outcome label, the model will learn to rely on the background type, namely grass or sand, to make its predictions. Subsequently, if in the test data set or deployed environment, the photographer includes pictures of camels on grass and photos of cow on sand, the trained image classifier is unlikely to perform well.

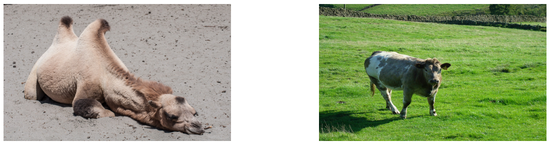

Figure 1: Camel with Sand as Background, Cow with Grass as Background, Training Set

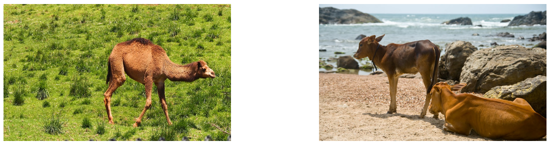

Figure 2: Camel with Grass as Background, Cow with Sand as Background, Testing Set

In contrast to the correlational image classifier, humans handle image classification tasks by exploiting underlying causal mechanisms. In this example, a human can tell whether a picture is of a cow or camel based on the shape of the object, which can be viewed as a "cause" of the picture being labeled as "cow" or "camel". This causal reasoning capability is robust across different background colors, lightning conditions, and other irrelevant, though perhaps correlated, features. However, the relationships between the "cause" and background colors or the relationships between the background and final labels are not robust, as shown in Figure 3. On the other hand, the relationship between the "cause" and the "result" is presumably robust as it is determined by some laws of nature. Therefore, capturing the direct causes of the outcome by exploiting the causal structure will give a robust model [3]. Nevertheless, it is usually hard to identify the causes of an object label from the observed data. In this work, we focus on the case where we have some variations in the environments [4]. We can define the environment as shown in Figure 4.

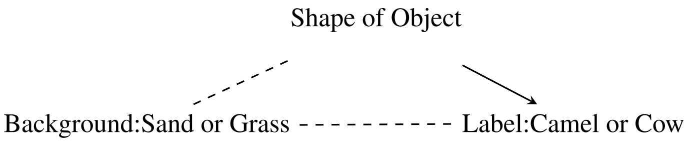

Figure 3: Causal Diagram for Image Classification Example, Solid Arrow for Robust Causal Relationships, Dashed Line for Spurious Relationship

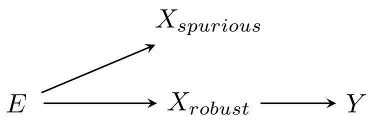

Figure 4: We posit that the robust features, X_{robust} are those that are causes of the outcome Y. Other features, $X_{spurious}$ may be spuriously correlated with the outcome, but such correlations will vary as the environment, E, changes.

The distribution of the direct causes themselves, or the confounding variables or some factors affected by the outcome, may vary across different environments. On the other hand, the causal mechanism, namely the relationships between direct causes and the outcome remain invariant across environments. This assumption cannot be justified empirically but perceived to be reasonable as the underlying natural mechanism should not change. In the next section, we will proceed in four different ways to define the invariant causal relationship and discuss the induced optimization problem to preserve the invariant property.

3 Proposed Algorithms

We present a general framework to define the causal invariant property in the learning problem. Suppose that we are interested in a supervised task predicting a target label Y based on raw features $\mathbf{X} \in \mathcal{R}^p$, such as the pixel of an image or the logs from an online advertisement system. We assume data is collected from at least two different environments and that the distribution of raw features differs across these environments. This difference may stem, for example, from the data generation mechanism. In the cow and camel example, the environments differ in the correlation between the object label and the background. We use $e = 1, 2, \cdots E$ to label the index of environment. Also, we represent the different probability distributions in various environments by $Pr^e(\cdot)$.

In many applications, the high-dimensional raw features may contain biases or confoundings. Therefore, we propose to learn an embedding or transformation of the raw features $T(x)$ which is of lower dimensions and robust in making prediction. Below, we discuss 4 approaches to define the robust features. Each of these methods builds a classifier or learner $G(\cdot)$ with the transformed features $T(X)$ as an input to predict label Y, with loss function $L(Y, G(T(X)))$. We present and compare these 4 distinct approaches to learn $T(X)$, including two proposed in prior work by [6] and [4]. Each of these approaches has different motivations and some are mutually compatible, indicating opportunities in future work to combine them for greater benefit.

Approach 1: Conditional Invariant Adversarial Network

Instead of matching the distribution of robust features $T(X)$ in different environments [1, 2],

$$\begin{aligned} \min_{G,T} \quad & \textstyle\sum_{e=1}^{E} E_{(x,y)\sim Pr^e(X,Y)}\{L(y, G(T(x))\}, & (1) \\ \text{s.t.} \quad & Pr^e(T(X)) = Pr^{e'}(T(X)), \text{for any } e \neq e', & (2) \end{aligned}$$

Li *et al* [6] propose to match the transformed features $T(X)$ conditioned on the final label Y across different environments or data source. Therefore, the definitions for the robust features induce the following constrained optimization problem,

$$\begin{aligned} \min_{G,T} \quad & \textstyle\sum_{e=1}^{E} E_{(x,y)\sim Pr^e(X,Y)}\{L(y, G(T(x))\}, & (3) \\ \text{s.t.} \quad & Pr^e(T(X)|Y) = Pr^{e'}(T(X)|Y), \text{for any } e \neq e'. & (4) \end{aligned}$$

Based on the constraint, this approach is actually treating the label Y as the direct cause for the robust features $T(X)$. In the context of cow and camel, the actual label causes the shape of the object. The assumed causal relationship is summarized in the Figure 5a.

$$E \longrightarrow Y \longrightarrow T(X)$$

(a) The Causal Diagram for Approach 1

$$E \longrightarrow T(X) \longrightarrow Y$$

(b) The Causal Diagram for Approach 3,4

Figure 5: The Causal Diagram for Different Approaches, the Implied Causal Relationship Between Environment E, Robust Features $T(X)$ and Label Y.

For implementations, they propose two losses to match the conditional distributions. Firstly, they build a domain classifier $D_j^e(\cdot)$ for each distinct value of label Y to classify the data point into different environment e based on the robust features. Specifically, $D_j^e(T(X))$ represents the predicted probability that sample is drawn from environment e given the label $Y = j$. They train the feature extractor in an adversarial way with these domain classifiers D_j^e to balance the conditional distribution of features $Pr^e(T(X)|Y)$ across domains. They add the loss in (5) to the objective in (3).

Build Class Conditional Domain Classifiers:

$$\text{min}_T \text{max}_{D_j} \sum_{e=1}^{E} E_{x\sim P^e(X|Y=j)} log D_j^e(T(x)), \forall j. \tag{5}$$

To avoid overfitting, Li et al [6] also proposes to build one unified classifier for all label values $D^e(\cdot)$ and reweight the loss function with the inverse of prior probability for each label value $Pr^e(Y = y)$ in different environments. The feature extractor $T(\cdot)$ is trained in adversarial way as in the first loss and the following loss is added to (3).

Build prior normalized marginal Domain Classifiers:

$$\text{min}_T \text{max}_D \sum_{e=1}^{E} E_{(x,y)\sim P^e(X,Y)} log(D^e(T(x))\beta^e(y), \beta^e(y) \propto \frac{1}{Pr^e(Y = y)}. \tag{6}$$

Approach 2: Posterior Probability Matching

The second approach is motivated by the assumption that robust features $T(X)$ should contains the same information on environment with the outcome Y. The conditional distribution of environment label $Pr(E = e|Y)$ can be calculated directly from data.

$$\text{min}_{G,T} \qquad \textstyle\sum_{e=1}^{E} E_{(x,y)\sim Pr^e(X,Y)}\{L(y, G(T(x))\}, \tag{7}$$

$$\text{s.t.} \quad Pr(E = e|T(X)) = Pr(E = e|Y), \text{for any } e. \tag{8}$$

For implementation, we can build a domain classifier D on top of the $T(x)$ to approximate the conditional distribution of environment label given $T(X)$. In the next step, we match the posterior probability for an environment label e given the robust features $T(X)$ and that given the label, $D^e(T(x)) = Pr(E = e|Y)$.Therefore, we add the Kullback–Leibler (KL) distance between these posterior distribution to objective (7):

$$\min_T \max_D \sum_{e=1}^{E} E_{(x,y)\sim Pr^e(X,Y)} KL(D^e(T(x))||Pr(E = e|Y = y)). \tag{9}$$

Approach 3: Invariant Risk Minimization

The third approach is proposed by Arjovsky [4] to minimize the invariant risk. Their approach is motivated by the observation that the robust features $T(X)$ should give the same optimal prediction in different environments,

$$\text{min}_{G,T} \qquad \textstyle\sum_{e=1}^{E} E_{(x,y)\sim Pr^e(X,Y)}\{L(y, G(T(x))\}, \tag{10}$$

$$\text{s.t.} \quad E^e(Y|T(X)) = E^{e'}(Y|T(X))\text{for any } e \neq e'. \tag{11}$$

They transformed the constraint above into the penalty on gradient of the final layer. Namely, they add the following term in the loss function to (10),

$$\text{min}_{G,T} \sum_{e=1}^{E} || \frac{\partial}{\partial w}_{|w=1} E_{(x,y)\sim Pr^e(X,Y)} L(y, G(wT(x)))||^2. \tag{12}$$

Approach 4: Robust Neural Network

We propose to define the robust features as the one that conditioning on which, the distribution of outcome should be invariant across environments. This corresponds to the idea in [3], which claims that the residual for the regressions from Y on the subset of features should the have the same distribution. We extend this invariance from regressions to a general condition here,

$$Pr^e(Y|T(X)) = Pr^{e'}(Y|T(X)), \text{for any } e \neq e' \tag{13}$$

We can therefore formulate the problem into the following constrained optimization programming:

$$\begin{aligned} \min_{G,T} \quad & \textstyle\sum_{e=1}^{E} E_{(x,y)\sim Pr^e(X,Y)} L(y, G(T(x)) && (14) \\ \text{s.t.} \quad & \qquad Y \perp\!\!\!\perp e|T(x) && (15) \end{aligned}$$

We are trying to find a representation of the raw features such that the we cannot make difference in predicting Y based on those features and meanwhile minimizing the training error for the model based on those features. This implies the same causal diagram with Approach 3, $T(X)$ being the direct causes and Y serving as the results, as shown in Figure 5b:

To solve this constraint optimization problem, we can transform the conditional independence condition into the following loss,

$$\min_{G,G',T} \quad \textstyle\sum_{e=1}^{E} E_{(x,y)\sim Pr^e(X,Y)} L(y, G(T(x)) + L(y, G'(T(x), e)) + \lambda||G'(y|T(x), e), G(y|T(x))||$$

where $G'(T(x), e)$ is the classifier for label Y with the robust features and environment label as an input. $||G'(y|T(x), e), G(y|T(x))||$ is some metric determine the discrepancy between the classifier based on the transformed features $T(X)$ only and the one including additional environment label e. We can choose the Jensen Shannon (JS) divergence here. The architecture of the robust neural network is drawn below.

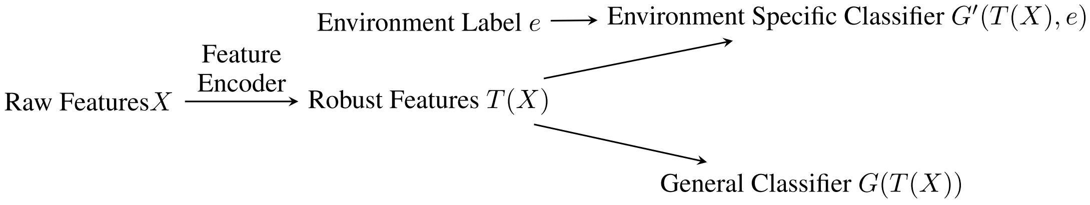
Environment Label e
Environment Specific Classifier $G'(T(X), e)$
Feature Encoder
Raw Features$X$
Robust Features $T(X)$
General Classifier $G(T(X))$

Figure 6: Architecture of the Robust Neural Network

4 Experiments

4.1 Colored MNIST Experiments

The first experiment is performed on colored MNIST dataset. We follow the same settings with the colored MNIST trial in [4]. The label Y is a binary label based on the actual digit in the handwriting image. We paint each greyscale MNIST image with either red or green such that the color is strongly correlated with the binary label. Specifically, we generate the synthetic data in the following steps:

1. Generate a binary label Y with each image:

$$\begin{aligned} \text{Digits:0-4} \quad &: \quad Y = 1 \text{ with } p = 0.75, Y = 0 \text{ with } p = 0.25. \\ \text{Digits:5-9} \quad &: \quad Y = 1 \text{ with } p = 0.25, Y = 0 \text{ with } p = 0.75. \end{aligned}$$

2. Color the MNIST image with either red or green according to digits.

$$\begin{aligned} \text{Digits:0-4} \quad &: \quad \text{Red with } p = e, \text{Green with } p = 1 - e. \\ \text{Digits:5-9} \quad &: \quad \text{Red with } p = 1 - e, \text{Green with } p = e. \end{aligned}$$

Hyper-paremter e defines the environment, different e values control the correlation between colors and the label Y. We choose the $e = 0.1, 0.2$ for the training environment and set

$e = 0.9$ for the testing environment. An illustration of the training environments and testing environments is shown in Figure 7 and Figure 8.

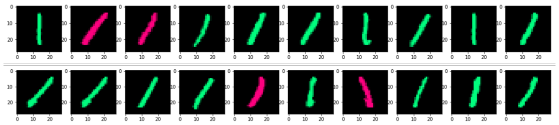

Figure 7: Examples of Colored MNIST for Digit "1" in Training Environment, $e = 0.1$(first row), $e = 0.2$ (second row). Most images of digit "1" are green.

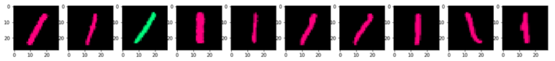

Figure 8: Examples of Colored MNIST for Digit "1" in Testing Environment $e = 0.9$. Most images of digit "1" are red.

We use the colored images as input to predict the binary label Y. With the following specifications, we posit the following causal mechanism in Figure 9, with color of images being the confounding factors and the spurious features while the actual digits being the robust features.

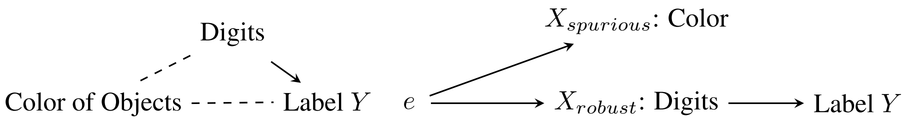

Figure 9: Left:Causal Diagram for the colored MNIST data, Solid Line for Robust Relationship, Dashed Line for Spurious Relationship. Right: The Causal Graph including the Environment Label

We compare the aforementioned 4 approaches, baseline method ignore the environment label by pooling the training data from two environments together and the model ignoring the color of images with only greyscale inputs. The empirical performances are summarized in Table 1.

Table 1: Performance Comparison for Different methods on Colored MNIST

Methods	Acc on Training Set %	Acc on Testing set %
Baseline	90.22	18.95
GreyScale	78.43	71.67
Random Guess	50.00	50.00
Oracle	75.00	75.00
(1) Conditional Inv. Adversarial Net.	90.13	28.14
(2) Posterior Prob. Matching	91.35	29.02
(3) Invariant Risk Minimization	69.76	66.26
(4) Robust Neural Network	80.97	60.16

We compare the accuracy of classification in the training and testing environments. The baseline method pooling two environments together has a higher accuracy on training set with a lower accuracy on the testing environment. We find the baseline method tends to use the color of the images as a key feature in predicting the label. As this is not actually a robust feature, its usage leads to worse performance in the testing environment, in which the relationship between color and label alternates. Among the 4 approaches to extracting robust features, Approach 3, Invariant Risk Minimization, demonstrates the best performance in terms of the testing accuracy, while our approach (Approach 4, Robust Neural Network) shows a comparable capability.

4.2 Click Calibration in Real Online Ads System

We perform a second experiment evaluating robust feature extraction on real-world datasets captured from an online advertising system. In this experiment, the task is to learn a click prediction model that is robust to policy changes in the advertising platform (e.g., ad selection or layout). Such a robust model allows better pre-deployment evaluation of the potential impact of the impact of policy changes [11, 12]. The challenge to learning a robust model is that, in normal operation, there are many confounding factors and spurious correlations. For example, the position of an ad on the page (mainline or bottom of the page) is an important feature for predicting whether a user will click. However, the position itself is also correlated with other relevant features, such as ad quality. Existing online advertisement system will pick up the ad with good quality to the top position (first slot in the mainline of the web page, e.g.). Therefore, it becomes hard to distinguish the causal effect of position from other features for the ads (quality, relevance eta). As a result, it becomes harder to predict whether a user will click if we make some manipulations on the existing system. We can view the position features as one of the direct causes for click behaviour, as the shape of cows or camels and other ads features as the background colors (except some of those features are also the direct causes), as shown in Figure 10.

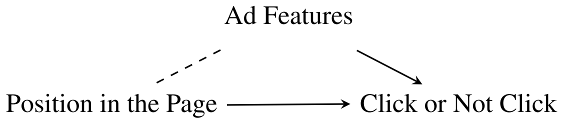

Figure 10: Causal Diagram for the Online Advertisement: Solid Line for Robust Relationship, Dashed Line for Spurious Relationship. The relationship between position and ads features is different under mainstream or randomized environments.

To better understand the true relationships between underlying features of advertisements and user clicks, the ads platform runs a small, ongoing experiment which randomizes ad selection, placement, layout and other policies. In our framework, this randomized experiment is an opportunity to collect data under a different environment, as compared to the mainstream mechanism. For example, the spurious relationship between ads features and the position is changing between the randomized and mainstream environments. In contrast to the mainstream environment, in the randomized setting, the position is randomly assigned to each ad regardless of its features. The changing relationship between position and other ads features are analogous to the changing relationship between the digit and color features in the previously described MNIST experiments.

To evaluate our framework in this experimental setting, we apply approach 4, robust neural networks, to extract robust features in logs and build a classifier for click prediction using these robust features, and use these to predict user clicks on advertisements in a third environment with a significant policy change. We train our model using 100K page impressions (including features of ads shown and resultant clicks) from a mainstream environment and another 100K impressions from a randomized environment. Our testing data consists of an environment with a radical policy change. As baselines, we compare our robust approach to the same neural network trained solely on 200k impressions from the mainstream environment or randomized environment, as well as a neural network trained on a mixture of data but without the regularization term provided by our robust neural network. We compare the following metrics, AUC, relative information gain (RIG), cumulative prediction error[6]. Table 2 demonstrates that the proposed method achieves the best performance compared with other neural net models, with a higher prediction power and lower bias.

[6] Relative information gain is defined as the $RIG = \text{Entropy(CTR)+LogLoss}/\text{Entropy(CTR)}$, Logloss is the loss given by the model. Cumulative prediction error is defined as $|\hat{\text{CTR}} - \text{CTR}|/\text{CTR}$, the bias for overall CTR divided by the actual CTR.

Table 2: Performance Comparison for Different methods on Online Ads Data

Methods	AUC	RIG	Prediction Error
NN trained on Mainstream Flight (200K)	0.901	0.3977	3.27%
NN trained on Randomized Flight (200K)	0.906	0.4050	2.03%
NN trained on Mixture (100K+100K)	0.912	0.4158	1.88%
(4) Robust Neural Network	0.924	0.4290	0.92%

5 Conclusion

We present a general framework to extract the robust features in the learning problem. If we can collect data from multiple sources or environments, we propose to treat the robust features as the direct causes for the outcome, conditioning on which the environment becomes independent with the outcome. We discuss four possible ways to exploit the causal invariant property and compare them on a synthetic colored MNIST experiment. We implement the proposed robust neural network method in a real Ads prediction task and demonstrate its advantages.

References

[1] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016.

[2] Fredrik Johansson, Uri Shalit, and David Sontag. Learning representations for counterfactual inference. In *International conference on machine learning*, pages 3020–3029, 2016.

[3] Jonas Peters, Peter Bühlmann, and Nicolai Meinshausen. Causal inference by using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 78(5):947–1012, 2016.

[4] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

[5] Mateo Rojas-Carulla, Bernhard Schölkopf, Richard Turner, and Jonas Peters. Invariant models for causal transfer learning. *The Journal of Machine Learning Research*, 19(1):1309–1342, 2018.

[6] Ya Li, Xinmei Tian, Mingming Gong, Yajing Liu, Tongliang Liu, Kun Zhang, and Dacheng Tao. Deep domain generalization via conditional invariant adversarial networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 624–639, 2018.

[7] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[8] David Lopez-Paz. From dependence to causation. *arXiv preprint arXiv:1607.03300*, 2016.

[9] Terrance de Vries, Ishan Misra, Changhan Wang, and Laurens van der Maaten. Does object recognition work for everyone? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 52–59, 2019.

[10] Sara Beery, Grant Van Horn, and Pietro Perona. Recognition in terra incognita. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 456–473, 2018.

[11] Léon Bottou, Jonas Peters, Joaquin Quiñonero-Candela, Denis X Charles, D Max Chickering, Elon Portugaly, Dipankar Ray, Patrice Simard, and Ed Snelson. Counterfactual reasoning and learning systems: The example of computational advertising. *The Journal of Machine Learning Research*, 14(1):3207–3260, 2013.

[12] Murat Ali Bayir, Mingsen Xu, Yaojia Zhu, and Yifan Shi. Genie: An open box counterfactual policy estimator for optimizing sponsored search marketplace. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, WSDM '19, pages 465–473, New York, NY, USA, 2019. ACM.