Causal Transfer Random Forest: Combining Logged Data and Randomized Experiments for Robust Prediction

Shuxi Zeng
Duke University
zengshx777@gmail.com

Murat Ali Bayir
Microsoft
mbayir@microsoft.com

Joel Pfeiffer
Microsoft
joelpf@microsoft.com

Denis Charles
Microsoft
cdx@microsoft.com

Emre Kıcıman
Microsoft
emrek@microsoft.com

ABSTRACT

It is often critical for prediction models to be robust to distributional shifts between training and testing data. Viewed from a causal perspective, the challenge is to distinguish the stable causal relationships from the unstable spurious correlations across shifts. We describe a *causal transfer random forest* (CTRF) that combines existing training data with a small amount of data from a randomized experiment to train a model which is robust to the feature shifts and therefore transfers to a new targeting distribution. Theoretically, we justify the robustness of the approach against feature shifts with the knowledge from causal learning. Empirically, we evaluate the CTRF using both synthetic data experiments and real-world experiments in the Bing Ads platform, including a click prediction task and in the context of an end-to-end counterfactual optimization system. The proposed CTRF produces robust predictions and outperforms most baseline methods compared in the presence of feature shifts.

CCS CONCEPTS

• **Computing methodologies → Machine learning**;

KEYWORDS

Random forest, Causal learning, Transfer learning, Robust prediction models, Covariate shifts

ACM Reference Format:
Shuxi Zeng, Murat Ali Bayir, Joel Pfeiffer, Denis Charles, and Emre Kıcıman. 2020. Causal Transfer Random Forest: Combining Logged Data and Randomized Experiments for Robust Prediction. In *In preparation for ACM Conference on Web Search and Data Miniing, March 8-12, 2021, JERUSALEM, ISRAEL.* ACM, New York, NY, USA, 9 pages.

1 INTRODUCTION

A central assumption of the majority of machine learning algorithms is that training and testing data is collected independently and identically from an underlying distribution. Contrary to this assumption, in many scenarios training data is collected under different conditions than the deployed environment [28]. For example, online services commonly use counterfactual models of user behavior to evaluate system and policy changes prior to online deployment [3]. In these scenarios, models train on interaction data gathered from previously deployed versions of the system, yet must make predictions in the context of the new system (prior to deployment). Other domains with distribution or covariate shifts include text and image classification [9, 33], information extraction [4], as well as prediction and now-casting [19].

Conventional machine learning algorithms exploit all correlations to predict a target value. Many of these correlations, however, can shift when parts of the environment are unrelated to our task change. Viewed from a causal perspective, the challenge is to distinguish causal relationships from unstable spurious correlations, as well as to disentangle the influence of co-varying features with the target value [1, 27, 29]. For example, in the counterfactual click prediction task we may wish to predict whether a user would have clicked on a link if we change the page layout (Figure 1). Training a prediction model based on current click logged data will find many factors related to an observation of a click (*e.g.*, display choices such as location and formatting, as well as factors related to ad quality and relevance). Yet, these factors are often entangled and co-vary due to platform policy, such as giving higher quality links more visual prominence through their location and formatting. In other cases, correlations may be unstable across environments as data generating mechanisms or the platform policy changes. A click prediction model based on this data may be unable to determine how much the likelihood of a click is due to relevant contextual features versus environmental factors. As long as the correlations among these features do not change, the prediction model will perform well. However, when the system is changed—perhaps a new page layout algorithm reassigns prominence or locations for links —the prediction model will fail to generalize.

One way to disentangle causal relationships from merely correlational ones is through experimentation [8, 16]. For example, if we randomize the location of links on a page it will break the spurious correlations between page location and all other factors. This allows us to determine the true influence or the “causal effect" of page location on click likelihood. Unfortunately, randomizing all important aspects of a system and policy is often prohibitively expensive, as employing the random platform policy in the system generally induces revenue loss compared with the a well-tuned

Prep for WDSM 2021 , March 8-12, 2021, JERUSALEM, ISRAEL

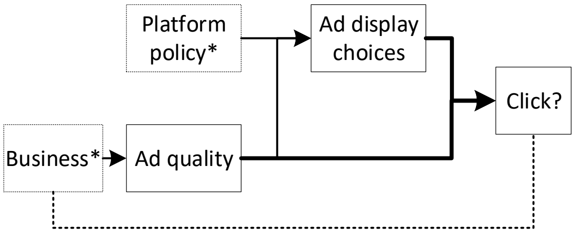


Figure 1: Challenges of robust prediction in a click prediction task: While click likelihood depends on display choices and ad quality, those two factors will co-vary in a way that changes as platform policy shifts. Other correlations (e.g,. business attributes) are unstable across environments.

production system. Gathering the scale of randomization data necessary for building a good prediction model is frequently not possible. Therefore, it is desirable to efficiently combine the relatively small scale randomized data and the large scale logged data for robust predictions after the policy changes.

In this paper, motivated by an offline evaluation application in the sponsored search engine, we describe a *causal transfer random forest* (CTRF). The proposed CTRF combines existing large-scale training data from past logs (**L-DATA**) with a small amount of data from a randomized experiment (**R-DATA**) to better learn the causal relationships for robust predictions. It uses a two-stage learning approach. First, we learn the CTRF tree structure from the **R-DATA**. This allows us to learn a decision structure that disentangles all the relevant randomized factors. Second, we calibrate each node (such as calculating the click probability) of the CTRF with both the **L-DATA** and the **R-DATA**. The calibration step allows us to achieve the high-precision predictions that are possible with large-scale data. Further, we complement our intuitions with theoretical foundations, showing that the model structure training on randomized data should provide a robust prediction across covariate shifts.

Our contributions in this paper are 3-fold. Firstly, we introduce a new method for building robust prediction models that combine large-scale **L-DATA** with a small amount of **R-DATA**. Secondly, we provide a theoretical interpretation of the proposed method and its improved performance from the causal reasoning and invariant learning perspective. Lastly, we provide an empirical evaluation of the robustness improvements of this algorithm in both synthetic experiments and multiple experiments in a real-world, large-scale online system at Bing Ads.

The Supplementary Material including reproducible code, experiment details and theorem proofs is provided in an anonymous GitHub repository :https://anonymous.4open.science/r/f874aa27-fd0c-46a5-b1c9-104cd28b9bc2/

2 RELATED WORK

2.1 Off-policy Learning in Online Systems

This work is motivated from the task of performing offline policy evaluation in the online system [6, 20]. Occasionally, we would like to know the outcome of performing an unexplored tuning in the current system, which is also known as the counterfactual outcome. For example, we are interested in the change in users click probability after modifying the auction mechanism in the online ads system [31]. Sometimes, the modifications can be drastic from the previous policy. Instead of running the costly online A/B testing [34], some offline methods are frequently used to predict the counterfactual outcomes based on the existing logged data from the current system. One novel solution is to build the model-based simulator. Specifically, we build the model simulating the users behaviour and measure the metrics change after implementing the proposed policy changes in the simulator [3]. We usually train the user-simulator model on the **L-DATA** generating under previous platform policy. As a result, the covariate shift problem happens if the proposed change is drastic.

2.2 Transfer Learning and Domain Adaptation

The discrepancy across training (large scale logged data e.g.) and testing (data after policy change e.g.) distribution is a long-standing problem in the machine learning community. Classic supervised learning might suffer from the generalization problem when the training data has a different distribution with the data for testing, which is also referred to the covariate (or distribution or dataset) shift problem, or the domain adaptation task [5, 9, 28]. Specifically, the model learned on a training data (source domain) is not necessarily minimizing the loss on the testing distribution (target domain). This hampers the ability of the model to transfer from one distribution or domain to another one.

Some researchers propose to correct for the difference through sample reweighting [14, 24, 30]. Ideally, we wish to weight each unit in the training set so that we can learn a model minimizing the loss averaged on the testing distribution after reweighting. However, this strand of approaches requires the knowledge of the testing distribution to estimate the density and is likely to fail when the testing distribution deviates a lot from the training distribution, with extreme values in the density ratio. Another type of methods is feature based. Some approaches aim at learning the features or representations that have predictive power while remaining a similar marginal distribution across source and target domain [10, 35]. However, the balance on marginal distributions does not ensure a similar performance on the target domain. We need to justify the predictive performance for the learnt features on the target domain.

2.3 Causality and Invariant Learning

Recently, some methods adapt the idea from causal inference to define the transfer learning with assumptions on the causality relationship among the features [1, 13, 18, 21, 22, 27, 29]. Specifically, researchers paraphrase the transfer difficulty as the confounding problem in causal inference literature [15, 25]. The reason for poor generalization performance is that the model is learning some spurious correlation relationships on the source domain, which are not expected to hold on the target domain. The invariant features across the domains should be the direct causes of the outcome (suppose being not intervened), as the causality relationship is presumably to be stable across training and testing distribution [26]. Our work focus on utilizing the **R-DATA** generating from a random policy, which is formally defined later, to exploit the causal relationship with limited sample size. Within the same causality framework, our model learns the invariant features that can transfer to the unknown target domain and be robust to severe covariate shifts.

3 CAUSAL TRANSFER RANDOM FOREST

In this section, we formulate the covariate shift problem and the transfer task. First, we formalize the problem and illustrate its role in sponsored search. Second, we introduce our proposed causal transfer random forest method, which can efficiently extract causality information from randomized data and improve generalization for a new testing distribution. Third, we provide theoretical interpretation for the proposed algorithm with causal reasoning.

3.1 Problem Setup

We formalize the covariate shift and transfer task mathematically. Let $y \in \mathcal{Y}$ be a binary outcome label given contextual features $x \in \mathcal{R}^p$ and intervenable features, $z \in \mathcal{R}^{p'}$. We desire a model to map from the feature space to a distribution over the outcome space, *i.e.* learning the conditional distribution $p(y|x,z)$. Taking our motivating application, sponsored search, as a concrete example, the contextual features x include user context and the query issued by the user; the features z encode aspects that the publishers can manipulate, for instance, the location or the quality of the ads; and y is whether or not a user clicked on the ad. In practice, an advertising system takes many steps to create the pages showing the ads.

The feature shift problem arises when there is a drastic change in the joint features distribution of $p(x,z)$. This shift might happen if the marginal distribution of contextual feature $p(x)$ varies. More commonly, the shift occurs when $p(z|x)$ changes to another distribution $p^*(z|x)$, namely, we change the data generating mechanism for z. This can happen when the platform policy change in the sponsored search system. In this case, the model learned from the training distribution $p(x,z) = p(x)p(z|x)$ might not generalize to the new distribution $p^*(x,z) = p(x)p^*(z|x)$. Therefore, we wish to learn a model $p(y|x,z)$ that is robust to the feature distribution, which can be safely transferred from original feature distribution $p(x,z)$ to the new $p^*(x,z)$.

We factorize the data (x,z,y) in the following way[6]:

$$p(x,z,y) = p(x)p(z|x)p(y|x,z), \tag{1}$$

where $p(x)$ denotes the distribution of contextual variable, $p(z|x)$ represents how the platform manipulates certain features, such as the process of selecting ads and allocating each ad to the position on a page, which involves a complicated system including auction, filtering and ranking decisions [31]. Here $p(y|x,z)$ is the user click model. One question of interest is how the click through rate $E(y)$ changes if we make modifications to the system, *i.e.*, replacing the usual mechanism $p(z|x)$ with a new one $p^*(z|x)$,

$$E^*(y) = \int\int\int p(x)p^*(z|x)p(y|x,z)\mathrm{d}x\mathrm{d}z. \tag{2}$$

Feature shifts happen if some radical modifications are proposed, namely $p(z|x)$ differs significantly from $p^*(z|x)$. The user click model $p(y|x,z)$ cannot produce a reliable estimate for the new click through rate $E^*(y)$ as we usually learn the click model based on $p(x,z)$ while the testing data for prediction is drawn from $p^*(x,z)$. As z depends on x differently under various policies, the correlation between z and y might change after policy changes from $p(z|x)$ to $p^*(z|x)$. In such a scenario, we wish to build a model that can transfer from training distribution $p(x,z)$ to the target distribution $p^*(x,z)$, allowing one to evaluate the impact of radical policy changes.

Currently, some publishers run experiments to randomize the features like the layout and advertisement in each impression shown to the user, which makes z independent of x. Now, we formally define the **R-DATA** as the data generated from $p(x)p(z)$, usually limited in size due to the low performance and revenue of a random policy. Meanwhile, we possess a large amount of past log data from the distribution $p(x)p(z|x)$, which we call **L-DATA**. This leads to the opportunity to more efficiently use **R-DATA** by pooling it with large-scale **L-DATA**.

Although our approach is motivated by the online advertising setting, it is not restricted to this domain or binary classification task. We aim at building a robust model $p(y|x,z)$ transferring from the smaller **R-DATA** and the large scale **L-DATA** to the targeting source $p^*(x,z)$. We focus on the case that $p^*(x,z)$ differs drastically from $p(x,z)$, which is either due to the change in the policy $p(z|x)$ or the variation in contextual features $p(x)$. Although in this application, we may know $p^*(x,z)$ in advance, the proposed method does not require any prior knowledge on the density of targeting source.

3.2 Proposed Algorithm

We base our algorithm on the random forest method [7], adapting prior work on honest causal trees and forests [2, 32]. Usually, the tree-based method is composed of two stages [12]: building decision boundary and calibrating each leaf value at the end of the branch to produce an estimate p_i. Furthermore, the random forest framework performs bagging on the training data and building decision tree on each bootstrap data to reduce variance. Advantages of random forests include their simplicity and ability to be paralleled.

To handle the feature shifts problem and use **R-DATA** efficiently, we propose the Causal Transfer Random Forest (CTRF) algorithm. The framework is shown in Figure 2. We propose to do bagging and build decision trees solely on the **R-DATA** and then calculate the predicted value (*e.g.*, click probability) on the nodes of each tree with pooled **R-DATA** and **L-DATA**. We make calibrations and aggregate over all trees with the simple average here, which can be extended to other approaches. We describe the detailed algorithm in Algorithm 1.

We design the algorithm with the intuition that the **R-DATA** reduces the problem of spurious correlation, one of the main reasons for the non-robustness of previous methods. Specifically, some of the correlations between z and the outcome y are influenced by the underlying generating mechanism, $p(z|x)$. In such cases, the correlation is spurious in the sense that it will disappear or change if we modify $p(z|x)$ to $p^*(z|x)$. The model trained on $p(x,z)$ will exploit those spurious correlations without the knowledge that the correlations will not hold on distribution $p^*(x,z)$. It is important to note that the spurious and non-spurious components of z's correlation with y are often not well-aligned with the raw feature representation of z. That is, this is not a feature selection problem.

Figure 3 demonstrates a spurious correlation instance in the ads system, depicting the relationships between ads relevance x, position z and the click outcome y. The solid lines represent the "stable" relationship or effect between the ads relevance or the position and the click, while the dashed line stands for the relationship we can

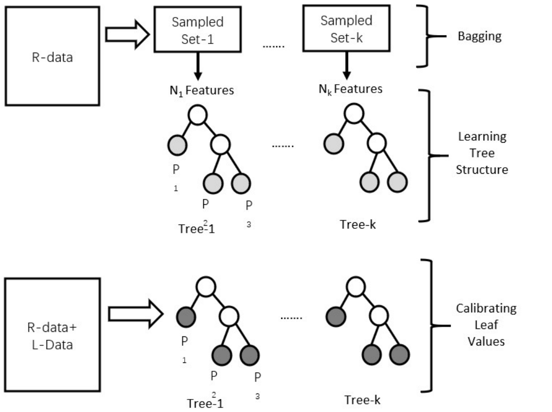


Figure 2: CTRF: building random forest from R-DATA and L-DATA

Algorithm 1: Causal Transfer Random Forest

```
Input: R-DATA $\mathcal{D}^R = \{(x_i, z_i, y_i), i \in \mathcal{I}^R\}$, L-DATA $\mathcal{D}^L = \{(x_i, z_i, y_i), i \in \mathcal{I}^L\}$ and the prediction point $(x^*, z^*)$.
Hyperparameters: bagging ratio: $r_{\text{bag}}$; feature subsampling ratio: $r_{\text{feature}}$; number of trees: $n^{\text{tree}}$.
Bagging: sample the data $\mathcal{D}^R$ with replacement for $n^{\text{tree}}$ times with sampling ratio $r_{\text{bag}}$ and sample on the feature set $(x, z)$ for each bootstrap data with ratio $r_{\text{feature}}$.
for $b = 1$ to $n^{\text{tree}}$ do
    Learn decison tree For the bootstrapped data, $\{(x_i^b, z_i^b, y_i^b)\}$, build decision tree $\mathcal{T}_b$ and corresponding leaf nodes $\mathcal{L}_j^b \subset \mathcal{R}^{p+p'}$, $j = 1, 2, \cdots, L_b$, $L_b$ is the number of nodes for $\mathcal{T}_b$ by maximizing the Information Gain (IG) or Gini Score.
    Calibrations For each node $\mathcal{L}_j^b$, we calculate the predicted value by the mean value of samples in this node: $\hat{y_j}^b = \bar{y}_i, (x_i, z_i) \in \mathcal{L}_j^b, i \in \mathcal{I}^R \cup \mathcal{I}^L$.
end for
Predictions Collect the predicted value $\hat{y}^b$ for each $\mathcal{T}_b$ by examining the node that $(x^*, z^*)$ belongs and produce a prediction after aggregation, such as $\hat{y} = \bar{\hat{y}}^b$.
Output Random forest $\{\mathcal{T}_b, b = 1, \cdots, n^{\text{tree}}\}$ and prediction $\hat{y}^*$.
```

manipulate. In the **L-DATA**, the position is not randomly assigned but instead associated with other features like ads relevance[6]. We tend to allocate ads of higher relevance to the top of the page. However, the correlation between position and click changes if we alter the policy allocating the position based on the relevance, namely $p(z|x)$. Despite the correlation between position and click being partially spurious, there is still a causal connection as well—higher positioned ads do attract more clicks, all else being equal.

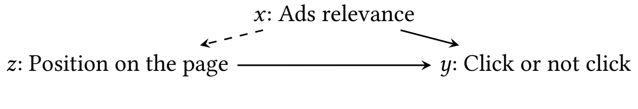


Figure 3: Causal Directed Acyclic Graph (DAG) for the online advertisement system

Suppose the tree algorithm makes a split on the position feature, subsequently it becomes hard to detect the importance of relevance in two sub-branches split by position. As a result, if we only train on **L-DATA**, the decision tree is likely to underestimate the importance of ad relevance. We wish the decision tree structure we learn to disentangle the unstable or spurious aspects of the correlation among the features and only learn the "stable" relationships. This task can be accomplished with the **R-DATA** as it removes the spurious correlation. We formally define the "stable" relationship and prove why **R-DATA** can learn those relationships in the next section.

3.3 Interpretations from Causal Learning

In this section, we justify our intuitions in the previous sections theoretically based on the results in causal learning. Previous literature builds the connections between the capability to generalize and the conditional invariant property. Theorem 4 in [29] demonstrates that if there is a subset of features S^* that are conditionally invariant, namely the conditional distribution $y|S^*$ remains unchanged across different distributions of $p(x, z, y)$, then the model built on those features S^* with pooling data, $E(y_i|S_i^*)$, gives the most robust performance. The robustness is measured by the worst performance with respect to all possible choices of the targeting distribution $p(x, z)$, which further ensures the model can transfer. This theorem indicates that we should build model on the set of features or the transformed features with conditional invariant property.

However, learning the stable features is not simple given we have only two types of distribution, The next theorem from [27, 29] states the relationship between conditional invariance and causality. Specifically, if we assume there are causal relationships or structural equation models (SEM) [25], the direct causes of the outcome are the conditionally invariant features , $S^* = \text{PA}_Y$, where PA_Y denotes the parents/direct causes for the outcome y.

With two well-established theorems above, we can look for the direct causes instead of the conditional invariant features. The following theorem shows that the **R-DATA** offers such opportunity.

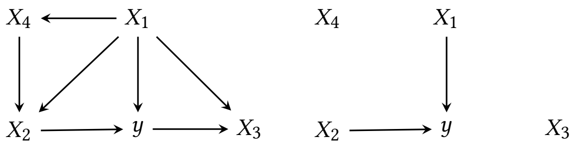
THEOREM (RETAIN STABLE RELATIONSHIPS WITH **R-DATA**). *Assume (x_i, z_i, y_i) can be expressed with a direct acyclic graph (DAG) or structural equation model (SEM). Then the model trained on **R-DATA**, $p(x_i, z_i) = p(x_i^1)p(x_i^2)\cdots p(x_i^p)p(z_i^1)p(z_i^2)\cdots p(z_i^{p'})$ is consistent for the most robust prediction:*

$$\hat{E}(y_i|x_i, z_i) \Rightarrow E(y_i|\boldsymbol{PA}^Y) = E(y_i|S_i^*) \tag{3}$$

Although there is a gap between the **R-DATA** we have and the one in the assumption as we cannot randomize the contextual features x, randomizing on the manipulable features z will suffice in practice as the correlation between x an y is likely to be stable. The theorem above suggests if the model is trained on **R-DATA**, it actually relies on the direct causes or robust features S_i^* to make prediction. The detailed theorem proof is provided in the Supplementary Material.

Figure 4 demonstrates this idea. Compared with Figure 4 (a), **R-DATA** in Figure 4 (b) removes all the effects other than the direct causes of y (PA_Y is (X_1, X_2) here), which indicates that the model trained with **R-DATA** will pick up the features that are robust for predictions.

Likewise, CTRF firstly learns the structure of the model or identifies the stable features for splitting the trees merely with the **R-DATA**. With our random forest method, the stable features are the leaves sliced in the decision tree, which can be viewed as a



(a) L-DATA (b) R-DATA

Figure 4: Causal DAG in L-DATA and R-DATA, only direct causes or stable predictors (X_1, X_2) remain correlated with y in R-DATA

transformation of the raw features. This step serves as an analogy to search for the direct causes or extract robust features. The calibration step on the leaf values with pooled data corresponds to make predictions conditioning on all robust features. The second step will not be "contaminated" by the spurious correlation in **L-DATA** as the the decision tree structure has already identified a valid adjustment set with **R-DATA** and is conditioning on that. We also test whether the proposed method can pick up the stable features in the synthetic experiments.

4 EXPERIMENTS ON SYNTHETIC DATA

4.1 Setup and Baselines

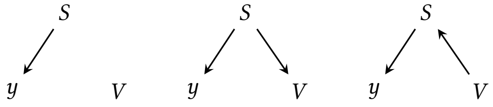
In this part, we evaluate the proposed method and compare with several baseline methods in the presence of covariate shifts. Given it is a novel scenario (small amount of **R-DATA** with large **L-DATA**), we design two synthetic experiments to create an artificial case that the data generating mechanism $p(z|x)$ changes. The first experiment specifies the causality relationship between variables explicitly. The second experiment is a simulated auction similar to the real-world online, in which the relationship between variables are specified implicitly. In both experiments, we have some parameters controlling the degree of covariate shift which allows us to evaluate the performance against different degree of distributional variation.

In our experiments, we compare the *causal transfer random forest* (CTRF) with the following methods: logistic regression (LR) [23], Gradient Boosting Decision Tree (GBDT) [17], logistic regression with sampling weighting (LR-IPW), Gradient Boosting Decision Tree with sample reweighting (GBDT-IPW), random forest model trained on **R-DATA** (RND-RF), random forest model trained on **L-DATA** (CNT-RF), random forest model trained with the **L-DATA** and **R-DATA** pooling together (Combine-RF). Among all those methods, LR-IPW and GBDT-IPW are designed to handle distribution shifts with a proper weighting with ratio of densities [5, 14]. Implementation details are included in the Supplementary Material.

As our method is designed to handle extreme covariate shifts, we evaluate different methods in terms of the performance on the shifted testing data only. Although our method is not restricted to classification task, we only focus on the binary outcome to be coherent with our motivated application from ads click. For binary classification task, we focus on the following two metrics, AUC (area under curve) and the cumulative prediction bias, $|\bar{\hat{y}}_i - \bar{y}_i|/\bar{y}_i$, which is the adjusted difference in the mean value of predicted values and actual outcomes. AUC captures the prediction power of the model while the cumulative prediction bias captures how our method can predict the counterfactual change, such as the change in the overall click rate.

4.2 Synthetic Data with Explicit Mechanism

We generate the data in a similar fashion with the experiments in [18]. We generate two sets of features S, V for predictions. S represents the stable feature or the direct cause of the outcome while V represents the unstable factors that have spurious correlation with the outcome. We consider three possible scenarios for the relationships between (S, V): (a)$S \perp\!\!\!\perp V$, S and V are independent; (b) $S \longrightarrow V$, S is the cause for V; (c) $V \longrightarrow S$, V is the cause for S. Figure 5 demonstrates these three cases. In all cases, $S = (S_1, \cdots, S_{p_s})$ is the stable feature while $V = (V_1, \cdots, V_{p_v})$ is the possible unstable factors sharing spurious correlation with the outcome.



(a) $S \perp\!\!\!\perp V$ (b) $S \rightarrow V$ (c) $V \rightarrow S$

Figure 5: Three possible relationships among the variables

In case (a), we generate (S, V) from independent standard Normal distributions and transform them into the binary vectors,

$$\tilde{S_j}, \tilde{V_k} \sim \mathcal{N}(0,1), \quad S_j = \mathbf{1}_{\tilde{S_j}>0}, V_k = \mathbf{1}_{\tilde{V_k}>0}.$$

In case (b), we generate S from Normal distributions first and generate V as a function of S.

$$\tilde{S_j} \sim \mathcal{N}(0,1), \tilde{V_k} = \tilde{S}_k + \tilde{S}_{k+1} + \mathcal{N}(0,2), \quad S_j = \mathbf{1}_{\tilde{S_j}>0}, V_k = \mathbf{1}_{\tilde{V_k}>0}.$$

In case (c), we generate V first and simulate S as a function of V.

$$\tilde{V_k} \sim \mathcal{N}(0,1), \tilde{S_j} = \tilde{V_j} + \tilde{V}_{j+1} + \mathcal{N}(0,2), \quad S_j = \mathbf{1}_{\tilde{S_j}>0}, V_k = \mathbf{1}_{\tilde{V_k}>0}.$$

For the outcome, we keep the generating procedure same across three cases. The binary outcome y is generated solely as a function of S,

$$\tilde{y} = \text{sigmoid}(\sum_{j=1}^{p_s} \alpha_j S_j + \sum_{j=1}^{p_s-1} \beta_j S_j S_{j+1}) + \mathcal{N}(0, 0.2), \quad y = \mathbf{1}_{\tilde{y}>0.5},$$

where $\text{sigmoid}(x) = 1/(1 + \exp(-x))$. This specification includes both the linear and non-linear effect of S. The parameters take values as $\alpha_j = (-1)^j (j\%3 + 1) * p/3, \beta_j = p/2$.

In addition to three different generating mechanisms, we introduce an additional spurious correlation with biased sample selection. Specifically, we set an inclusion rate $r = (0, 1)$ to create a spurious correlation between y and V. If the average value of $\bar{V}_i = \sum_{j=1}^{p_v} V_{ij}$ and $\tilde{y}_i$ exceed or fall below 0.5 together, we include sample i with probability r. Otherwise, we include the sample with probability $1 - r$. Namely, if $r > 0.5$, V and y share positive correlation and the correlation is negative if $r < 0.5$. The parameter r controls the degree of spurious correlation which induces the covariate shifts.

In the experiments, we generate a small amount of **R-DATA** following case (a) with size $n_r = 1000$, a large amount of **L-DATA** following case (b) $n_l = 5000$ and the testing data from case (c) with size $n_t = 2000$ to mimic the policy change on testing data. Additionally, we set $r = 0.7$ on the **L-DATA** and let r vary from 0.1 to 0.9 on the testing data to create additional deviance in the distribution. We also vary the number of features in total $p \in [20, 40, 80]$ and keep $p_s = 0.4p$. Within each configuration, we perform the experiments 200 times and calculate the average AUC and cumulative bias.

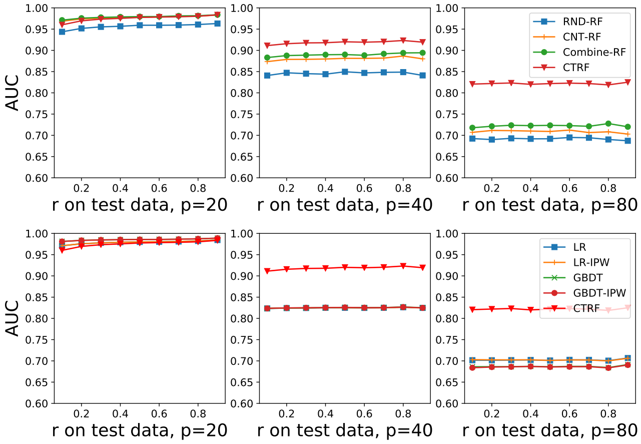


Figure 6: AUC comparison when $p = 20, 40, 80$. The top row compares with random forest based method and the bottom row compares other baselines. CTRF produces largest AUC in most cases.

Figure 6 shows the comparison of AUC against the variation on both p and r. The top row demonstrates the comparison within the domain of random forest. The CTRF (red lines) performs the best regardless of feature dimensions. Also, the advantage of CTRF over other methods increases as feature size goes up, which also boosts the chance of learning the model based on the features with spurious correlation. The second row in Figure 6 shows the comparison with LR, LR-IPW, GBDT and GBDT-IPW. Although the performances are indistinguishable when $p = 20$, the advantage of CTRF emerges as we have more factors with spurious correlation.

Figure 7 shows the comparison in terms of the bias. A lower value represents a better performance. The top row shows the comparison with other random forest based methods. Generally, the cumulative bias increases as r on the testing data decreases, which means the testing data deviates more from the **L-DATA**. However, the advantage of CTRF (red lines) increases slightly as r decreases, which demonstrates the robustness against covarites shifts. The comparison with LR or GBDT based methods at the bottom row shows a similar trend with the AUC. The CTRF achieves a lower bias among all the approaches and its advantage increases as we have more features.

4.3 Synthetic Auction: Implicit Mechanism

In this subsection, we setup a synthetic auction scenario with a single tuning parameter in the policy, demonstrating both how simple parameters can introduce bias into a domain and CTRF's ability to transfer between them. We first generate synthetic samples of classification data, or a mapping from features to a true relevant/irrelevant binary label. From this data, we build a true relevance model with random forest to estimate the probability an item is relevant. Second, we build our **L-DATA** and testing auctions by sampling (20 per auction) from the underlying relevance features and assigning a relevance score. Per auction, the items are thresholded with the corresponding *relevance reserve* parameter and the remaining items are ranked. This provides layout and position information, in addition to the relevance score and relevance features. Third, Given the layout and items, a simulated user chooses a single ad as relevant uniformly at random to click, and leaves the others not clicked. The choice of click is uniform across positions, which means that *position* is purely a factor spuriously correlated with the relevance while not affecting the click. We provide the detailed generating mechanism in the Supplementary Material.

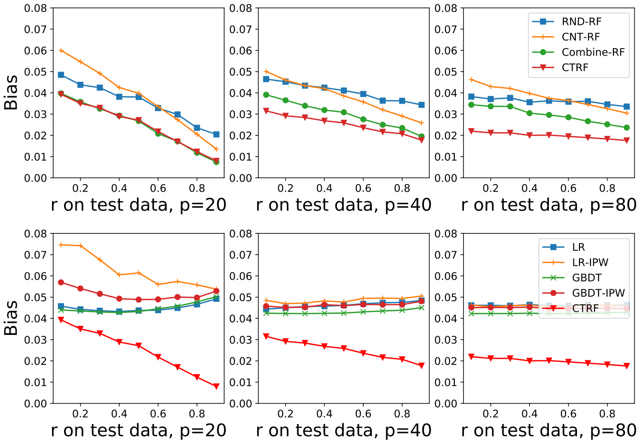


Figure 7: Bias comparison when $p = 20, 40, 80$, with top row comparing with random forest based method and bottom row comparing other baselines. CTRF achieves the lowest bias in all cases.

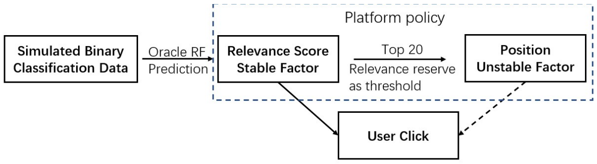

Platform policy
Simulated Binary Classification Data
Oracle RF Prediction
Relevance Score Stable Factor
Top 20
Relevance reserve as threshold
Position Unstable Factor
User Click

Figure 9: Procedures for simulating auctions. Position is an unstable factor for predicting click as the users pick ads uniformly on a page to click and its correlation with relevance score varies across policy, which is implicitly determined by the relevance reserve parameter.

The tuning parameter in the experiment is the *relevance reserve* parameter r, controlling the requirement that any item shown to a user meet a minimum relevance, which controls $p(z|x)$ implicitly. The mechanism to generate simulated auction is illustrated in Figure 9. This parameter affects the correlation between relevance and position, which can vary between **L-DATA** and testing data. Specifically, we generate the **L-DATA** with relevance reserve parameter $r = 0.5$ while the testing data with the relevance reserve varying in $r \in [0.5, 0.9]$, simulating a desire to increase the quality of items presented to a user (with a higher threshold). A larger value in

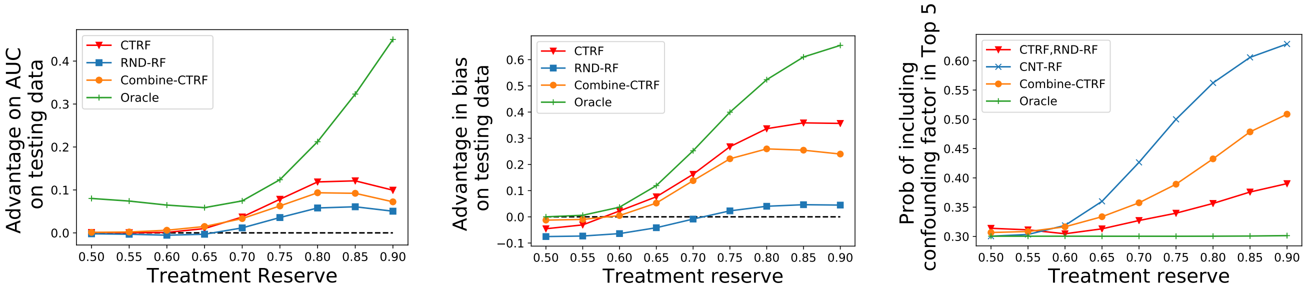


Figure 8: AUC (left graph), cumulative prediction bias (middle graph) and probability of including confounding factor "position" as Top 5 important features (right graph) versus treatment reserve r. Higher r represents a larger change in the testing distribution. CTRF performs the best among all random forest methods.

$r > 0.5$ represents a higher deviation from the **L-DATA**with $r = 0.5$. For the **R-DATA**, we do not have the auction procedure and we pick up the advertisement uniformly random to display on the page. The size of **R-DATA** is approximately 20% of the **L-DATA**.

As we use the random forest model to generate the true relevance score, we compare the CTRF within the domain of random forest based methods only, including CNT-RF, RND-RF, Combine-RF and the oracle one training RF on the testing data. Figure 8 illustrates prediction performance of all method while setting CNT-RF as the baseline. To illustrate the advantage over the baseline method, CNT-RF, we minus the AUC of CNT-RF from that of all other methods and minus the bias of the corresponding model from the bias of CNT-RF. Therefore, a larger value in the graphs indicates a better performance of the corresponding method.

In Figure 8, we observe that when the reserve for testing data lies close to 0.5, all models show similar performance. However, as we increase r on testing data and raise the degree of covariate shift, the CTRF method (red lines) greatly improves in both AUC and bias. Also, the CTRF demonstrates a better prediction power and lower bias compared with the RND-RF and Combine-RF. This illustrates CTRF's ability to transfer knowledge from one domain to a similar but distinct domain with unstable factor (in this case, an ad's position).

We calculate the probability of including the "position", which is a known spuriously correlated factor by design, in the top 5 factors ranking by feature importance [11] evaluated on the training dataset. As shown in the right panel of Figure 8, the random forest learned on the **R-DATA** (RND-RF,CTRF are identical) has a lower probability of identifying the unstable or confounding factor as important predictors, compared with the one utilizing the **L-DATA** (CNT-RF, Combine-RF). This demonstrates that the first stage of structure learning or the decision boundary on **R-DATA** can reduce spurious correlation. This also validates utilizing the large amount of the **L-DATA** to calibrate the parameters in the structure or trees in the second stage as the prediction does not rely on the unstable factor.

5 EXPERIMENTS ON REAL-WORLD DATA

In this section, we present experimental results in the real-world application with data collected from a sponsored search search platform (Bing Ads). First, we discuss how **R-DATA** is collected from real traffic. Next, we demonstrate the robustness of CTRF-trained click models against the distribution shifts. Finally, we show that CTRF-enabled holistic counterfactual policy estimation improves global marketplace optimization problem real business scenarios.

5.1 Randomized Experiment (R-DATA)

Randomized data (**R-DATA**) collection is very important step to create CTRF since training requires **R-DATA** to learn the structure of trees. In order to collect **R-DATA**, we used existing randomization policy on paid search engine which is triggered less than %1 of the live traffic. The existing randomization policy is triggered in typical sponsored search requests and there is no difference between randomized and mainstream traffic in terms of user and advertiser selection. For a given paid search request, if randomization is enabled, special uniform randomization policy is triggered. In this uniform randomization policy, all choices that depend on models are completely randomized. In particular, the ads are randomly permuted and the page layout (where ads are shown on the page) is chosen at random from the feasible layouts. The user cost (due to lower relevance) of such randomization is very high and consequently, limits the trigger rate for the randomized policy.

5.2 Robustness to Real-World Data Shifts

We train the user click model on the data collected from the mainstream traffic and randomized traffic in the search engine, corresponding to the **L-DATA** and **R-DATA** respectively. We validate the proposed method on an exploration traffic with some radical experiments (layout template change, for example), which is the testing data with covariate shifts. We only compare the method with CNT-RF, Combine-RF and Oracle-RF, which trains a random forest on the testing data. The last one cannot be implemented in practice yet it serves to illustrate the capacity of the random forest method. We fix the total training size to be one million with each method [1] for a fair comparison. We focus on three metrics of interests: AUC (area under curve), RIG (Relative Information Gain) and cumulative prediction bias[2].

Table 1 shows that CTRF achieves the best performance among all the random forest candidates. Also, the AUC of CTRF is very close to Oracle-RF, which indicates its nearly-optimal prediction

[1] The ratio of **R-DATA** and **L-DATA** is about 1:7, after down-sampling on the **L-DATA**

[2] Relative information gain is defined as the RIG $= (H(\bar{y}) + L)/H(\bar{y})$, L is the log loss produced by the model and $H(p) = -p\log(p) - (1-p)\log(1-p)$ is the entropy function. Higher value indicates better performance.

Table 1: Performance comparison for different random forest based model, evaluated on some exploration flights with radical policy changes

Methods	AUC	RIG	Cumulative Bias
CNT-RF	0.9273	0.4424	3.87%
Combine-RF	0.9282	0.4460	3.39%
CTRF	**0.9285**	**0.4477**	**2.90**%
Oracle-RF	0.9287	0.4484	0.58%

performance. Although there still remains a gap with the Oracle-RF, the CTRF reduces the bias for click rate prediction to a non-negligible degree, which is very essential to the publishers in decision making. As we are evaluating all the performance on a part of the traffic performing some radical changes, the results demonstrate that the CTRF improves the robustness of user click model in terms of prediction power.

5.3 End-to-end Marketplace Optimization

In addition to the prediction power of the model, we also evaluate how the usage of CTRF can advance the decision making procedure in real business optimization at Bing Ads.

5.3.1 Marketplace Optimization in a Nutshell. The goal of Marketplace Optimization for sponsored search is to find optimal operating points for each component of the search engine given all marketplace constraints. Marketplace optimization is very different from optimizing certain objective functions with a given machine learning model. While model training focuses on reducing prediction error for unobserved data, Ads Marketplace Optimization focuses on improving global objectives like total clicks, revenue when new machine learning model is used as part of a bigger system. Due to data distribution shifts between components of a larger system, a locally optimized click model does not necessarily give best performances for global metrics. Therefore, whole components of the system may need to be tuned together by using more holistic approaches like A/B testing or similar.

5.3.2 Experimental Data Selection and Simulation Setup. Robust click prediction plays a very crucial role in improving holistic Ads Marketplace Optimizer like an open box simulator [3] which can easily have biased estimations due to data distribution shifts in counterfactual environments. In our problem context, we integrate CTRF to an open box offline simulator and show that a new simulator with CTRF will give better results for offline policy estimation scenarios when data distribution shift is significant.

For experimental runs, we use an open box simulator with two versions of random forest, CTRF and CNT-RF (typical RF used), along with the generalized linear Probit model for click prediction. Then, we run offline counterfactual policy estimation jobs with modified inputs over logs collected from real traffic. Finally, we compare predictions for marketplace level click metrics with different models against A/B testing by using same production data that is collected from A/B testing experiment.

To select experimental data, we checked the counterfactual vs factual feature distribution similarity of multiple scenarios in search engine traffic. We applied Jensen-Shannon (JS) divergence to compute the similarity of two distributions. Based on this metric, we selected two tuning use cases which has significantly higher distribution shift. First use case belongs to capacity change for Text Ads blocks. Second use belongs to page layout change. Details on this procedure are included in the Supplementary Material.

5.3.3 Experiments on Real Case Studies. In the first case, the capacity of the particular ad block that contains Textual Ads was increased on the traffic in May 2019 for 10 days time period during A/B testing. The change was expected to increase both overall click yield and click yield on textual ad slice for target ad block. For simulator runs, we used 4.6 million samples from control traffic (**L-DATA**) and 100K samples from the randomized traffic (**R-DATA**) that belongs to 3 weeks time period before end date of A/B testing. The randomized traffic corresponds to page view requests where the mechanisms in online system are randomized, as described in Section 5.1.

In simulator runs with CTRF, we train the forest and tree structures from **R-DATA** and combine the **L-DATA** and **R-DATA** to calibrate the leaves of trees in the forest. Each simulation job uses its trained model to score counterfactual page views that generated from replying control traffic logs in open box manner with the suggested input modification (capacity change of ad block). Table 2 presents the comparison of an open box simulator with generalized Probit model, with CTRF and the random forest trained on control traffic (CNT-RF) based on relative Click Yield delta error [3] against A/B testing experiment that was active for 10 days in May 2019. To make a fair comparison, we use the same amount of training data for different variants of random forest models. We observe that click yield deltas coming from simulator results with CTRF is significantly better than other approaches since results from CTRF enabled simulator are closer to A/B Testing results from real traffic.

Table 2: Performance comparison in two scenario with radical changes

Ad capacity change	ΔCY Error	ΔCY Error (Text Ads)
Probit Model	34.94%	17.13%
CNT-RF	12.11%	9.96%
CTRF	**2.07%**	**8.76%**
Layout change	ΔCY Error	ΔCY Error (Shopping Ads)
Probit Model	35.48%	45.08%
CNT-RF	58.06%	34.92%
CTRF	**22.58%**	**13.38%**

In the second scenario, the layout of product shopping ads was significantly updated in May 2019 for a week time period during A/B testing. The change was expected to increase both overall click yield and click yield on product shopping ads slice for target ad block. In this experiment, we used 15M samples from the control traffic in A/B testing and the same randomized traffic in the previous experiment. The bottom part in Table 2 presents the comparison

[3] Relative Click Yield delta error is defined as $|\Delta CY_{\text{Method}} - \Delta CY_{\text{AB}}|/|\Delta CY_{\text{AB}}|$. $\Delta CY_{\text{Method}}$ is the predicted change in click rate by the model. ΔCY_{AB} is the actual change in A/B testing.

of different model-based simulators in the relative error against the A/B testing experiment that was active for a week in May 2019. Since the modification for the second experiment yielded a radical shift in feature distribution of product shopping Ads. The difference with CTRF enabled simulator vs other approaches is more prominent. Thus, open box simulator with CTRF also outperforms other approaches in this scenario.

6 DISCUSSION AND CONCLUSIONS

We present a novel method, causal transfer random forest, to combine limited randomized data (**R-DATA**) and large scale logged data (**L-DATA**) in the learning problem. We propose to learn the tree structure or the decision boundary with the **R-DATA** and calibrate the leaf value of each tree with the whole data (**R-DATA** and **L-DATA**). This approach overcomes the spurious correlation in **L-DATA** and the limitations on sample size for the **R-DATA** to provide robustness against covariate shifts. We evaluate the proposed model in the extensive synthetic data experiments and implement it in Bing ads system to train the user click model. The empirical results demonstrate its advantage over other baselines against the radical policy changes and robustness in real-world prediction tasks. For future work, there are some important research questions to explore, such as a better understanding of the relative importance of the **R-DATA** versus the **L-DATA**, how much **R-DATA** is needed and how this quantity related to the degree of distributional shift.

REFERENCES

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

[2] Susan Athey and Guido Imbens. 2016. Recursive partitioning for heterogeneous causal effects. *Proceedings of the National Academy of Sciences* 113, 27 (2016), 7353–7360.

[3] Murat Ali Bayir, Mingsen Xu, Yaojia Zhu, and Yifan Shi. 2019. 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*. ACM, 465–473.

[4] Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. 2007. Analysis of representations for domain adaptation. In *Advances in neural information processing systems*. 137–144.

[5] Steffen Bickel, Michael Brückner, and Tobias Scheffer. 2009. Discriminative learning under covariate shift. *Journal of Machine Learning Research* 10, Sep (2009), 2137–2155.

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

[7] Leo Breiman. 2001. Random forests. *Machine learning* 45, 1 (2001), 5–32.

[8] Thomas D Cook, Donald Thomas Campbell, and William Shadish. 2002. *Experimental and quasi-experimental designs for generalized causal inference.* Houghton Mifflin Boston, MA.

[9] Hal Daume III and Daniel Marcu. 2006. Domain adaptation for statistical classifiers. *Journal of artificial Intelligence research* 26 (2006), 101–126.

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

[11] Robin Genuer, Jean-Michel Poggi, and Christine Tuleau-Malot. 2010. Variable selection using random forests. *Pattern recognition letters* 31, 14 (2010), 2225–2236.

[12] Trevor Hastie, Robert Tibshirani, Jerome Friedman, and James Franklin. 2005. The elements of statistical learning: data mining, inference and prediction. *The Mathematical Intelligencer* 27, 2 (2005), 83–85.

[13] Biwei Huang, Kun Zhang, Jiji Zhang, Joseph Ramsey, Ruben Sanchez-Romero, Clark Glymour, and Bernhard Schölkopf. 2020. Causal discovery from heterogeneous/nonstationary data. *Journal of Machine Learning Research* 21, 89 (2020), 1–53.

[14] Jiayuan Huang, Arthur Gretton, Karsten Borgwardt, Bernhard Schölkopf, and Alex J Smola. 2007. Correcting sample selection bias by unlabeled data. In *Advances in neural information processing systems*. 601–608.

[15] Guido W Imbens and Donald B Rubin. 2015. *Causal inference in statistics, social, and biomedical sciences.* Cambridge University Press.

[16] Nathan Kallus, Aahlad Manas Puli, and Uri Shalit. 2018. Removing hidden confounding by experimental grounding. In *Advances in Neural Information Processing Systems*. 10888–10897.

[17] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems*. 3146–3154.

[18] Kun Kuang, Peng Cui, Susan Athey, Ruoxuan Xiong, and Bo Li. 2018. Stable prediction across unknown environments. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1617–1626.

[19] David Lazer, Ryan Kennedy, Gary King, and Alessandro Vespignani. 2014. The parable of Google Flu: traps in big data analysis. *Science* 343, 6176 (2014), 1203–1205.

[20] Lihong Li, Wei Chu, John Langford, Taesup Moon, and Xuanhui Wang. 2012. An unbiased offline evaluation of contextual bandit algorithms with generalized linear models. In *Proceedings of the Workshop on On-line Trading of Exploration and Exploitation 2*. 19–36.

[21] Sara Magliacane, Thijs van Ommen, Tom Claassen, Stephan Bongers, Philip Versteeg, and Joris M Mooij. 2018. Domain adaptation by using causal inference to predict invariant conditional distributions. In *Advances in Neural Information Processing Systems*. 10846–10856.

[22] Nicolai Meinshausen. 2018. Causality from a distributional robustness point of view. In *2018 IEEE Data Science Workshop (DSW)*. IEEE, 6–10.

[23] Scott Menard. 2002. *Applied logistic regression analysis*. Vol. 106. Sage.

[24] Radford M Neal. 2001. Annealed importance sampling. *Statistics and Computing* 11, 2 (2001), 125–139.

[25] Judea Pearl. 2009. *Causality*. Cambridge university press.

[26] Judea Pearl et al. 2009. Causal inference in statistics: An overview. *Statistics Surveys* 3 (2009), 96–146.

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

[28] Joaquin Quionero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence. 2009. *Dataset shift in machine learning*. The MIT Press.

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

[30] Hidetoshi Shimodaira. 2000. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference* 90, 2 (2000), 227–244.

[31] Hal R Varian. 2007. Position auctions. *international Journal of industrial Organization* 25, 6 (2007), 1163–1178.

[32] Stefan Wager and Susan Athey. 2018. Estimation and inference of heterogeneous treatment effects using random forests. *J. Amer. Statist. Assoc.* 113, 523 (2018), 1228–1242.

[33] Mei Wang and Weihong Deng. 2018. Deep visual domain adaptation: A survey. *Neurocomputing* 312 (2018), 135–153.

[34] Ya Xu, Nanyu Chen, Addrian Fernandez, Omar Sinno, and Anmol Bhasin. 2015. From infrastructure to culture: A/b testing challenges in large scale social networks. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2227–2236.

[35] Kun Zhang, Bernhard Schölkopf, Krikamol Muandet, and Zhikun Wang. 2013. Domain adaptation under target and conditional shift. In *International Conference on Machine Learning*. 819–827.