# Towards Robust Off-Policy Learning for Runtime Uncertainty

**Da Xu,**[1] **Yuting Ye,** [2] **Chuanwei Ruan,** [3] **Bo Yang** [4]

[1] Walmart Labs
[2] University of California, Berkeley
[3] Instacart
[4] LinkedIn
daxu5180@gmail.com, yeyt@berkeley.edu, ruanchuanwei@gmail.com, by9ru@virginia.edu

## Abstract

Off-policy learning plays a pivotal role in optimizing and evaluating policies prior to the online deployment. However, during the real-time serving, we observe varieties of interventions and constraints that cause inconsistency between the online and offline settings, which we summarize and term as runtime uncertainty. Such uncertainty cannot be learned from the logged data due to its abnormality and rareness nature. To assert a certain level of robustness, we perturb the off-policy estimators along an adversarial direction in view of the runtime uncertainty. It allows the resulting estimators to be robust not only to observed but also unexpected runtime uncertainties. Leveraging this idea, we bring runtime-uncertainty robustness to three major off-policy learning methods: the inverse propensity score method, reward-model method, and doubly robust method. We theoretically justify the robustness of our methods to runtime uncertainty, and demonstrate their effectiveness using both the simulation and the real-world online experiments.

## Introduction

The offline optimization and evaluation have been studied intensively in the past few years, as deploying a sub-optimal policy for real-time experiments can be costly and even risky (Dudík, Langford, and Li 2011; Bottou et al. 2013; Swaminathan and Joachims 2015a; Athey and Wager 2017). Nevertheless, it is arguably true that off-policy learning can barely represent real-world scenarios except for a few ideal settings. In particular, real-time policy deployments are inevitably subject to various interventions and constraints that can *not* be unaccounted for in standard off-policy learning. We categorize these unexpected events as *runtime uncertainty* since they are brought about by some anomalous events of the online execution mechanism. Examples include:

1. in e-commerce recommendation, the product displacement is sometimes subject to real-time business factors such as trending popularity;
2. in personalized online advertisement, during peak hours, the policy execution may be replaced by the backup plan if the response-time agreement can not be satisfied;
3. in autonomous driving, the self-driving vehicle can be hit by accidents, harsh weather, or intrusions that limit the ability to carry out the designed policy.

We conclude that the impacts of runtime uncertainty are often reflected in the changes to the policy's execution. In the above examples, one may expect the following consequences: 1). the original exposure probabilities are adjusted by the instant trends of the items; 2). the exceeding web traffic will be directed to the rollback setting, e.g., non-personalized policy, rather than calling the online inference service of the designed policy; 3). the emergency mechanism, e.g., emergent brake, is triggered and takes over the running policy.

In stark contrast to the issue of a non-stationary environment where the underlying reward mechanism changes over time (Bubeck and Slivkins 2012), runtime uncertainty may not alter the reward mechanism. Additionally, runtime uncertainty does not cause distribution change or missing observation of the contextual features, so our problem clearly distinguishes from the setting of distribution-robust and confounding-robust policy optimization (see Section  for detail). The major challenge here is that the designed policy cannot be executed as-is due to unexpected events on the executors, which leads to discrepancies between the online and offline settings. It is implausible to remove or characterize these exceptional uncertainties by learning from the logged data, since they are abnormal and intractable by nature. As a consequence, the runtime uncertainty raises a novel problem for off-policy learning, which to the best of our knowledge, has not been studied.

Our strategy is to incorporate a certain level of robustness against potential runtime uncertainty via an offline max-min learning. We search for an adversarial scenario where runtime uncertainties induce the worst impact on a value function of interest. Since runtime uncertainties act on policy's execution, we adversarially perturb the logging policy in search for the worst case as if additional runtime uncertainties have played a role. Then we identify the candidate policy with the best offline performance under this worst case. Specifically, for any candidate policy $\pi$, let $\hat{V}(\pi, \pi_u)$ be the estimated reward, e.g., via inverse-propensity score weighting, assuming that the logged feedback was generated under $\pi_u$ ($u$ represents some perturbation mechanism). The key is to introduce an uncertainty set $\mathcal{U}_\alpha(\pi_0)$ that surrounds the logging policy $\pi_0$, constructed by such as the $\ell_2$, $\ell_\infty$ or Wasserstein balls with radius $\alpha$. To find the optimal policy that is robust to any possible data-generating mechanism in $\mathcal{U}_\alpha(\pi_0)$, we formulate adversarial learning objective as:

$$\underset{\pi}{\text{maximize}} \min_{\pi_u \in \mathcal{U}_\alpha(\pi_0)} \hat{V}(\pi, \pi_u),$$

where $\pi_u$ plays the role of the uncertainty-perturbed policy as if it generated the logged feedback. We can also evaluate the learned policy in such an adversarial scenario. When $\alpha = 0$, we revert to the standard off-policy learning where the solution is only optimal under $\pi_0$ and can be sensitive to future online uncertainties. We will further elaborate on the objective and the uncertainty set in Section .

Based on this framework, we enhance the robustness of the most common offline estimation methods, i.e., the reward-model (RM) method (Q-learning) (Jin et al. 2018), the inverse propensity weighting (IPS) method (Horvitz and Thompson 1952) and the doubly robust (DR) method (Robins, Rotnitzky, and Zhao 1994). We provide a practical characterization of the impact of runtime uncertainty. We find it gives the most interpretable and tractable solutions to measure the point-wise deviation from the original design to the final execution, which corresponds to constructing $\mathcal{U}$ using the $\ell_\infty$ distance. Then, we study the bounds that reveal how the estimators may fluctuate if the same uncertainty were applied to the logging policy, i.e. the range of $\left[\min_{\pi_u \in \mathcal{U}(\pi_0)} \hat{V}(\pi, \pi_u), \max_{\pi_u \in \mathcal{U}(\pi_0)} \hat{V}(\pi, \pi_u)\right]$ for any given $\pi$. The bounds provide a scope of the possible performances, which in turn enables developing an efficient algorithm for the max-min training objective. Furthermore, we study the generalization behavior of the proposed algorithm and rigorously analyze how it leads to robust estimators.

We conduct comprehensive simulation studies to examine the effectiveness of the proposed approach. We also conduct real-world online testings on an e-commerce platform, where our approach compares favorably to standard offline learning. We conclude our contributions as follows.

- We study the novel problem of obtaining a runtime-robust policy in the context of off-policy learning.
- We design a max-min framework to devise a runtime-robust policy and propose the optimization procedures to bound the estimators in the adversarial setting.
- We propose an off-policy algorithm for robust optimization, and theoretically show the tradeoffs and guarantees.
- We rigorously examine the effectiveness of our approach via both simulation and real-world experiments.

## Background and Related Work

Different types of robust machine learning and optimization have been intensively studied in the literature. They provide powerful tools to tackle various problems affected by uncontrolled factors (Ben-Tal, El Ghaoui, and Nemirovski 2009; Bertsimas, Brown, and Caramanis 2011). To better understand the runtime uncertainty, which has not been investigated before, we compare it with other four kinds of robustness, i.e., non-stationary environment, unobserved confounder, noisy label & measurement error, as well as adversarial learning.

We use $f_0$ to denote a reference distribution of the covariates (or treatment) $X$, outcome (or reward) $Y$, and unknown factors $U$, whose interpretations will be made case-specific. We use $f_{\text{train}}$, $f_{\text{test}}$ and $\hat{f}_{\text{train}}$ to denote the training distribution, the testing distribution, and the learned distribution based on the training data, respectively. See Table **??** for the summarized comparisons.

**Non-stationary environment (mechanism change)**. In the context of off-policy learning, the uncertainties (or known as mechanism change) caused by non-stationary environment have been studied in the recent literature (Dudík et al. 2014; Kallus and Zhou 2018; Jagerman, Markov, and de Rijke 2019; Si et al. 2020; Bareinboim, Forney, and Pearl 2015; Xu et al. 2021). This field focuses on a problem where the conditional distribution for $Y|X$ under $f_{\text{test}}$ differs from that of the training distribution $f_{\text{train}} := f_0$. It is caused by changes in environments, e.g., customers' interests change over time. To obtain a policy that can work for $f_{\text{test}}$, the precondition is that there exists shared structures between $f_{\text{train}}$ and $f_{\text{test}}$.

**Unobserved Confounder**. When there exist unobserved confounders $U$ that affect both the response $Y$ and the treatment $X$, the true underlying distribution $f_0$ is a function of $X$, $U$ and $Y$, but only $f_{\text{train}}(X, Y)$ is observed. The policy developed ignoring these confounders can be sub-optimal and risky to deploy. Research on this topic attempts to develop a robust policy that does not malfunction in the worst case of $f_{\text{test}} = f_0$. In particular, the sensitivity analysis is often employed to characterize the robustness of inference outcome when the non-confounding causal assumption is violated to various extents, and confounding-robust techniques are proposed accordingly (Rosenbaum and Rubin 1983; Rosenbaum and Silber 2009; Ding and VanderWeele 2016; Zhao, Small, and Bhattacharya 2019; Kallus and Zhou 2018).

**Noisy Label & Measurement Error**. The observed response or reward $Y$ may well be contaminated during the collection of large datasets, which is referred to as noisy label learning (Natarajan et al. 2013; Northcutt, Jiang, and Chuang 2021; Zheng et al. 2020; Ghosh, Kumar, and Sastry 2017). In addition, the field of measurement error concerns the case where there exist some errors in measuring the covariates $X$, or $X$ is blurred by some systematic noise (Neumayer and Plümper 2017; Blackwell, Honaker, and King 2017; Ye and Bickel 2021). Most studies in this venue focus on recovering the true underlying distribution $f_0(X, Y)$ given the noisy observed training distribution $f_{\text{train}}(X, Y)$.

**Adversarial Learning**. Recent years have seen a flurry of studies on the development of algorithms for generating and learning from adversarial examples (Goodfellow, Shlens, and Szegedy 2015; Kurakin, Goodfellow, and Bengio 2016; Carlini and Wagner 2017; Madry et al. 2018). Such maliciously perturbed examples show little difference from original samples in human perception but can mislead machine learning models to incorrect decisions. In this setting, we still expect $f_{\text{train}}(X, Y) = f_0(X, Y)$, while $f_{\text{test}}(X, Y) \approx f_0(X, Y)$ since for the designed $X$ that may fool the algorithm, $f_{\text{test}}(X, Y) \neq f_0(X, Y)$ .

**Runtime Uncertainty**. The runtime uncertainty is caused by unexpected events during execution. It drives $f_{\text{train}}$ and $f_{\text{test}}$ away from the expected $f_0$ generated by the designed policy.

| | Data distribution | Goal | Source of uncertainty |
|---|---|---|---|
| Non-stationary environment | $f_{\text{train}}(Y\|X) = f_0(Y\|X)$<br>$f_{\text{test}}(Y\|X) \neq f_0(Y\|X)$ | Adapt $\hat{f}_{\text{train}}$ to $f_{\text{test}}$ | $f_{\text{train}}$ and $f_{\text{test}}$ only overlap to some extent with perhaps shared common structures. |
| Unobserved Confounder | $f_{\text{train}}(X, Y) \neq f_0(X, U, Y)$<br>$f_{\text{test}}(X, U, Y) = f_0(X, U, Y)$ | Develop a policy for $f_0(X, U, Y)$ | Unable to observe some confounders $U$. |
| Noisy Label & Measurement Error | $f_{\text{train}}(X, Y) \neq f_0(X, Y)$ | Recover the true $f_0$ | $X$ or $Y$ might be contaminated. |
| Adversarial Learning | $f_{\text{train}}(X, Y) = f_0(X, Y)$<br>$f_{\text{test}}(X, Y) \approx f_0(X, Y)$ | Make $\hat{f}_{\text{train}}$ robust to adversarial examples. | $f_{\text{test}}$ and $f_{\text{train}}$ are similar, but there are adversarially designed cases that drive the former away from the latter. |
| Runtime Uncertainty | $f_{\text{train}}(X, Y) \neq f_0(X, Y)$<br>$f_{\text{test}}(X, Y) \neq f_0(X, Y)$<br>$f_{\text{train}}(X, Y) \neq f_{\text{test}}(X, Y)$ | Make $\hat{f}_{\text{train}}$ robust to runtime uncertainties | $f_{\text{train}}$ and $f_{\text{test}}$ deviate from $f_0$ irregularly, so we aim to develop a policy that is robust to future uncertainty in $f_{\text{test}}$. |

Table 1: A brief comparison of the runtime uncertainty to other four types of robustness learning.

The changes in distributions are irregular and unstructured, which suggests that: 1). $f_{\text{train}} \neq f_0$ and $f_{\text{test}} \neq f_0$; 2). the deviations are unsystematic and cannot be learned from data. In spite of certain similarities, the properties of the runtime perturbations fundamentally differ from the above scenarios:

- the distribution-robust methods only assume the perturbations of the contextual features $X$ (Si et al. 2020); the perturbations in the noisy label & measurement error occur marginally to the response $Y$ or the covariates $X$; the adversarial learning increases the likelihood of seeing the problematic covariates $(X, Y)$ that the machine can make mistakes on. On the other hand, the runtime perturbations allow the entire policy to be perturbed;
- by the definition of confounders, most confounding-robust methods assumes perturbations to be independent of the actions that were taken, e.g., $\pi(a|x, u) \propto \pi_0(a|x) + \lambda u$ (Kallus and Zhou 2018); however, the nature of runtime uncertainty is much more complex since its effect can be dependent on the actions, so the perturbation should be able to adapt to any given policy.

Another line of research studies the estimator instability caused by the variance issues of specific off-policy estimators (Swaminathan and Joachims 2015b; Ma, Wang, and Narayanaswamy 2019; Farajtabar, Chow, and Ghavamzadeh 2018; Xie, Ma, and Wang 2019; Thomas and Brunskill 2016; Vlassis et al. 2019). Their methods for improving the estimators' stability can be adapted to our solutions.

## Preliminary

In this section, we introduce the notations, basic concepts, problem setup as well as the off-policy estimators of interest.

**Notation**. Let $\mathcal{X}$ be the context (feature) space, $\mathcal{A} = \{1, \dots, k\}$ be the action space, and $r(a, x)$ be the fixed *value* (reward or regret) that is revealed under action $a \in \mathcal{A}$ and context $x \in \mathcal{X}$. The challenge of offline learning and evaluation is largely due to the partial-observation of the complete reward — we do not know the rewards of untaken actions. For an individual with context $x^{(i)}$ and provided with action $a^{(i)} \in \mathcal{A}$ at the $i^{\text{th}}$ round, the feedback data collected after $T$ rounds is given by the set of triplets $h_T = \big\{(x^{(i)}, a^{(i)}, r(a^{(i)}, x^{(i)}))\big\}_{i=1}^{T}$. The *logging policy*, which gives the conditional probability of an action under particular context and history, is denoted by: $\pi_0\big(a|x^{(T+1)}, h_T\big)$. Since our primary focus is runtime uncertainty, we assume the *logging* policy is *stationary* and the contexts are static such that $\pi_0\big(a|x^{(T+1)}, h_T\big) = \pi_0\big(a|x^{(T+1)}\big)$.

**Value function**. For off-policy learning, given the context $x$ and the complete reward $r(a, x)$ for all actions, the *value* of a policy is: $V(\pi) = \mathbb{E}_{\pi(a|x)}[r(a, x)]$. Policy evaluation estimates the value $\hat{V}(\pi)$ using the collected feedback data $\{(x_i, a_i, r_i)\}_{i=1}^{n}$ and the logging policy $\pi_0(a_i|x_i)$ if available. We aim at *reward-maximization* unless otherwise specified. Policy optimization address the problem of finding the optimal policy from a parametric family of candidate policies $\mathcal{F}$, for instance, $\pi^* = \arg\max_{\pi \in \mathcal{F}} \hat{V}(\pi)$.

Estimating the value of an alternative policy $\pi$ is challenging because we do not observe the *potential value* (defined as below) for the actions that are not selected.

**Definition 1.** A *potential value* $R(a, x)$ is the *value* that would have been observed if the individual with context $x$ received action $a$.

By definition, the reward $r(a, x)$ is an averaged version of $R(a, x)$ that integrates out all the possible actions under $\pi_0$:

$$\begin{aligned} r(a, x) &:= \mathbb{E}_{\pi_0}\big[R(a, x)\big|X = x\big] \\ &= \sum_{\tilde{a} \in \mathcal{A}} \mathbb{E}\big[R(a, x) \,\big|\, A = \tilde{a}, X = x\big]\pi_0(\tilde{a}|x), \end{aligned} \quad (1)$$

which takes the form of the *averaged potential value (APV)*. In view of Pearl's framework (Pearl and Mackenzie 2018), $R(a, x)$ is a rung-3 quantity and $r(a, x)$ is a rung-2 quantity that averages out the counterfactual effect. This viewpoint has a profound implication for offline evaluation. For example, an item is tagged with a price $a$ while the true price suggested by the market is $\tilde{a}$. When $\tilde{a}$ is larger than $a$, the quantity $\mathbb{E}\big[R(a, x) \,\big|\, A = \tilde{a}, X = x\big]$ reveals how the customer may react to the lowered price. In theory, averaging all possible actions $\tilde{a}$ forms a comprehensive and systematic evaluation for any specified action $a$ and context $x$.

**Offline estimators**. It is unrealistic to observe $(r(a), \tilde{a}, x_i)$ for $a \neq \tilde{a}$ due to the partial-observation nature. A huge body of literature has been devoted to estimating the averaged potential value $\mathbb{E}_{\pi_0}\big[R(a,x)\big|X=x\big]$, such as the potential outcome models (Robins, Rotnitzky, and Zhao 1994), structural models (Robins, Hernán, and Brumback 2000; Pearl 2009), two-stage regression models (Angrist and Imbens 1995). Broadly speaking, they attempt to model the reward mechanism, and is also referred to as the Q-learning in reinforcement learning (Sutton and Barto 2018). Without loss of generality, we refer to them as the *reward-model (RM)* method denoted by $\hat{V}_{\text{RM}}$.

$$\hat{V}_{\text{RM}}(\pi;\pi_0) = \frac{1}{n}\sum_{i=1}^{n}\sum_{a\in\mathcal{A}}\pi(a|x_i)\hat{r}(a,x_i), \tag{2}$$

where *hatr* is an estimate of Eq. (1). RM is often subject to model misspecifications that lead to biased estimation. Using the idea of importance sampling, the *inverse-propensity score method* (IPS) described below can estimate the policy value unbiasedly (Horvitz and Thompson 1952). By correcting for the shift in action probability between $\pi_0$ and $\pi$, IPS is less prone to the bias issues.

$$\hat{V}_{\text{IPS}}(\pi;\pi_0) = \frac{1}{n}\sum_{i=1}^{n}\frac{\pi(a_i|x_i)}{\pi_0(a_i|x_i)}r_i. \tag{3}$$

In practice, the variance-stabilized versions of IPS, e.g., the normalized and truncated IPS, are considered more often (Vlassis et al. 2019; Gilotte et al. 2018). We defer the discussion to the appendix to avoid unnecessary repetitions.

The *doubly robust estimator* (DR) is a popular *control variate* method that effectively leverages both RM and IPS (Robins, Rotnitzky, and Zhao 1994). The notion comes from the fact that DR gives consistent estimation when either RM or IPS is consistent. Here, $\hat{V}_{\text{DR}}$ is given by:

$$\hat{V}_{\text{DR}}(\pi;\pi_0) = \frac{1}{n}\sum_{i=1}^{n}\Big\{\hat{r}(x_i) + \frac{\pi(a_i|x_i)}{\pi_0(a_i|x_i)}\big(r_i - \hat{r}(a_i,x_i)\big)\Big\},$$

where $\hat{r}(x_i)$ is the RM estimation: $\sum_{a\in\mathcal{A}}\pi(a|x_i)\hat{r}(a,x_i)$.

## Offline Estimators under $\ell_\infty$ Uncertainty

Unlike other domains where $\mathcal{U}_\alpha$ can have many options, finding the right uncertainty set is critical to the interpretation and effectiveness of our approach. The reasons are:

1. The formulation of $\mathcal{U}_\alpha$ should conform to how online uncertainty arises in practice, which in turn explains the type of robustness we are asserting. For instance, the $\ell_2$ ball would suffice if we believe the impact of online uncertainty is close to average on each action.
2. The constraint optimization of $\min_{\pi_u\in\mathcal{U}_\alpha(\pi_0)}\hat{V}(\pi,\pi_u)$ can be intractable or extremely challenging to solve; for instance, when $\mathcal{U}_\alpha(\pi_0)$ is given by the Wasserstein ball[1].

In our study, we find the $\ell_\infty$ distance gives a nice trade-off in terms of the *interpretability*, *practicality*, and the *robustness guarantee*. We first discuss the interpretation of using $\ell_\infty$ distance. Recall from Section  that runtime uncertainty causes unexpected interventions and constraints to the policy's execution, which can impact $\pi(a|x)$ for *any* $a$ and $x$. It means runtime uncertainty can cause deviation that is best measured in a point-wise fashion, e.g. $\max_a\big|\pi_u(a|x)-\pi_0(a|x)\big|$ for a given $x$, rather than by some average, e.g. $\frac{1}{k}\sum_a\big|\pi_u(a|x)-\pi_0(a|x)\big|$. To better leverage the fraction format in IPS and DR, we replace the absolute difference by the ratio of: $\max\{\frac{\pi_u(a|x)}{\pi_0(a|x)},\frac{\pi_0(a|x)}{\pi_u(a|x)}\}$. If we treat the policies $\pi_0$ and $\pi_u$ as vectors, by the same essence, their deviation is exactly measured by the $\ell_\infty$ norm. Therefore, the uncertainty set $\mathcal{U}_\alpha(\pi_0)$ follows:

$$\Big\{\pi_u : \max_{a\in\mathcal{A},x\in\mathcal{X}}\max\Big\{\frac{\pi_u(a|x)}{\pi_0(a|x)},\frac{\pi_0(a|x)}{\pi_u(a|x)}\Big\}\leq e^{\alpha}\Big\}. \tag{4}$$

Here, the adversarialness endowed in $\pi_u$ can depend on the actions and contexts given the logged data, e.g. $\pi_u(a|x)\propto\pi_0(a|x)+u(a,x)$ where $u$ is some random function. It means the perturbation can be made *policy-specific*, as opposed to policy-agnostic case where the perturbation is uniform for all the actions (such as in confounding-robust optimization). It more closely resembles the mechanism of real-world online uncertainty. For the sake of notation, we also denote the constraints by the short hand: $e^{-\alpha}\leq\pi_u/\pi_0\leq e^{\alpha}$. We now establish the minimax objective for robust off-policy learning under the $\ell_\infty$ uncertainty set:

$$\underset{\pi}{\text{maximize}}\min_{\pi_u:\,e^{-\alpha}\leq\pi_0/\pi_u\leq e^{\alpha}}\hat{V}(\pi;\pi_u). \tag{5}$$

Note that the candidate policy $\pi$ is not involved in the constraints of the minimization step. Therefore, if we have a subroutine that efficiently computes for any candidate $\pi_\theta$:

$$\tilde{V}(\pi_\theta) := \min\hat{V}(\pi_\theta;\pi_u)\text{ s.t. }e^{-\alpha}\leq\pi_0/\pi_u\leq e^{\alpha}, \tag{6}$$

or provides the lower bound, i.e. $\underline{\hat{V}(\pi_\theta)}\leq\tilde{V}(\pi_\theta)$, we can divide and conquer the minimax objective. This technique resembles the well-known *expectation-maximization* and *minorize-maximization* algorithms (Lange 2016; Dempster, Laird, and Rubin 1977), whose implications are discussed in the appendix. In the sequel, we focus on deriving the bounds $\underline{\hat{V}(\pi_\theta)}$ for the three off-policy estimators of interest.

**The Reward-model Estimator**. To derive the constrained lower bound of $\hat{V}_{RM}$ defined in (2), it amounts to bounding the APV of (1) such that $\pi_u$ satisfies the constraint (4). We convert the constraint optimization into a subproblem of the standard ERM, as we show in Lemma 1. The technical details and proof are deferred to the appendix.

**Lemma 1.** *When $\pi_u$ satisfies* (4)*, then*

$$\begin{aligned}&\mathbb{E}_{\pi_u}\big[R(a,x)|A=a',X=x\big]\\ \geq\;&\min_{f_{a,a'}(\cdot)}\mathbb{E}\big[\ell_\alpha\big(R(a,x)\,,\,f_{a,a'}(x)\big)\,\big|\,A=a,X=x\big],\end{aligned}$$

*where the loss function $\ell_\alpha$ is specified by:*

$$\begin{aligned}\ell_\alpha\big(R(a,x)\,,\,f_{a,a'}(x))\big) = \big\{R(a,x)-f_{a,a'}(x)\big\}_+^2&\\ +\,e^{2\alpha}\big\{R(a,x)-f_{a,a'}(x)\big\}_-^2.&\end{aligned} \tag{7}$$

[1] ERM under Wasserstein's constraint are often converted to another min-max optimization (Lee and Raginsky 2018); which means our objective will become a max-min-max problem.

*Here, $\{\cdot\}_+$ and $\{\cdot\}_-$ are respectively the positive and negative parts. Since the potential value $R(a,x)$ is observed given $A=a$, the above setting describes a subproblem of the ERM.*

The result in Lemma 1 holds for all the alternative $a' \in \mathcal{A}$ with $a' \neq a$, so we only need to compute $f_a := f_{a,a'}$ under any $a'$. The lower bound $\underline{\hat{V}_{\text{RM}}(\pi_\theta)}$ is easily obtained with:

$$\sum_{a'\in\mathcal{A}} \mathbb{E}\big[R(a,x)\,\big|\,A=a',X=x\big]\pi_u(a'|x)$$
$$\geq \min_{\pi_u: e^{-\alpha}\leq \pi_0/\pi_u \leq e^{\alpha}} \big\{\big(1-\pi_u(a|x)\big)\hat{f}_a(x)+$$
$$\pi_u(a|x)\mathbb{E}\big[R(a,x)\,\big|\,A=a,x\big]\big\}$$

where $\hat{f}_a$ is the solution to the auxiliary ERM problem in (7), and $\mathbb{E}\big[R(a,x)\,\big|\,A=a,x\big]$ can be estimated as in the standard off-policy setting. Therefore, computing the lower bound for RM involves solving a standard and a subproblem of the ERM, which is computationally efficient. **Upper bound**: we point out that lower-bounding the DR estimator also requires the upper bound of RM. Using the same arguments, the upper bound can be obtained using a similar auxiliary ERM approach, by replacing the $e^{2\alpha}$ in (7) with $e^{-2\alpha}$. We denote the upper bounds by $\big\{\hat{g}_a\big\}_{a\in\mathcal{A}}$.

**The IPS Estimator**. Lower-bounding the IPS estimator is more straightforward according to our design. The lower-bounding objective in (6) directly becomes:

$$\begin{aligned}\text{minimize}\quad & \tfrac{1}{n}\textstyle\sum_{i=1}^n \frac{\pi(a_i|x_i)}{p(a_i|x_i)} r_i\\ \text{s.t.}\quad & e^{-\alpha}\pi_0(a_i|x_i) \leq p(a_i|x_i) \leq e^{\alpha}\pi_0(a_i|x_i),\\ & \textstyle\sum_{a\in\mathcal{A}} p(a|x_i) = 1,\ \text{for } i\in[n],\end{aligned}$$

where we use $p(\cdot|\cdot)$ to denote the optimization variables. The second set of constraints is necessary because it makes sure that the solution constitutes a valid policy. The optimization problem for IPS can be solved explicitly using a change of variable: $w(a_i,x_i) := 1/p(a_i|x_i)$. It follows that both the objective and constraints are convex in $w$. In practice, the IPS estimator can suffer from the variance issues, so the variants such as the normalized and truncated IPS are often used instead in practice (Vlassis et al. 2019; Gilotte et al. 2018). We defer the discussions for the variant methods to the appendix to avoid unnecessary repetitions.

**The DR Estimator**. Lower-bounding DR is a straightforward combination of what we have shown for RM and IPS:

$$\text{minimize}\quad \hat{V}_{\text{DR}}(\pi_\theta;p,r) := \frac{1}{n}\sum_{i=1}^n\Big\{\sum_{a\in\mathcal{A}}\pi(a|x_i)r(a_i,x_i)$$
$$+\frac{\pi(a_i|x_i)}{p(a_i|x_i)}\big(r_i - r(a_i,x_i)\big)\Big\}$$
$$\text{s.t. } r(a_i,x_i)\in\mathcal{R}(\pi_0,\alpha) \text{ and } p(a_i,x_i)\in\Pi(\pi_0,\alpha),$$

with the constraint sets given by:

$$\mathcal{R}(\pi_0,\alpha) := \big\{r(a_i,x_i) : \hat{f}_{a_i}(x_i) \leq r(a_i,x_i) \leq \hat{g}_{a_i}(x_i); i\in[n]\big\},$$
$$\Pi(\pi_0,\alpha) := \big\{p(a_i|x_i) : e^{-\alpha} \leq \frac{p(a_i|x_i)}{\pi_0(a_i|x_i)} \leq e^{\alpha};$$
$$\sum_{a\in\mathcal{A}} p(a|x_i) = 1; i\in[n]\big\}.$$

The first set of constraints use the RM bounds to characterize the uncertainty of the reward models, and both the $\hat{f}_a$ and $\hat{g}_a$ can be computed beforehand. Although the objective for DR is not jointly convex in $r(a,x)$ and $w(a,x) := 1/p(a|x)$, it is coordinate-wise convex (affine). Therefore, we can employ any off-the-shelf solver. Also, the objectives for the IPS and DR are separable for each $x_i$, so we can efficiently parallelize the computations for each observation.

## Learning Algorithm and Guarantee

The learning objective can be nonconcave-nonconvex in general, so we cannot switch the $\min$ and $\max$. Therefore, the lower-bounding methods from the previous section, which holds for *any* given $\pi_\theta$, play an essential role in finding the approximate solution. We illustrate how to plug in the lower bounds for DR as an example since it includes both the IPS and RM estimator as special cases (see Algorithm 1).

From the learning-theoretical perspective, it is important to understand how the proposed approach affects the generalization performance while asserting robustness against runtime uncertainty. In the following theorem, we characterize the generalization of policy improvement of the max-min solution for DR, given any $\alpha > 0$.

**Theorem 1.** *Suppose that for all $\alpha > 0$, there exists a constant $M_\alpha$ such that $\max_{a,x}|\hat{f}_a(x)| \leq M_\alpha$ and $\max_{a,x}|\hat{g}_a(x)| \leq M_\alpha$. Also, we assume $\pi_0(a|x_i)\in(q,1-q)$ for some $q>0$. Let $\bar{r} := \max_i |r_i|$, then for all $\pi_\theta\in\mathcal{F}$ and $\forall\delta>0$, with probability as least $1-\delta$:*

$$V(\pi_\theta) - V(\pi_0) \geq \underbrace{\min_{p\in\Pi, r\in\mathcal{R}} \hat{V}_{DR}(\pi_\theta;p,r) - \hat{V}(\pi_0)}_{I}$$
$$\underbrace{-6\big(\frac{q+1}{q}M_\alpha + \bar{r}\big)\sqrt{\frac{2\log\frac{3}{\delta}}{n}} - 2\max\big\{M_\alpha, \frac{\bar{r}}{q}\big\}\mathcal{R}_n(\mathcal{F})}_{II},$$

*where $\hat{V}(\pi_0)$ is the logging policy's value on the training data, and $V(\pi) = \mathbb{E}\hat{V}(\pi)$ is defined as in Section .*

The proof is relegated to the appendix. The significance of Theorem 1 is to reveal the two critical components that control generalization: *I*. the empirical policy improvement under the proposed minorize-maximization algorithm; *II*. the composite terms of the policy complexity and the degree of uncertainty reflected via $M_\alpha$ (see Appendix A for detail).

In particular, by enriching $\mathcal{F}$, we are more likely to make *I* positive on training data, but we then suffer from a larger negativity of *II*. This tradeoff is consistent with the standard generalization for supervised learning. More importantly, the magnitude of *II* also increases with $M_\alpha$. Notice that $M_\alpha$ is often non-decreasing in $\alpha$ as the RM bounds get looser. As a consequence, having $\alpha > 0$ further penalizes the model complexity and the slack term in *II*, so increasing $\alpha$ will encourage the algorithm to select the policy that achieves less empirical improvement but has smaller complexity. It explains from the theoretical perspective how our approach can lead to a policy that performs better under runtime uncertainty. To summarize, the result in Theorem 1 shows rigorously how and why introducing $\pi_u$ with $e^{-\alpha} \leq \pi_0/\pi_U \leq e^{\alpha}$ can improve the robustness of the learned policy.

**Algorithm 1:** Robust Off-policy Learning with DR

```
**Input**  : The uncertainty level $\alpha$, history logging
             policy $\pi_0$, feedback data.
**Initialize** $\theta^{\text{new}}$, $p^*$, $r^*$;
Compute the constraints sets $\mathcal{R}(\pi_0; \alpha)$ for RM by
  solving the ERM subproblem as in Lemma 1;
Compute the constraints sets $\Pi(\pi_0; \alpha)$;
**while** $\min_{p,r} \hat{V}_{DR}(\pi_{\theta^{new}}) \geq \min_{p,r} \hat{V}_{DR}(\pi_{\theta^{old}})$ **do**
    Let $\theta^{\text{old}} = \theta^{\text{new}}$, compute
      $\theta^{\text{new}} = \arg\max_{\theta} \hat{V}_{\text{DR}}(\pi_{\theta}; p^*, r^*)$ using suitable
      optimization method;
    Solve: $p^*, r^* = \arg\min_{p \in \Pi, r \in \mathcal{R}} \hat{V}_{\text{DR}}(\pi_{\theta^{\text{new}}}; p, r)$.
**end**
```

## Experiment and Result

We first conduct simulation experiments to examine the following questions:

**Q1:** how does the lower-bounding methods for RM, IPS and DR described in Section  perform under different $\alpha$?

**Q2:** does the proposed Algorithm 1 improve the robustness of the learned policy against runtime uncertainty?

We also show the real-world performance of our robust off-policy learning approach by deploying the trained policy to an online e-commerce platform, where runtime uncertainties are frequent, for personalized product recommendation. Due to the space limitation, we only show the key outcome in this paper and leave the complete numerical results in the appendix. All the results are obtained from ten repetitions.

**Simulation**. We adopt the classical setting that generates bandit feedback according to multiclass classification (Dudík, Langford, and Li 2011; Vlassis et al. 2019). In particular, a $k$-class classification task is turned into the $k$-arm contextual bandit problem. In the classification problem, the data $\{(x, c)\}$ for the classification task are i.i.d observations where $x \in \mathcal{X}$ is the context (feature) vector and $c \in \{1, \dots, k\}$ is the class label. Here, each data point $(x, c)$ is converted into a cost-sensitive classification sample $(x, r_1, \dots, r_k)$, where $r_a = I[a = c]$ is the 0-1 reward for predicting with label $a$.

We now describe the data-generating mechanism. The feedback data under a given policy $\pi$ is constructed as follows. For each classification instance, we sample the label $a$ with the probability $\pi(a|x)$, and reveal the corresponding reward $r_a$. We use the same benchmark datasets from the UCI repository as in (Dudík, Langford, and Li 2011; Vlassis et al. 2019), with the descriptions provided below. We design the logging policy as: $\pi_0(a|x) \propto \theta_a^{\intercal} x$ for all $a = 1, \dots, k$, where $\theta_a$ are sampled i.i.d from the standard multivariate normal distribution. We use $\pi_0$ for off-policy learning.

| Dataset | Ecoli | Glass | Letter | PenDigits | SatImage | Vehicle |
|---|---|---|---|---|---|---|
| #samples | 336 | 214 | 20000 | 10992 | 6435 | 846 |
| #classes | 8 | 6 | 26 | 10 | 6 | 4 |

**Adding Runtime uncertainty.** Since the real-world runtime uncertainty can depend on $a$ and $x$, given a policy $\pi$, we add noise to $\pi$ and obtain the uncertainty-injected policy from which the feedback data is actually generated:

$$\tilde{\pi}(a|x) := \frac{\pi(a|x) \cdot U_{a,x}(\alpha)}{\sum_{\tilde{a}} \pi(\tilde{a}|x) \cdot U_{\tilde{a},x}(\alpha)},$$

where $U_{a,x}(\alpha)$ is sampled from the truncated normal distribution with unit variance and mean $\gamma_a^{\intercal} x$, where $\gamma_a$ is also sampled from standard multivariate normal distributions. We set the truncation interval to be $\big[0, \exp(\alpha)\big]$. Then it is easy to check that $e^{-\alpha} \leq \tilde{\pi}(a|x)\big/\pi(a|x) \leq e^{\alpha}$ almost surely for all $a$ and $x$, which conforms to (4).

**Estimators & Experiment setting**. We experiment with the IPS, RM and DR estimators as described in Section . The model family of the RM estimator (and the RM part of DR), as well as the bounding functions $\hat{f}_a$ and $\hat{g}_a$, are given by the standard *Regression Tree*. The tuning and other implementation details are left in the appendix. We obtain the feedback data using the noise-injected $\tilde{\pi}_0$, do the train-validation-test split detailed in the appendix, and conduct off-policy estimation & learning with $\pi_0$. To answer **Q1**, we first compute the RM estimator together with its bounding functions $\hat{f}_a$ and $\hat{g}_a$ using the training & valuation data, and plot their values on the testing data (Figure 1). IPS does not require further training, so we directly report their values, as well as the corresponding (lower) bounds, on the testing data. We combine IPS and RM to obtain the results for DR. To answer **Q2**, we first conduct off-policy optimization using both the standard off-policy learning and the proposed minimax learning method, with DR as the estimator, to obtain the trained policies $\pi^*$. We then report their associated regrets on the testing data after adding the runtime uncertainty to $\pi^*$'s execution. We also study their robustness by measuring how much their values may fluctuate on the testing data, quantified by $\hat{V}_{\text{DR}}(\pi^*, \pi_0) - \underline{\hat{V}_{\text{DR}}}(\pi^*, \pi_0)$, computed on testing data.

**Simulation results and analysis.** We consider a wide range of uncertainty, i.e. $\alpha \in \{0.01, 0.2, 0.4, 0.6\}$. We first examine the proposed bounding methods for a given policy. Here, we use $\pi^*$ obtained from the standard off-policy learning. From the results in Figure 1, we first observe that the solutions to the ERM subproblem ($\hat{f}_a$ and $\hat{g}_a$) provide reasonable bounds for the RM method. The solutions to the proposed optimization problems in Section  also reasonably provide lower bounds to the IPS and DR estimators. Other than the fact that IPS estimator suffers from variance issues, the performances of the bounding methods are generally consistent. It is also expected that a larger $\alpha$ leads to looser bounds for all the estimators.

Next, we compare the testing performance and the robustness of $\pi^*$ optimized by standard off-policy learning and our approach. In particular, the testing regret is computed by first adding the runtime uncertainty to $\pi^*$. Then we examine the robustness of $\pi^*$ by checking how much it might fluctuate via the gap of: $\hat{V}_{\text{DR}}(\pi^*, \pi_0) - \underline{\hat{V}_{\text{DR}}}(\pi^*, \pi_0)$. The gap provides a reasonable measurement since $\underline{\hat{V}_{\text{DR}}}(\pi^*, \pi_0)$ gives the worst-possible performance downgrade caused by runtime uncertainty. The results are in Figure 2. In the upper panel, we see that the proposed approach achieves better

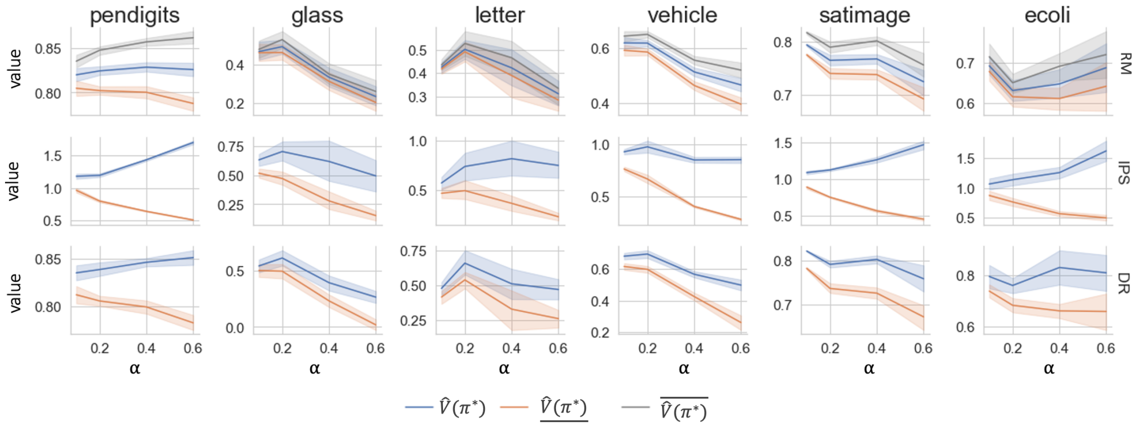

Figure 1: The bounds for the RM, IPS and DR estimators for the uncertainty-perturbed data, under different values of $\alpha$. The values for the original estimators are in blue, and the values of the bounding methods are given by orange and gray. We only show the lower bounds for IPS and DR for the sake of presentation as their tail gets loose under large $\alpha$, which is caused by the extreme (small) propensity weights.

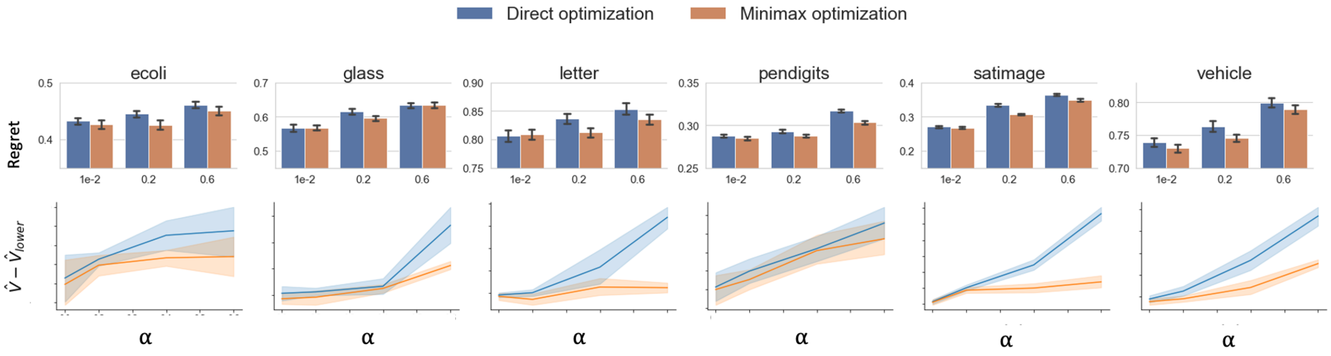

Figure 2: Upper: the regret $(1 - \text{reward})$ on the uncertainty-perturbed testing data under different $\alpha$; Lower: the robustness of the trained policies using standard off-policy learning (direct optimization) and our approach (minimax optimization), measured on the uncertainty-perturbed testing data by how much they would fluctuate, i.e. $\hat{V}^*_{\text{DR}} - \underline{\hat{V}^*_{\text{DR}}}$, where $V^*$ denotes the trained policies.

regrets for most cases, especially under larger $\alpha$. The simulation results justify the effectiveness of our learning approach, particularly as runtime uncertainty increases. In the lower panel of Figure 2, we see the proposed algorithm significantly improves the robustness of the optimized policy $\pi^*$, and the degree of improvement increases with $\alpha$. In the appendix, we provide in-depth analysis of the impact of $\alpha$ on off-policy learning, as well as the comparisons with the *oracle* regret.

**Real-world deployment and analysis**. We conduct both offline and online experiments via the production platform of a major e-commerce website in the U.S. We use contextual bandit to personalize the daily homepage product recommendations, and we consider the click-through rate (CTR) as the reward. We leave the background detail, offline results and analysis in the appendix, and discuss the online testing below. We conduct online A/B testing to compare the proposed approach and the standard off-policy learning; both rely on the DR method with properly truncated propensity scores. The value of $\alpha$ is treated as a tuning parameter and selected via the validation data. For a typical e-commerce platform, runtime uncertainties can be induced by *special offers*, *stock availability*, *upselling events*, and *infrastructural malfunction* with the pipeline, caching, front-end computation and interleaving experiments. Trained under the same data and model family, during the 21-day testing period, the proposed approach consistently outperforms the standard off-policy learning, which often suffers from the previously identified runtime uncertainties. We defer the details of the onine experiments and results to the appendix.

## Conclusion

We study the novel problem of robust off-policy learning for runtime uncertainty. We propose a principled solution with max-min learning, and justify the theoretical implications and guarantees. Our solution is examined via simulation and real-world testings. In the presence of runtime uncertainty, our approach compares favorably to standard off-policy learning, and can extend directly to diverse problem settings. We hope our work promotes future research on practical post-deployment robustness of AI solutions.

# References

Angrist, J. D.; and Imbens, G. W. 1995. Two-stage least squares estimation of average causal effects in models with variable treatment intensity. *Journal of the American statistical Association*, 90(430): 431–442.

Athey, S.; and Wager, S. 2017. Efficient policy learning. *arXiv preprint arXiv:1702.02896*.

Bareinboim, E.; Forney, A.; and Pearl, J. 2015. Bandits with unobserved confounders: A causal approach. In *Advances in Neural Information Processing Systems*, 1342–1350.

Ben-Tal, A.; El Ghaoui, L.; and Nemirovski, A. 2009. *Robust optimization*, volume 28. Princeton University Press.

Bertsimas, D.; Brown, D. B.; and Caramanis, C. 2011. Theory and applications of robust optimization. *SIAM review*, 53(3): 464–501.

Blackwell, M.; Honaker, J.; and King, G. 2017. A unified approach to measurement error and missing data: overview and applications. *Sociological Methods & Research*, 46(3): 303–341.

Bottou, L.; Peters, J.; Quiñonero-Candela, J.; Charles, D. X.; Chickering, D. M.; Portugaly, E.; Ray, D.; Simard, P.; and Snelson, E. 2013. Counterfactual reasoning and learning systems: The example of computational advertising. *The Journal of Machine Learning Research*, 14(1): 3207–3260.

Bubeck, S.; and Slivkins, A. 2012. The best of both worlds: Stochastic and adversarial bandits. In *Conference on Learning Theory*, 42–1.

Carlini, N.; and Wagner, D. A. 2017. Towards evaluating the robustness of neural networks. CoRR abs/1608.04644 (2016). *IEEE Symposium on Security and Privacy*, 39–57.

Dempster, A. P.; Laird, N. M.; and Rubin, D. B. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1): 1–22.

Ding, P.; and VanderWeele, T. J. 2016. Sensitivity analysis without assumptions. *Epidemiology (Cambridge, Mass.)*, 27(3): 368.

Dudík, M.; Erhan, D.; Langford, J.; Li, L.; et al. 2014. Doubly robust policy evaluation and optimization. *Statistical Science*, 29(4): 485–511.

Dudík, M.; Langford, J.; and Li, L. 2011. Doubly robust policy evaluation and learning. *arXiv preprint arXiv:1103.4601*.

Farajtabar, M.; Chow, Y.; and Ghavamzadeh, M. 2018. More robust doubly robust off-policy evaluation. *arXiv preprint arXiv:1802.03493*.

Ghosh, A.; Kumar, H.; and Sastry, P. 2017. Robust loss functions under label noise for deep neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.

Gilotte, A.; Calauzènes, C.; Nedelec, T.; Abraham, A.; and Dollé, S. 2018. Offline a/b testing for recommender systems. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, 198–206.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2015. Explaining and harnessing adversarial examples. *Proceedings of the International Conference on Learning Representations*.

Horvitz, D. G.; and Thompson, D. J. 1952. A generalization of sampling without replacement from a finite universe. *Journal of the American statistical Association*, 47(260): 663–685.

Jagerman, R.; Markov, I.; and de Rijke, M. 2019. When people change their mind: Off-policy evaluation in non-stationary recommendation environments. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, 447–455.

Jin, C.; Allen-Zhu, Z.; Bubeck, S.; and Jordan, M. I. 2018. Is Q-learning provably efficient? In *Advances in Neural Information Processing Systems*, 4863–4873.

Kallus, N.; and Zhou, A. 2018. Confounding-robust policy improvement. In *Advances in neural information processing systems*, 9269–9279.

Kurakin, A.; Goodfellow, I.; and Bengio, S. 2016. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*.

Lange, K. 2016. *MM optimization algorithms*. SIAM.

Lee, J.; and Raginsky, M. 2018. Minimax statistical learning with wasserstein distances. In *Advances in Neural Information Processing Systems*, 2687–2696.

Ma, Y.; Wang, Y.-X.; and Narayanaswamy, B. 2019. Imitation-regularized offline learning. In *The 22nd International Conference on Artificial Intelligence and Statistics*, 2956–2965. PMLR.

Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2018. Towards deep learning models resistant to adversarial attacks. *International Conference on Learning Representations*.

Natarajan, N.; Dhillon, I. S.; Ravikumar, P. K.; and Tewari, A. 2013. Learning with noisy labels. *Advances in neural information processing systems*, 26: 1196–1204.

Neumayer, E.; and Plümper, T. 2017. *Robustness tests for quantitative research*. Cambridge University Press.

Northcutt, C.; Jiang, L.; and Chuang, I. 2021. Confident learning: Estimating uncertainty in dataset labels. *Journal of Artificial Intelligence Research*, 70: 1373–1411.

Pearl, J. 2009. *Causality*. Cambridge university press.

Pearl, J.; and Mackenzie, D. 2018. *The book of why: the new science of cause and effect*. Basic books.

Robins, J. M.; Hernán, M. A.; and Brumback, B. 2000. Marginal Structural Models and Causal Inference in Epidemiology. *Epidemiology*, 11(5): 551.

Robins, J. M.; Rotnitzky, A.; and Zhao, L. P. 1994. Estimation of regression coefficients when some regressors are not always observed. *Journal of the American statistical Association*, 89(427): 846–866.

Rosenbaum, P. R.; and Rubin, D. B. 1983. Assessing sensitivity to an unobserved binary covariate in an observational study with binary outcome. *Journal of the Royal Statistical Society: Series B (Methodological)*, 45(2): 212–218.

Rosenbaum, P. R.; and Silber, J. H. 2009. Amplification of sensitivity analysis in matched observational studies. *Journal of the American Statistical Association*, 104(488): 1398–1405.

Si, N.; Zhang, F.; Zhou, Z.; and Blanchet, J. 2020. Distributional Robust Batch Contextual Bandits. *arXiv preprint arXiv:2006.05630*.

Sutton, R. S.; and Barto, A. G. 2018. *Reinforcement learning: An introduction*. MIT press.

Swaminathan, A.; and Joachims, T. 2015a. Counterfactual risk minimization: Learning from logged bandit feedback. In *International Conference on Machine Learning*, 814–823.

Swaminathan, A.; and Joachims, T. 2015b. The self-normalized estimator for counterfactual learning. In *advances in neural information processing systems*, 3231–3239.

Thomas, P.; and Brunskill, E. 2016. Data-efficient off-policy policy evaluation for reinforcement learning. In *International Conference on Machine Learning*, 2139–2148.

Vlassis, N.; Bibaut, A.; Dimakopoulou, M.; and Jebara, T. 2019. On the Design of Estimators for Bandit Off-Policy Evaluation. In *International Conference on Machine Learning*, 6468–6476.

Xie, T.; Ma, Y.; and Wang, Y.-X. 2019. Towards optimal off-policy evaluation for reinforcement learning with marginalized importance sampling. In *Advances in Neural Information Processing Systems*, 9668–9678.

Xu, D.; Ye, Y.; Ruan, C.; Korpeoglu, E.; Kumar, S.; and Achan, K. 2021. From Intervention to Domain Transportation: A Novel Perspective to Optimize Recommendation. In *International Conference on Learning Representations*.

Ye, Y.; and Bickel, P. J. 2021. Binomial Mixture Model With U-shape Constraint. *arXiv preprint arXiv:2107.13756*.

Zhao, Q.; Small, D. S.; and Bhattacharya, B. B. 2019. Sensitivity analysis for inverse probability weighting estimators via the percentile bootstrap. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 81(4): 735–761.

Zheng, S.; Wu, P.; Goswami, A.; Goswami, M.; Metaxas, D.; and Chen, C. 2020. Error-bounded correction of noisy labels. In *International Conference on Machine Learning*, 11447–11457. PMLR.