Counterfactual Predictions under Runtime Confounding

Amanda Coston
Heinz College & Machine Learning Dept.
Carnegie Mellon University
acoston@cs.cmu.edu

Edward H. Kennedy
Department of Statistics
Carnegie Mellon University
edward@stat.cmu.edu

Alexandra Chouldechova
Heinz College
Carnegie Mellon University
achould@cmu.edu

Abstract

Algorithms are commonly used to predict outcomes under a particular decision or intervention, such as predicting whether an offender will succeed on parole if placed under minimal supervision. Generally, to learn such *counterfactual* prediction models from observational data on historical decisions and corresponding outcomes, one must measure all factors that jointly affect the outcomes and the decision taken. Motivated by decision support applications, we study the counterfactual prediction task in the setting where all relevant factors are captured in the historical data, but it is either undesirable or impermissible to use some such factors in the prediction model. We refer to this setting as **runtime confounding**. We propose a doubly-robust procedure for learning counterfactual prediction models in this setting. Our theoretical analysis and experimental results suggest that our method often outperforms competing approaches. We also present a validation procedure for evaluating the performance of counterfactual prediction methods.

1 Introduction

Algorithmic tools are increasingly prevalent in domains such as health care, education, lending, criminal justice, and child welfare [2, 7, 12, 15, 30]. In many cases, the tools are not intended to replace human decision-making, but rather to distill rich case information into a simpler form, such as a risk score, to inform human decision makers [1, 9]. The type of information that these tools need to convey is often *counterfactual* in nature. Decision-makers need to know what is likely to happen if they choose to take a particular action. For instance, an undergraduate program advisor determining which students to recommend for a personalized case management program might wish to know the likelihood that a given student will graduate if enrolled in the program. In criminal justice, a parole board determining whether to release an offender may wish to know the likelihood that the offender will succeed on parole under different possible levels of supervision intensity.

A common challenge to developing valid counterfactual prediction models is that all the data available for training and evaluation is observational: the data reflects historical decisions and outcomes under those decisions rather than randomized trials intended to assess outcomes under different policies. If the data is confounded—that is, if there are factors not captured in the data that influenced both the outcome of interest and historical decisions—valid counterfactual prediction may not be possible. In this paper we consider the setting where all relevant factors are captured in the data, and so historical decisions and outcomes are unconfounded, but where it is either undesirable or impermissible to use some such factors in the decision support model. We refer to this setting as **runtime confounding**.

Runtime confounding naturally arises in a number of different settings. First, relevant factors may not yet be measured at the desired runtime. Consider an in-person parole hearing, where the parole board makes a recommendation after reviewing documents and hearing spoken testimony. The testimony may provide information that both influences the board's decision and reveals drivers of the offender's likelihood to succeed if released. But the parole board would generally wish to see the predictions of a criminal risk and needs assessment tool prior to the hearing. Any such tool could not therefore use spoken testimony as a model input, thereby leading to runtime confounding. Second, runtime confounding arises when historical decisions and outcomes have been affected by sensitive or protected attributes which for legal or ethical reasons are deemed ineligible as inputs to algorithmic predictions. We may for instance be concerned that parole boards implicitly relied on race in their decisions, but it would not be permissible to include race as a model input. Third, runtime confounding may result from interpretability or simplicity requirements. For example, a university may require algorithmic tools used for case management to be interpretable. While information conveyed during student-advisor meetings is likely informative both of case management decisions and student outcomes, natural language processing models are not classically interpretable, and thus the university may wish instead to only use structured information like GPA in their tools.

In practice, when it is undesirable or impermissible to use particular features as model inputs at runtime, it is common to discard the ineligible features from the training process. This can induce considerable bias in the resulting prediction model when the discarded features are significant confounders. To our knowledge, the problem of learning valid counterfactual prediction models under runtime confounding has not been considered in the prior literature, leaving practitioners without the tools to properly incorporate runtime-ineligible confounding features into the training process.

Contributions: Drawing upon techniques used in low-dimensional treatment effect estimation [6, 37, 41], we propose a procedure for the full pipeline of learning and evaluating prediction models under runtime confounding. We (1) formalize the problem of counterfactual prediction with runtime confounding [§ 2]; (2) propose a solution based on doubly-robust techniques that has desirable theoretical properties [§ 3.3]; (3) theoretically and empirically compare this solution to an alternative counterfactually valid approach as well as the standard practice, describing the conditions under which we expect each to perform well [§ 3 & 5]; and (4) provide an evaluation procedure to assess performance of the methods in the real-world [§ 4]. All proofs are presented in the Supplement.

1.1 Related work

Our work builds upon a growing literature on counterfactual risk assessments for decision support that proposes methods for the unconfounded prediction setting [8, 28]. Following this literature, our goal is to predict outcomes under a proposed decision (interchageably referred to as 'treatment' or 'intervention') in order to inform human decision-makers about what is likely to happen under that treatment. This prediction task is different from the common causal inference problem of treatment effect estimation, which targets a contrast of outcomes under two different treatments [29, 38]. Treatment effects are useful for describing responsiveness to treatment. While responsiveness is relevant to some types of decisions, it is insufficient, or even irrelevant, to consider for others. For instance, a doctor considering an invasive procedure may make a different recommendation for two patients with the same responsiveness if one has a good probability of successful recovery without the procedure and the other does not. In other settings, such as loan approval, the responsiveness to different loan terms is irrelevant; all that matters is that the likelihood of default be sufficiently small under some feasible terms.

Our proposed prediction (Contribution 2) and evaluation methods (Contribution 4) draw upon the literature on double machine learning and doubly-robust estimation, which uses the efficient influence function to produce estimators with reduced bias [5, 13, 22, 23, 37]. Of particular relevance are methods for estimating treatment effects conditional on only a subset of confounders [6, 18, 37, 41]. In our case, we are interested in *predictions* conditional on only those features that are permissible or desirable to consider at runtime. Our methods are specifically designed for minimizing prediction error, rather than providing inferential guarantees such as confidence intervals, as is common in the treatment effect estimation setting.

Our work is also related to the literature on marginal structure models (MSMs) [21, 24]. An MSM is a model for a marginal mean of a counterfactual, possibly conditional on a subset of baseline covariates. The standard MSM approach is semiparametric, employing parametric assumptions for the marginal mean but leaving other components of the data-generating process unspecified [36]. Nonparametric variants were studied in the unconditional case for continuous treatments by Rubin and van der Laan [25]. In contrast our setting can be viewed as a nonparametric MSM for a binary treatment, conditional on a large subset of covariates. This is similar in spirit to partly-conditional treatment effect estimation [35]; however we do not target a contrast since our interest is in predictions rather than treatment effects. Our results are also less focused on model selection [34], and more on error rates for particular estimators. We draw on techniques for sample-splitting and cross-fitting, which have been used in the regression setting for model selection and tuning [10, 37] and in treatment effect estimation [4, 20, 40].

Recent work has also considered methods to accommodate confounding due to sources other than missing confounders at runtime. A line of work has considered how to use causal techniques to correct runtime dataset shift [17, 31, 32]. By contrast, in our case the runtime setting is different from the training setting not because of distributional shift but because we can no longer access all confounders. These methods also differ from ours in that they are not seeking to predict outcomes under specific decisions.

Lastly, there is a line of work that considers confounding in the *training* data [11, 16]. While confounded training data is common in various applications, our work targets decision support settings where the factors used by the decision-maker are recorded in the training data but are not available at prediction time.

2 Problem setting

Our goal is to predict outcomes under a proposed treatment $A = a \in \{0, 1\}$ based on runtime-available predictors $V \in \mathcal{V} \subseteq \mathbb{R}^{d_{\mathrm{V}}}$.[1] Using the potential outcomes framework [19, 26], our prediction target is $\nu_a(v) := \mathbb{E}[Y^a \mid V = v]$ where $Y^a \in \mathcal{Y} \subseteq \mathbb{R}$ is the potential outcome we would observe under treatment $A = a$. We let $Z \in \mathcal{Z} \subseteq \mathbb{R}^{d_{\mathrm{Z}}}$ denote the runtime-hidden confounders, and we denote the propensity to assign treatment a by $\pi_a(v, z) := \mathbb{P}(A = a \mid V = v, Z = z)$. We also define the outcome regression by $\mu_a(v, z) := \mathbb{E}[Y^a \mid V = v, Z = z]$. For brevity, we will generally omit the subscript, using notation ν, π and μ to denote the functions for a generic

[1] For exposition, we focus on making predictions for a single binary treatment a. To make predictions under multiple discrete treatments, our method can be repeated for each treatment using a one-vs-all setup.

treatment a.

Definition 2.1. Formally, the task of counterfactual prediction under **runtime-only confounding** is to estimate $\nu(v)$ from iid training data (V, Z, A, Y) under the following two conditions:

Condition 2.1.1 (Training Ignorability)**.** Outcomes and decisions are unconfounded given V and Z: $Y^a \perp A \mid V, Z$.

Condition 2.1.2 (Runtime Confounding)**.** Outcomes and decisions are confounded given only V: $Y^a \not\perp A \mid V$; equivalently, $A \not\perp Z \mid V$ and $Y^a \not\perp Z \mid V$

To ensure that the target quantity is identifiable, we require two further assumptions, which are standard in causal inference and not specific to the runtime confounding setting.

Condition 2.1.3 (Consistency)**.** A case that receives treatment a has outcome $Y = Y^a$.

Condition 2.1.4 (Positivity)**.** $\mathbb{P}(\pi(V, Z) > 0) = 1$

Identifications. Under conditions 2.1.1-2.1.4, we can write the counterfactual regression functions μ and ν in terms of observable quantities. We can identify $\mu(v, z) = \mathbb{E}[Y \mid V = v, Z = z, A = a]$ and our target $\nu(v) = \mathbb{E}[\mathbb{E}[Y \mid V = v, Z = z, A = a] \mid V = v] = \mathbb{E}[\mu(V, Z) \mid V = v]$. The iterated expectation in the identification of ν suggests a two-stage approach that we propose in § 3.2 after reviewing current approaches.

Miscellaneous notation. Throughout the paper we let $p(x)$ denote probability density functions; $\hat{f}$ denote an estimate of f; $L \lesssim R$ indicate that $L \leq C \cdot R$ for some universal constant C; $\mathbb{I}$ denote the indicator function; and define $\|f\|^2 := \int (f(x))^2 p(x) dx$.

3 Prediction methods

3.1 Standard practice: Treatment-conditional regression (TCR)

Standard counterfactual prediction methods train models on the cases that received treatment a [8, 28], a procedure we will refer to as **treatment-conditional regression** (TCR). This procedure estimates $\omega(v) = \mathbb{E}[Y \mid A = a, V = v]$. This method works well given access to all the confounders at runtime; if $A \perp Y^a \mid V$, then $\omega(v) = \mathbb{E}[Y^a \mid V = v] = \nu(v)$. However, under runtime confounding, $\omega(v) \neq \mathbb{E}[Y^a \mid V = v]$, so this method does not target the right counterfactual quantity, and may produce misleading predictions. For instance, consider a risk assessment setting that historically assigned risk-mitigating treatment to cases that have higher risk under the null treatment $(A = 0)$. Using TCR to predict outcomes under the null treatment will underestimate risk since $\mathbb{E}[Y \mid V, A = 0] = \mathbb{E}[Y^0 \mid V, A = 0] < \mathbb{E}[Y^0 \mid V]$. We can characterize the bias of this approach by analyzing $b(v) := \omega(v) - \nu(v)$, a quantity we term the pointwise *confounding bias*.

Proposition 3.1. Under runtime confounding, $\omega(v)$ has pointwise confounding bias

$$b(v) = \int_{\mathcal{Z}} \mu(v, z)\Big(p(z \mid V = v, A = a) - p(z \mid V = v)\Big)dz \quad \neq \quad 0 \tag{1}$$

Algorithm 1 The plug-in (PL) approach

```
Stage 1: Learn $\hat{\mu}(v, z)$ by regressing $Y \sim V, Z \mid A = a$
Stage 2: Learn $\hat{\nu}_{\mathrm{PL}}(v)$ by regressing $\hat{\mu}(V, Z) \sim V$
```

Algorithm 2 The plug-in (PL) approach with cross-fitting

```
Randomly divide training data into two partitions $\mathcal{W}^1$ and $\mathcal{W}^2$.
for $(p, q) \in \{(1, 2), (2, 1)\}$ do
    Stage 1: On partition $\mathcal{W}^p$, learn $\hat{\mu}^p(v, z)$ by regressing $Y \sim V, Z \mid A = a$
    Stage 2: On partition $\mathcal{W}^q$, learn $\hat{\nu}^q_{\mathrm{PL}}(v)$ by regressing $\hat{\mu}^p(V, Z) \sim V$
PL prediction: $\hat{\nu}_{\mathrm{PL}}(v) = \frac{1}{2}\sum_{i=1}^{2} \hat{\nu}^i_{\mathrm{PL}}(v)$
```

By Condition 2.1.2, this confounding bias will be non-zero. Nonetheless we might expect the TCR method to perform well if $b(v)$ is small enough. We can formalize this intuition by decomposing the error of a TCR predictive model $\hat{\nu}_{\mathrm{TCR}}$ into estimation error and confounding bias:

Proposition 3.2. The pointwise regression error of the TCR method can be bounded as follows:

$$\mathbb{E}[(\nu(v) - \hat{\nu}_{\mathrm{TCR}}(v))^2] \lesssim \mathbb{E}[(\omega(v) - \hat{\nu}_{\mathrm{TCR}}(v))^2] + b(v)^2$$

3.2 A simple proposal: Plug-in (PL) approach

We can avoid the confounding bias of TCR through a simple two-stage procedure we call the **plug-in** approach that targets the proper counterfactual quantity. This approach, described in Algorithm 1, first estimates μ and then uses this to construct a pseudo-outcome which is regressed on V to yield prediction $\hat{\nu}_{\mathrm{PL}}$. Cross-fitting techniques (Alg. 2) can be applied to prevent issues that may arise due to potential overfitting when learning both $\hat{\mu}$ and $\hat{\nu}_{\mathrm{PL}}$ on the same training data. Sample-splitting (or cross-fitting) also enables us to get the following upper bound on the error of the PL approach.

Proposition 3.3. Under sample-splitting for stages 1 and 2 and stability conditions on the 2nd stage estimators (appendix, [14]), the PL method has pointwise regression error bounded by

$$\mathbb{E}\left[\left(\hat{\nu}_{\mathrm{PL}}(v) - \nu(v)\right)^2\right] \lesssim \mathbb{E}\left[\left(\tilde{\nu}(v) - \nu(v)\right)^2\right] + \mathbb{E}\left[\left(\hat{\mu}(V, Z) - \mu(V, Z)\right)^2 \mid V = v\right]$$

where the oracle-quantity $\tilde{\nu}(v)$ describes the function we would get in the second-stage if we had oracle access to Y^a.

3.3 Our main proposal: Doubly-robust (DR) approach

Our main proposed method is what we call the **doubly-robust** (DR) approach, which improves upon the PL procedure by using a bias-corrected pseudo-outcome in the second stage (Alg. 4). The DR approach estimates both μ and π, which enables the method to perform well in situations in which π is easier to estimate than μ. We propose a cross-fitting (Alg. 3) variant that satisfies the sample-splitting requirements of Theorem 3.1.

Algorithm 3 The proposed doubly-robust (DR) approach

```
Stage 1: Learn $\hat{\mu}(v, z)$ by regressing $Y \sim V, Z \mid A = a$.
         Learn $\hat{\pi}(v, z)$ by regressing $\mathbb{I}\{A = a\} \sim V, Z$
Stage 2: Learn $\hat{\nu}_{\text{DR}}(v)$ by regressing $\left(\frac{\mathbb{I}\{A=a\}}{\hat{\pi}(V,Z)}(Y - \hat{\mu}(V, Z)) + \hat{\mu}(V, Z)\right) \sim V$
```

Algorithm 4 The proposed doubly-robust (DR) approach with cross fitting

```
Randomly divide training data into three partitions $\mathcal{W}^1$, $\mathcal{W}^2$, $\mathcal{W}^3$.
for $(p, q, r) \in \{(1, 2, 3), (3, 1, 2), (2, 3, 1)\}$ do
    Stage 1: On $\mathcal{W}^p$, learn $\hat{\mu}^p(v, z)$ by regressing $Y \sim V, Z \mid A = a$.
             On $\mathcal{W}^q$, learn $\hat{\pi}^q(v, z)$ by regressing $\mathbb{I}\{A = a\} \sim V, Z$
    Stage 2: On $\mathcal{W}^r$, learn $\hat{\nu}^r_{\text{DR}}$ by regressing $\left(\frac{\mathbb{I}\{A=a\}}{\hat{\pi}^q(V,Z)}(Y - \hat{\mu}^p(V, Z)) + \hat{\mu}^p(V, Z)\right) \sim V$
DR prediction: $\hat{\nu}_{\text{DR}}(v) = \frac{1}{3}\sum_{i=1}^{3} \hat{\nu}^i_{\text{DR}}(v)$
```

Theorem 3.1. Under sample-splitting to learn $\hat{\mu}$, $\hat{\pi}$, and $\hat{\nu}_{\text{DR}}$ and stability conditions on the 2nd stage estimators (appendix, [14]), the DR method has pointwise error bounded by:

$$\mathbb{E}\Big[\big(\hat{\nu}_{\text{DR}}(v) - \nu(v)\big)^2\Big] \lesssim \mathbb{E}\Big[\big(\tilde{\nu}(v) - \nu(v)\big)^2\Big] + \mathbb{E}\Big[(\hat{\pi}(V, Z) - \pi(V, Z))^2 \mid V = v\Big]\mathbb{E}\Big[(\hat{\mu}(V, Z) - \mu(V, Z))^2 \mid V = v\Big]$$

This implies a similar bound on the integrated MSE (given in appendix).

Theorem 3.1 indicates that the DR approach is oracle-efficient, in the sense that it achieves (up to a constant factor) the same error rate as an oracle with access to Y^a when the product of nuisance function errors is smaller than the oracle error.[2] We emphasize that these bounds hold for any regression method. However, it is also instructive to consider the form of these bounds in a specific context. The next result specialized to the sparse high-dimensional setting.

Corollary 3.1. Assume stability conditions on the 2nd stage regression estimator (appendix, [14]) and that a k-sparse model can be estimated with squared error $k^2\sqrt{\frac{\log d}{n}}$ (e.g. [3]). With k_ω-sparse ω, the pointwise error for the TCR method is

$$\mathbb{E}\Big[\big(\hat{\nu}_{\text{TCR}}(v) - \nu(v)\big)^2\Big] \lesssim\ k_\omega^2\sqrt{\frac{\log d_{\text{V}}}{n}} + b(v)^2$$

With k_μ-sparse μ and k_ν-sparse ν, the pointwise error for the PL method is

$$\mathbb{E}\Big[\big(\hat{\nu}_{\text{PL}}(v) - \nu(v)\big)^2\Big] \lesssim\ k_\nu^2\sqrt{\frac{\log d_{\text{V}}}{n}} + k_\mu^2\sqrt{\frac{\log d}{n}}$$

Additionally with k_π-sparse π, the pointwise error for the DR method is

$$\mathbb{E}\Big[\big(\hat{\nu}_{\text{DR}}(v) - \nu(v)\big)^2\Big] \lesssim\ k_\nu^2\sqrt{\frac{\log d_{\text{V}}}{n}} + k_\mu^2 k_\pi^2\frac{\log d}{n}$$

Corollary 3.1 suggests that when $d_{\text{V}} \approx d$, the DR and PL methods will perform similarly. When $d_{\text{V}} \ll d$, we expect the DR to outperform the PL method. When $d_{\text{V}} \ll d$ and the amount of confounding is small, we expect the TCR to perform well. This theoretical analysis helps us understand when we expect the prediction methods to perform well. However, in practice, these upper bounds may not be tight and the degree of confounding is typically unknown. To compare the prediction methods in practice, we require a method for counterfactual model evaluation.

[2] The term *nuisance* refers to functions μ and π.

Algorithm 5 Cross-fitting approach to evaluation of counterfactual prediction methods

```
Input: Test samples $\{(V_j, Z_j, A_j, Y_j)\}_{j=1}^{2n}$ and prediction models $\{\hat{\nu}_1, ...\hat{\nu}_h\}$
Randomly divide test data into two partitions $\mathcal{W}^0 = \{(V_j^0, Z_j^0, A_j^0, Y_j^0)\}_{j=1}^n$ and $\mathcal{W}^1 = \{(V_j^1, Z_j^1, A_j^1, Y_j^1)\}_{j=1}^n$.
for $(p, q) \in \{(0,1), (1,0)\}$ do
  On $\mathcal{W}^p$, learn $\hat{\pi}^p(v, z)$ by regressing $\mathbb{I}\{A = a\} \sim V, Z$.
  for $m \in \{1, ...., h\}$ do
    On $\mathcal{W}^p$, learn $\hat{\eta}_m^p(v, z)$ by regressing $(Y - \hat{\nu}_m(V))^2 \sim V, Z \mid A = a$
    On $\mathcal{W}^q$, for $j \in \{1, ..., n\}$ compute $\phi_{m,j}^q = \frac{\mathbb{I}\{A_j^q = a\}}{\hat{\pi}^p(V_j^q, Z_j^q)}((Y_j^q - \hat{\nu}_m(V_j^q))^2 - \hat{\eta}_m^p(V_j^q, Z_j^q)) + \hat{\eta}_m^p(V_j^q, Z_j^q)$
Output error estimate confidence intervals: for $m \in \{1, ..., h\}$:
$\mathrm{MSE}_m = \left(\frac{1}{2n} \sum_{i=0}^{1} \sum_{j=1}^{n} \phi_{m,j}^i\right) \pm 1.96\sqrt{\frac{1}{2n}\mathrm{var}(\phi_m)}$
```

4 Evaluation method

We describe an approach for evaluating the prediction methods using observed data. In our problem setting (§ 2.1), the prediction error of a model $\hat{\nu}$ is identified as $\mathbb{E}[(Y^a - \hat{\nu}(V))^2] = \mathbb{E}[\mathbb{E}[(Y - \hat{\nu}(V))^2 \mid V, Z, A = a]]$. We propose a doubly-robust procedure to estimate the prediction error that follows the approach in [8], which focused on classification metrics and therefore did not consider MSE. Defining the error regression $\eta(v, z) := \mathbb{E}[(Y^a - \hat{\nu}(V))^2 | V = v, Z = z]$, which is identified as $\mathbb{E}[(Y - \hat{\nu}(V))^2 \mid V = v, Z = z, A = a]$, the **doubly-robust estimate of the MSE of** ν is

$$\frac{1}{n}\sum_{i=1}^{n}\left[\frac{\mathbb{I}\{A_i = a\}}{\hat{\pi}(V_i, Z_i)}\Big(\big(Y_i - \hat{\nu}(V_i)\big)^2 - \hat{\eta}(V_i, Z_i)\Big) + \hat{\eta}(V_i, Z_i)\right]$$

The doubly-robust estimation of MSE is $\sqrt{n}$-consistent under sample-splitting and $n^{1/4}$ convergence in the nuisance function error terms, enabling us to get estimates with confidence intervals. Algorithm 5 describes this procedure.[3] This evaluation method can also be used to select the regression estimators for the first and second stages.

5 Experiments

We evaluate our methods against ground truth by performing experiments on simulated data, where we can vary the amount of confounding in order to assess the effect on predictive performance. While our theoretical results for PL and DR are obtained under sample splitting, in practice there may be a reluctance to perform sample splitting in training predictive models due to the potential loss in efficiency. In this section we present results where we use the full training data to learn the 1st-stage nuisance functions and 2nd-stage regressions for DR and PL and we use the full training data for the one-stage TCR.[4] This allows us to examine performance in a setting outside what our theory covers.

We first analyze how the methods perform in a sparse linear model. This simple setup enables us to explore how properties like correlation between V and Z impact performance.

[3] The appendix describes a cross-fitting approach to jointly learn and evaluate the three prediction methods.

[4] We report error metrics on a random heldout test set.

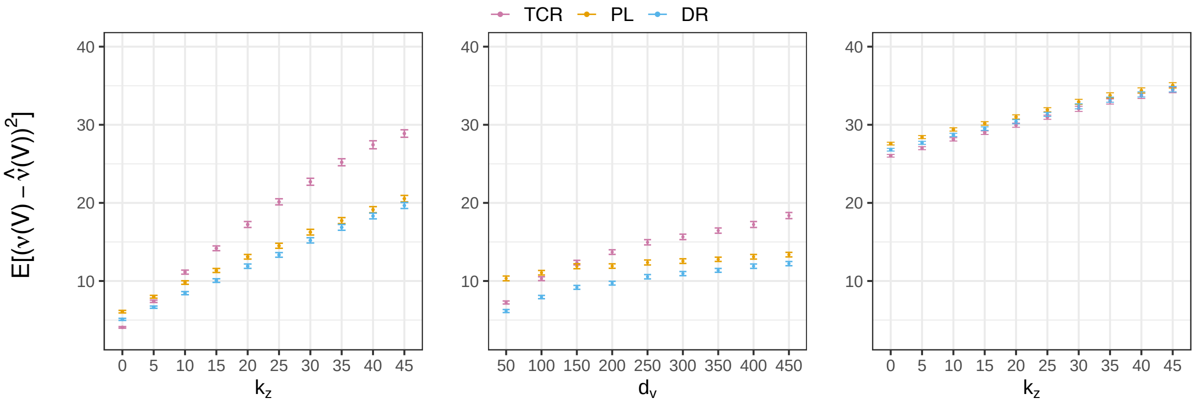
We simulate data as

$$
\begin{aligned}
V_i &\sim \mathcal{N}(0,1) \qquad\qquad ; \ 1 \le i \le d_{\text{V}} \\
Z_i &\sim \mathcal{N}(\rho V_i, 1-\rho^2) \qquad ; \ 1 \le i \le d_{\text{Z}} \\
\mu(V,Z) &= \frac{k_v}{k_v+\rho k_z}\Big(\sum_{i=1}^{k_v} V_i + \sum_{i=1}^{k_z} Z_i\Big) \qquad\qquad Y^a = \mu(V,Z) + \epsilon \ ; \ \epsilon \sim \mathcal{N}\left(0, \frac{1}{2n}\left\|\mu(V,Z)\right\|_2^2\right) \\
\nu(V) &= \frac{k_v}{k_v+\rho k_z}\Big(\sum_{i=1}^{k_v} V_i + \rho\sum_{i=1}^{k_z} V_i\Big) \\
\pi(V,Z) &= 1-\sigma\left(\frac{1}{\sqrt{k_v+k_z}}\Big(\sum_{i=1}^{k_v} V_i + \sum_{i=1}^{k_z} Z_i\Big)\right) \qquad\qquad A \sim \text{Bernoulli}(\pi(V,Z))
\end{aligned}
$$

where $\sigma(x) = \frac{1}{1+e^{-x}}$. We normalize $\pi(v,z)$ by $\frac{1}{\sqrt{k_v+k_z}}$ to satisfy Condition 2.1.4 and use the coefficient $k_v/(k_v+\rho k_z)$ to facilitate a fair comparison as we vary ρ. For all experiments, we report test MSE for 300 simulations where each simulation generates $n = 2000$ data points split randomly and evenly into train and test sets.[5] In the first set of experiments, for fixed $d = d_{\text{V}} + d_{\text{Z}} = 500$, we vary d_{V} (and correspondingly d_{Z}). We also vary k_z, which governs the runtime confounding. Larger values of k_z correspond to more confounding variables. The theoretical analysis (§ 3) suggests that when confounding (k_z) is small, then the TCR and DR methods will perform well. More confounding (larger k_z) should increase error for all methods, and we expect this increase to be significantly larger for the TCR method that has confounding bias. We expect the TCR and DR methods to perform better at smaller values of d_{V}; by contrast, we expect the PL performance to vary less with d_{V} since the PL method suffers from the full d-dimensionality in the first stage regardless of d_{V}. For large values of d_{V}, we expect the PL method to perform similarly to the DR method. Figure 1 plots the MSE in estimating ν for $\rho = 0$ and $k_v = 25$ using LASSO and random forests. The LASSO plots in Figure 1a and 1b show the expected trends. However when using random forests (Figure 1c), we only see a small increase in error across the methods as we increase confounding, suggesting that the estimation error dominates the confounding error when using random forests. In this setting, the TCR method may outperform the other methods, and in fact we see that the TCR performs best at low levels of confounding.

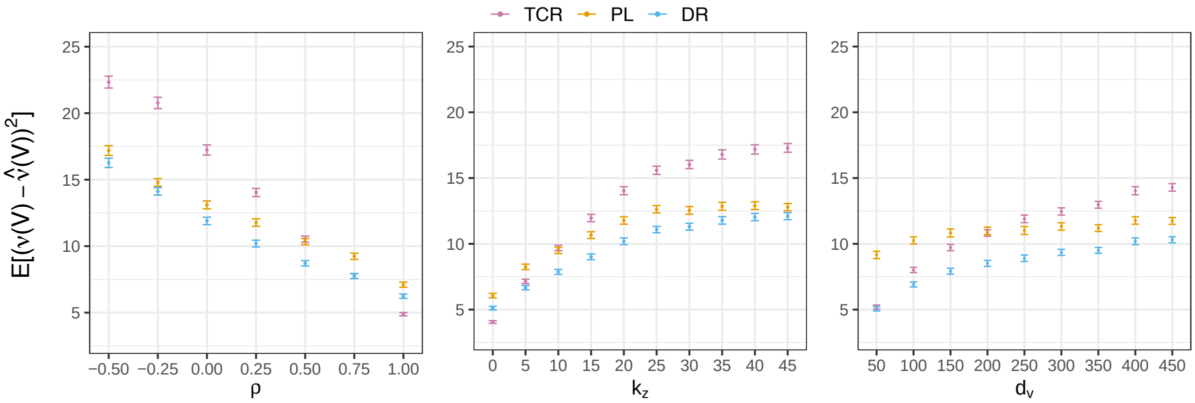
We next consider the case were V and Z are correlated. If V and Z are perfectly correlated, there is no confounding. For our data where higher values of V and Z both decrease π and increase μ, a positive correlation should reduce confounding. A negative correlation may exacerbate confounding by increasing the probability that Z is small given $A = a$ and V is large and therefore increasing the gap $\mathbb{E}[Y^a \mid V = v] - \mathbb{E}[Y^a \mid V = v, A = a]$. In Figure 2 we report MSE for correlated V and Z. As expected, error overall decreases with ρ (Figure 2a). Relative to the uncorrelated setting (Figure 1), the weak positive correlation reduces MSE for all methods, particularly at larger values of k_z and d_{V}. The DR method achieves the lowest error for all settings with confounding, except when $d_{\text{V}} = 50$ the TCR performs on par with DR. Note the inflection point at $k_z = k_v = 25$. After this point, adding confounders only slightly increases error since for $k_v < i \le k_z$, V_i is a correlate for confounder Z_i and V_i itself is not a confounding factor in μ or π.

[5] Code will be available shortly at https://github.com/mandycoston



(a) LASSO (b) LASSO (c) Random forests

Figure 1: **(a)** MSE as we vary k_z using cross-validated LASSO to learn $\hat{\pi}$, $\hat{\mu}$, $\hat{\nu}_{\text{TCR}}$, $\hat{\nu}_{\text{PL}}$, $\hat{\nu}_{\text{DR}}$ for $\rho = 0$, $d_{\text{V}} = 400$, $k_v = 25$. At low levels of confounding (k_z), the TCR method does well but performance degrades with k_z. For any non-zero confounding, our DR method performs best. **(b)** MSE against d_{V} using cross-validated LASSO and $\rho = 0$, $k_v = 25$ and $k_z = 20$. The DR method performs the best across the range of d_{V}. When d_{V} is small, the TCR method also does well since its estimation error is small. The PL method has higher error since it suffers from the full d-dimensional estimating error in the first stage. **(c)** MSE as we vary k_z using random forests and $\rho = 0$, $d_{\text{V}} = 400$ and $k_v = 25$. Compared to LASSO in (a), there is a relatively small increase in error as we increase k_z, suggesting that estimation error dominates the confounding error. The TCR method performs best at lower levels of confounding and on par with the DR method for larger values of k_z.
Error bars denote 95% confidence intervals.



(a) LASSO (b) LASSO (c) LASSO

Figure 2: **(a)** MSE against correlation ρ_{V_i,Z_i} for $k_z = 20$, $k_v = 25$, and $d_{\text{V}} = 400$. Error decreases with ρ for all methods. Our DR method achieves the lowest error under confounding ($\rho < 1$). **(b)** MSE as we increase k_z for $\rho = 0.25$, $k_v = 25$, and $d_{\text{V}} = 400$. Compare to Figure 1a; the weak positive correlation reduces MSE, particularly for $k_v < i \leq k_z$ when V_i is only a correlate for the confounder Z_i but not a confounder itself. **(c)** MSE against d_{V} for $\rho = 0.25$, $k_z = 20$, and $k_v = 25$. The DR method is among the best-performing for all d_{V}. As with the uncorrelated setting (1b), the DR and TCR methods are better able to take advantage of low d_{V} than the PL method.
Error bars denote 95% confidence intervals.

Experiments with Second-Stage Misspecification Next, we explore a more complex data generating process through the lens of model interpretability. Interpretability requirements allow for a complex training process as long as the final model outputs interpretable predictions [27, 33, 39]. Since the PL and DR first stage regressions are only a part of the training process, we can use any flexible model to learn the first stage functions as accurately as possible without impacting interpretability. Constraining the second-stage learning class to interpretable models (e.g. linear classifiers) may cause misspecification since the interpretable class may not contain the true model. We simulate such a setting by modifying the setup (for $\rho = 0$):

$$V_i \sim \mathcal{N}(0,1) \ \text{ for } \ 1 \leq i \leq \frac{d_{\text{V}}}{2} \quad ; \ V_i \coloneqq V_j^2 \ \text{ for } \ \frac{d_{\text{V}}}{2} < i \leq d_{\text{V}}, \ \ j = i - \frac{d_{\text{V}}}{2}$$

$$\mu(V,Z) = \sum_{i=1}^{k_v/2} \Big(V_i + (2(i \bmod 2) - 1)V_i^2 \Big) + \sum_{i=1}^{k_z} Z_i \ \ ; \ \nu(V) = \sum_{i=1}^{k_v/2} \Big(V_i + (2(i \bmod 2) - 1)V_i^2 \Big)$$

We restrict our second stage models and the TCR model to predictors V_i for $1 \leq i \leq \frac{d_{\text{V}}}{2}$ to simulate a real-world setting where we are constrained to linear classifiers using only V at runtime. We allow the first stage models access to the full V and Z since the first stage is not constrained by variables or model class. We use cross-validated LASSO models for both stages and compare this setup to the setting where the model is correctly specified. The DR method achieves the lowest error for both settings (Table 1), although the error is significantly higher for all methods under misspecification.

Method	Correct specification	2nd-stage misspecification
TCR	16.64 (16.28, 17.00)	35.52 (35.18, 35.85)
PL	12.32 (12.03, 12.61)	32.09 (31.82, 32.36)
DR (ours)	**11.10 (10.84, 11.37)**	**31.33 (31.06, 31.59)**

Table 1: MSE $\mathbb{E}[\big(\nu(V) - \hat{\nu}(V)\big)^2]$ under correct specification vs misspecification in the 2nd stage for $d = 500$, $d_{\text{V}} = 400$, $k_v = 24$, $k_z = 20$ and $n = 3000$ (with 95% confidence intervals). Our DR method has the lowest error in both settings. Errors are larger for all methods under misspecification.

6 Conclusion

We propose a generic procedure for learning counterfactual predictions under runtime confounding that can be used with any parametric or nonparametric learning algorithm. Our theoretical and empirical analysis suggests this procedure will often outperform other methods, particularly when the level of runtime confounding is significant.

Acknowledgements

This project received funding from the Tata Consultancy Services (TCS) Presidential Fellowship, Block Center for Technology and Society at Carnegie Mellon University, and NSF grant DMS1810979.

References

[1] Tim Bezemer, Mark CH De Groot, Enja Blasse, Maarten J Ten Berg, Teus H Kappen, Annelien L Bredenoord, Wouter W Van Solinge, Imo E Hoefer, and Saskia Haitjema. A human (e) factor in clinical decision support systems. *Journal of medical Internet research*, 21(3):e11732, 2019.

[2] Rich Caruana, Yin Lou, Johannes Gehrke, Paul Koch, Marc Sturm, and Noemie Elhadad. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1721–1730. ACM, 2015.

[3] Sourav Chatterjee. Assumptionless consistency of the lasso. *arXiv preprint arXiv:1303.5817*, 2013.

[4] Victor Chernozhukov, Denis Chetverikov, Mert Demirer, Esther Duflo, Christian Hansen, Whitney Newey, and James Robins. Double/debiased machine learning for treatment and causal parameters. *The Econometrics Journal*, 2018.

[5] Victor Chernozhukov, Denis Chetverikov, Mert Demirer, Esther Duflo, Christian Hansen, Whitney Newey, and James Robins. Double/debiased machine learning for treatment and structural parameters, 2018.

[6] Victor Chernozhukov, Mert Demirer, Esther Duflo, and Ivan Fernandez-Val. Generic machine learning inference on heterogenous treatment effects in randomized experiments. Technical report, National Bureau of Economic Research, 2018.

[7] Alexandra Chouldechova, Diana Benavides-Prado, Oleksandr Fialko, and Rhema Vaithianathan. A case study of algorithm-assisted decision making in child maltreatment hotline screening decisions. In *Conference on Fairness, Accountability and Transparency*, pages 134–148, 2018.

[8] Amanda Coston, Alan Mishler, Edward H Kennedy, and Alexandra Chouldechova. Counterfactual risk assessments, evaluation, and fairness. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 582–593, 2020.

[9] Maria De-Arteaga, Riccardo Fogliato, and Alexandra Chouldechova. A case for humans-in-the-loop: Decisions in the presence of erroneous algorithmic scores. *arXiv preprint arXiv:2002.08035*, 2020.

[10] László Györfi, Michael Kohler, Adam Krzyzak, and Harro Walk. *A distribution-free theory of nonparametric regression*. Springer Science & Business Media, 2006.

[11] Nathan Kallus and Angela Zhou. Confounding-robust policy improvement. In *Advances in Neural Information Processing Systems*, pages 9269–9279, 2018.

[12] Danielle Leah Kehl and Samuel Ari Kessler. Algorithms in the criminal justice system: Assessing the use of risk assessments in sentencing. 2017.

[13] Edward H Kennedy. Semiparametric theory and empirical processes in causal inference. In *Statistical causal inferences and their applications in public health research*, pages 141–167. Springer, 2016.

[14] Edward H Kennedy. Optimal doubly robust estimation of heterogeneous causal effects. *arXiv preprint arXiv:2004.14497*, 2020.

[15] Amir E Khandani, Adlar J Kim, and Andrew W Lo. Consumer credit-risk models via machine-learning algorithms. *Journal of Banking & Finance*, 34(11):2767–2787, 2010.

[16] David Madras, Elliot Creager, Toniann Pitassi, and Richard Zemel. Fairness through causal awareness: Learning causal latent-variable models for biased data. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 349–358. ACM, 2019.

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

[18] Maggie Makar, Adith Swaminathan, and Emre Kıcıman. A distillation approach to data efficient individual treatment effect estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4544–4551, 2019.

[19] J Neyman. Sur les applications de la theorie des probabilites aux experiences agricoles: essai des principes (masters thesis); justification of applications of the calculus of probabilities to the solutions of certain questions in agricultural experimentation. excerpts english translation (reprinted). *Stat Sci*, 5:463–472, 1923.

[20] James Robins, Lingling Li, Eric Tchetgen, Aad van der Vaart, et al. Higher order influence functions and minimax estimation of nonlinear functionals. In *Probability and statistics: essays in honor of David A. Freedman*, pages 335–421. Institute of Mathematical Statistics, 2008.

[21] James M Robins. Marginal structural models versus structural nested models as tools for causal inference. In *Statistical models in epidemiology, the environment, and clinical trials*, pages 95–133. Springer, 2000.

[22] James M Robins and Andrea Rotnitzky. Semiparametric efficiency in multivariate regression models with missing data. *Journal of the American Statistical Association*, 90(429): 122–129, 1995.

[23] James M Robins, Andrea Rotnitzky, and Lue Ping Zhao. Estimation of regression coefficients when some regressors are not always observed. *Journal of the American statistical Association*, 89(427):846–866, 1994.

[24] James M Robins, Miguel Angel Hernan, and Babette Brumback. Marginal structural models and causal inference in epidemiology, 2000.

[25] Daniel Rubin and Mark J van der Laan. Extending marginal structural models through local, penalized, and additive learning. 2006.

[26] Donald B Rubin. Causal inference using potential outcomes: Design, modeling, decisions. *Journal of the American Statistical Association*, 100(469):322–331, 2005.

[27] Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215, 2019.

[28] Peter Schulam and Suchi Saria. Reliable decision support using counterfactual models. In *Advances in Neural Information Processing Systems*, pages 1697–1708, 2017.

[29] Uri Shalit, Fredrik D Johansson, and David Sontag. Estimating individual treatment effect: generalization bounds and algorithms. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3076–3085. JMLR. org, 2017.

[30] Vernon C Smith, Adam Lange, and Daniel R Huston. Predictive modeling to forecast student outcomes and drive effective interventions in online community college courses. *Journal of Asynchronous Learning Networks*, 16(3):51–61, 2012.

[31] Adarsh Subbaswamy and Suchi Saria. Counterfactual normalization: Proactively addressing dataset shift and improving reliability using causal mechanisms. *Uncertainty in Artificial Intelligence*, 2018.

[32] Adarsh Subbaswamy, Peter Schulam, and Suchi Saria. Preventing failures due to dataset shift: Learning predictive models that transport. *arXiv preprint arXiv:1812.04597*, 2018.

[33] Sarah Tan, Rich Caruana, Giles Hooker, and Yin Lou. Distill-and-compare: Auditing black-box models using transparent model distillation. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pages 303–310, 2018.

[34] Mark J Van Der Laan and Sandrine Dudoit. Unified cross-validation methodology for selection among estimators and a general cross-validated adaptive epsilon-net estimator: Finite sample oracle inequalities and examples. 2003.

[35] Mark J van der Laan and Alexander R Luedtke. Targeted learning of an optimal dynamic treatment, and statistical inference for its mean outcome. 2014.

[36] Mark J Van der Laan, MJ Laan, and James M Robins. *Unified methods for censored longitudinal data and causality*. Springer Science & Business Media, 2003.

[37] Mark J Van der Laan, MJ Laan, and James M Robins. *Unified methods for censored longitudinal data and causality*. Springer Science & Business Media, 2003.

[38] Stefan Wager and Susan Athey. Estimation and inference of heterogeneous treatment effects using random forests. *Journal of the American Statistical Association*, 113(523): 1228–1242, 2018.

[39] Jiaming Zeng, Berk Ustun, and Cynthia Rudin. Interpretable classification models for recidivism prediction. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 180(3):689–722, 2017.

[40] Wenjing Zheng and Mark J van der Laan. Asymptotic theory for cross-validated targeted maximum likelihood estimation. *UC Berkeley Division of Biostatistics Working Paper Series*, 2010.

[41] Michael Zimmert and Michael Lechner. Nonparametric estimation of causal heterogeneity under high-dimensional confounding. *arXiv preprint arXiv:1908.08779*, 2019.

A Details on Proposed Learning Procedure

We describe a joint approach to learning and evaluating the three prediction methods in Algorithm 6. This approach efficiently makes use of the need for both prediction and evaluation methods to estimate the propensity score π.

Algorithm 6 Cross-fitting procedure to learn and evaluate the three prediction methods

```
Input: Data samples $\{(V_j, Z_j, A_j, Y_j)\}_{j=1}^{4n}$
Randomly divide training data into four partitions $\mathcal{W}^1$, $\mathcal{W}^2$, $\mathcal{W}^3$, $\mathcal{W}^4$ where $\mathcal{W}^1 = \{(V_j^1, Z_j^1, A_j^1, Y_j^1)\}_{j=1}^{n}$ (and similarly for $\mathcal{W}^2$, $\mathcal{W}^3$, $\mathcal{W}^4$).
for $(p, q, r, s) \in \{(1,2,3,4), (4,1,2,3), (3,4,1,2), (2,3,4,1)\}$ do
    Stage 1: On $\mathcal{W}^p$, learn $\hat{\mu}^p(v, z)$ by regressing $Y \sim V, Z \mid A = a$.
             On $\mathcal{W}^q$, learn $\hat{\pi}^q(v, z)$ by regressing $\mathbb{I}\{A = a\} \sim V, Z$
    Stage 2: On $\mathcal{W}^r$, learn $\hat{\nu}_{\text{DR}}^r$ by regressing $\left(\frac{\mathbb{I}\{A=a\}}{\hat{\pi}^q(V,Z)}(Y - \hat{\mu}^p(V, Z)) + \hat{\mu}^p(V, Z)\right) \sim V$
             On $\mathcal{W}^r$ and $\mathcal{W}^q$, learn $\hat{\nu}_{\text{PL}}^r$ by regressing $\hat{\mu}^p(V, Z) \sim V$
             On $\mathcal{W}^r$, $\mathcal{W}^q$, and $\mathcal{W}^p$, learn $\hat{\nu}_{\text{TCR}}^r$ by regressing $Y \sim V \mid A = a$
    Evaluate for $m$ in { TCR, PL, DR }:
             On $\mathcal{W}^q$, learn $\hat{\eta}_m^q(v, z)$ by regressing $(Y - \hat{\nu}_m^r(V))^2 \sim V, Z \mid A = a$
             On $\mathcal{W}^s$, for $j = 1, ...n$ compute $\phi_{m,j}^s = \frac{\mathbb{I}\{A_j=a\}}{\hat{\pi}^q(V_j,Z_j)}((Y_j - \hat{\nu}_m^r(V_j))^2 - \hat{\eta}_m^q(V_j, Z_j)) + \hat{\eta}_m^q(V_j, Z_j)$
Output prediction models: $\hat{\nu}_{\text{DR}}(v) = \frac{1}{4}\sum_{j=1}^{4} \hat{\nu}_{\text{DR},j}(v))$;   $\hat{\nu}_{\text{PL}}(v) = \frac{1}{4}\sum_{j=1}^{4} \hat{\nu}_{\text{PL},j}(v)$ ;   $\hat{\nu}_{\text{TCR}}(v) = \frac{1}{4}\sum_{j=1}^{4} \hat{\nu}_{\text{TCR},j}(v)$
Output error estimate confidence intervals: for $m$ in { TCR, PL, DR }:
$\text{MSE}_m = \left(\frac{1}{4n}\sum_{i=1}^{4}\sum_{j=1}^{n} \phi_{m,j}^i\right) \pm 1.96\sqrt{\frac{1}{4n}\text{var}(\phi_m)}$
```

B Proofs and derivations

In this section we provided detailed proofs and derivations for all results in the main paper.

B.1 Derivation of Identifications of μ and ν

We first show the steps to identify $\mu(v, z)$:

$$\begin{aligned} \mu(v, z) &= \mathbb{E}[Y^a \mid V = v, Z = z] \\ \mathbb{E}[Y^a \mid V = v, Z = z] &= \mathbb{E}[Y^a \mid V = v, Z = z, A = a] \\ &= \mathbb{E}[Y \mid V = v, Z = z, A = a] \end{aligned}$$

The first line applies the definition of μ.. The second line follows from training ignorability (Condition 2.1.1). The third line follows from consistency (Condition 2.1.3).

Next we show the identification of $\nu(v)$:

$$\begin{aligned} \nu(v) &= \mathbb{E}[Y^a \mid V = v] \\ \mathbb{E}[Y^a \mid V = v] &= \mathbb{E}[\mathbb{E}[Y^a \mid V = v, Z = z] \mid V = v] \\ &= \mathbb{E}[\mathbb{E}[Y^a \mid V = v, Z = z, A = a] \mid V = v] \\ &= \mathbb{E}[\mathbb{E}[Y \mid V = v, Z = z, A = a] \mid V = v] \end{aligned}$$

The first line applies the definition of ν from Section 2. The second line follows from iterated expectation. The third line follows from training ignorability (Condition 2.1.1). The fourth line follows from consistency (Condition 2.1.3).

Note that we can concisely rewrite the last line as $\mathbb{E}[\mu(V,Z) \mid V=v]$ since we have identified μ.

B.2 Proof that TCR method underestimates risk under mild assumptions on a risk assessment setting

Proof. In Section 3.1 we posited that the TCR method will often underestimate risk in a binary risk assessment setting. We demonstrate this for the setting with a binary outcome $Y \in \{0,1\}$, but the logic extends to settings with a discrete or continuous outcome. We assume larger values of Y are adverse i.e. $Y=0$ is desired and $Y=1$ is adverse. We start by recalling runtime confounding condition (2.1.2): $\mathbb{P}(A=0 \mid V, Y^0=1) \neq \mathbb{P}(A=0 \mid V, Y^0=0)$. Here we further refine this by assuming we are in the common setting where treatment $A=1$ is more likely to be assigned to people who are higher risk. Then $\mathbb{P}(A=1 \mid V, Y^0=1) > \mathbb{P}(A=1 \mid V, Y^0=0)$. Equivalently $\mathbb{P}(A=0 \mid V, Y^0=1) < \mathbb{P}(A=0 \mid V, Y^0=0)$. By the law of total probability,

$$\mathbb{P}(A=0 \mid V) = \mathbb{P}(A=0 \mid V, Y^0=1)\mathbb{P}(Y^0=1 \mid V) + \mathbb{P}(A=0 \mid V, Y^0=0)\mathbb{P}(Y^0=0 \mid V)$$

Assuming $\mathbb{P}(Y^0=1 \mid V) > 0$, this implies

$$\mathbb{P}(A=0 \mid V, Y^0=0) > \mathbb{P}(A=0 \mid V) \tag{2}$$

By Bayes' rule,

$$\mathbb{P}(A=0 \mid V, Y^0=0) = \mathbb{P}(Y^0=0 \mid V, A=0)\frac{\mathbb{P}(A=0 \mid V)}{\mathbb{P}(Y^0=0 \mid V)}$$

Using this in the LHS of Equation 2 and dividing both sides of Equation 2 by $\mathbb{P}(A=0 \mid V)$, we get

$$\frac{\mathbb{P}(Y^0=0 \mid V, A=0)}{\mathbb{P}(Y^0=0 \mid V)} > 1$$

Equivalently $\mathbb{E}[Y^0 \mid V, A=0] < \mathbb{E}[Y^0 \mid V]$. □

B.3 Derivation of Proposition 3.1: confounding bias of the TCR method

We recall **Proposition 3.1**:
Under runtime confounding, a model that perfectly predicts $\omega(v)$ has pointwise confounding bias $b(v) = \omega(v) - \nu(v) =$

$$\int_{\mathcal{Z}} \mu(v,z)\Big(p(z \mid V=v, A=a) - p(z \mid V=v)\Big)dz \quad \neq \quad 0 \tag{3}$$

Proof. By iterated expectation and the definition of expectation we have that

$$\begin{aligned}\omega(v) &= \int_{\mathcal{Z}} \mathbb{E}[Y \mid V=v, Z=z, A=a]\; p(z \mid V=v, A=a)dz \\ &= \int_{\mathcal{Z}} \mu(v,z)p(z \mid V=v, A=a)dz\end{aligned}$$

In the identification derivation above we saw that $\nu(v) = \mathbb{E}[\mu(V, Z) \mid V = v]$. Using the definition of expectation, we can rewrite this as

$$= \int_{\mathcal{Z}} \mu(v, z) p(z \mid V = v) dz$$

Therefore the pointwise bias is

$$\omega(v) - \nu(v) = \int_{\mathcal{Z}} \mu(v, z) \Big(p(z \mid V = v, A = a) - p(z \mid V = v) \Big) dz \tag{4}$$

We can prove that this pointwise bias is non-zero by contradiction. Assuming the pointwise bias is zero, we have $\omega(v) = \nu(v) \implies Y^a \perp A \mid V = v$ which contradicts the runtime confounding condition 2.1.2. □

We emphasize that the confounding bias does not depend on the treatment effect. This approach is problematic whenever treatment assignment depends on Y^a to an extent that is not measured by V, even for settings with no treatment effect.

B.4 Proof of Proposition 3.2

Proof. We first decompose the pointwise error into the estimation error and the bias of the TCR method.

$$\begin{aligned}
\mathbb{E}[(\nu(v) - \hat{\nu}_{\text{TCR}}(v))^2] &= \mathbb{E}[\Big((\nu(v) - \omega(v)) + (\omega(v) - \hat{\nu}_{\text{TCR}}(v)) \Big)^2] \\
&\leq 2 \Bigg(\mathbb{E}[(\nu(v) - \omega(v))^2] + \mathbb{E}[(\omega(v) - \hat{\nu}_{\text{TCR}}(v))^2] \Bigg) \\
&\lesssim (\nu(v) - \omega(v))^2 + \mathbb{E}[(\omega(v) - \hat{\nu}_{\text{TCR}}(v))^2] \\
&= b(v)^2 + \mathbb{E}[(\omega(v) - \hat{\nu}_{\text{TCR}}(v))^2]
\end{aligned}$$

Where the second line is due to the fact that $(a + b)^2 \leq 2(a^2 + b^2)$. In the third line, we drop the expectation on the first term since there is no randomness in two fixed functions of v. □

B.5 Proofs of Proposition 3.3 and Theorem 3.1

We begin with additional notation needed for the proofs of the error bounds. For brevity let $W = (V, Z, A, Y)$ indicate a training observation. The theoretical guarantees for our methods rely on a two-stage training procedure that assumes independent training samples. We denote the first-stage training dataset as $\mathcal{W}^1 := \{W_1^1, W_2^1, W_3^1, ... W_n^1\}$ and the second-stage training dataset as $\mathcal{W}^2 := \{W_1^2, W_2^2, W_3^2, ... W_n^2\}$. Let $\hat{\mathbb{E}}_n[Y \mid V = v]$ denote an estimator of the regression function $\mathbb{E}[Y \mid V = v]$. Let $L \asymp R$ denote $L \lesssim R$ and $R \lesssim L$.

Definition B.1. (Stability conditions) The results assume the following two stability conditions from [14] on the second-stage regression estimators:

Condition B.1.1. $\hat{\mathbb{E}}_n[Y \mid V = v] + c = \hat{\mathbb{E}}_n[Y + c \mid V = v]$ for any constant c

Condition B.1.2. For two random variables R and Q, if $\mathbb{E}[R \mid V = v] = \mathbb{E}[Q \mid V = v]$, then

$$\mathbb{E}\left[\Big(\hat{\mathbb{E}}_n[R \mid V = v] - \mathbb{E}[R \mid V = v] \Big)^2 \right] \asymp \mathbb{E}\left[\Big(\hat{\mathbb{E}}_n[Q \mid V = v] - \mathbb{E}[Q \mid V = v] \Big)^2 \right]$$

B.5.1 Proof of Proposition 3.3

The theoretical results for our two-stage procedures rely on the theory for pseudo-outcome regression in Kennedy [14] which bounds the error for a two-stage regression on the full covariates. However, our setting is different since our second-stage regression is on a subset of the full covariates. Therefore, Theorem 1 of Kennedy [14] does not immediately give the error bound for our setting, but we can follow the same approach in order to get the bound for our V-conditional second-stage estimators.

Proof. The first step is to derive the error function for our setting. For the PL approach the error function is defined as $\hat{\hat{r}}_{\text{PL}}(v)$

$$
\begin{aligned}
&= \mathbb{E}[\hat{\mu}(V,Z) \mid V = v, \mathcal{W}^1] - \nu(v) \\
&= \mathbb{E}[\hat{\mu}(V,Z) \mid V = v, \mathcal{W}^1] - \mathbb{E}[\mu(V,Z) \mid V = v] \\
&= \mathbb{E}[\hat{\mu}(V,Z) - \mu(V,Z) \mid V = v, \mathcal{W}^1]
\end{aligned}
$$

The first line is our definition of the error function (following [14]). The second line uses iterated expectation, and the third lines uses the fact that $\mathcal{W}^1$ is a random sample of the training data. Next we square the error function and apply Jensen's inequality to get

$$
\hat{r}_{\text{PL}}(v)^2 = \Big(\mathbb{E}[\hat{\mu}(V,Z) - \mu(V,Z) \mid V = v, \mathcal{W}^1]\Big)^2 \leq \mathbb{E}\Big[\Big(\hat{\mu}(V,Z) - \mu(V,Z)\Big)^2 \mid V = v, \mathcal{W}^1\Big]
$$

Taking the expectation over $\mathcal{W}^1$ on both sides, we get

$$
\begin{aligned}
\mathbb{E}[\hat{r}_{\text{PL}}(v)^2 \mid V = v] &\leq \mathbb{E}\left[\mathbb{E}\Big[\Big(\hat{\mu}(V,Z) - \mu(V,Z)\Big)^2 \mid V = v, \mathcal{W}^1\Big] \mid V = v\right] \\
&= \mathbb{E}\left[\Big(\hat{\mu}(V,Z) - \mu(V,Z)\Big)^2 \mid V = v\right]
\end{aligned}
$$

Next, under our stability conditions(§ B.1), we can apply Theorem 1 of Kennedy [14] (stated in the next section for reference) to get the pointwise bound

$$
\mathbb{E}\left[\Big(\hat{\nu}_{\text{PL}}(v) - \nu(v)\Big)^2\right] \lesssim \mathbb{E}\left[\Big(\tilde{\nu}(v) - \nu(v)\Big)^2\right] + \mathbb{E}\left[\Big(\hat{\mu}(V,Z) - \mu(V,Z)\Big)^2 \mid V = v\right]
$$

Theorem 1 of Kennedy also implies a bound on the integrated MSE of the PL approach:

$$
\mathbb{E}\,\|\hat{\nu}_{\text{PL}}(v) - \nu(v)\|^2 \lesssim \mathbb{E}\,\|\tilde{\nu}(v) - \nu(v)\|^2 + \int_{\mathcal{V}} \mathbb{E}\Big[(\hat{\mu}(V,Z) - \mu(V,Z))^2 \mid V = v\Big] p(v) dv
$$

□

B.5.2 Theorem for Pseudo-Outcome Regression (Kennedy)

The proofs of Proposition 3.3 and Theorem 3.1 rely on Theorem 1 of Kennedy [14] which we restate here for reference. In what follows we provide the proof for Theorem 3.1.

Theorem B.1 (Kennedy)**.** Recall that $\mathcal{W}^1$ denotes our n first-stage training data samples. Let $\hat{f}(w) := \hat{f}(w; \mathcal{W}^1)$ be an estimate of the function $f(w)$ using the training data $\mathcal{W}^1$. Denote an independent sample as W. The true regression function is $m(v) := \mathbb{E}[f(W) \mid V = v]$. Denote the second stage regression as $\hat{m}(v) := \hat{\mathbb{E}}_n[\hat{f}(W) \mid V = v]$. Denote its oracle equivalent (if we had access to Y^a) as $\tilde{m}(v) := \hat{\mathbb{E}}_n[f(W) \mid V = v]$. Under stability conditions(§ B.1) on the regression estimator $\hat{\mathbb{E}}_n$, we have the following bound on the pointwise MSE:

$$\mathbb{E}\left[\Big(\hat{m}(v) - m(v)\Big)^2\right] \lesssim \mathbb{E}\left[\Big(\tilde{m}(v) - m(v)\Big)^2\right] + \mathbb{E}\left[\hat{r}(v)^2\right]$$

where $\hat{r}(v)$ describes the error function $\hat{r}(v) := \mathbb{E}[\hat{f}(W) \mid V = v, \mathcal{W}^1] - m(v)$. This implies the following bound for the integrated MSE:

$$\mathbb{E}\left\|\hat{m}(v) - m(v)\right\|^2 \lesssim \mathbb{E}\left\|\tilde{m}(v) - m(v)\right\|^2 + \int \mathbb{E}\big[\hat{r}(v)^2\big]p(v)dv$$

B.5.3 Proof of Theorem 3.1

Here we provide the proof for our main theoretical result which bounds the error of our proposed DR method.

Proof. As for the PL error bound above, the first step is to derive the form of the error function for our DR approach. For clarity and brevity, we denote the measure of the expectation in the subscript.

$$\begin{aligned}
\hat{r}_{\mathrm{DR}}(v) &= \mathbb{E}_{W|V=v,\mathcal{W}^1}\left[\frac{\mathbb{I}\{A=a\}}{\hat{\pi}(v,Z)}(Y - \hat{\mu}(v,Z)) + \hat{\mu}(v,Z)\right] - \nu(v) \\
&= \mathbb{E}_{Z,A|V=v,\mathcal{W}^1}\left[\mathbb{E}_{W|A=a,V=v,Z=z,\mathcal{W}^1}\left[\frac{\mathbb{I}\{A=a\}}{\hat{\pi}(v,Z)}(Y - \hat{\mu}(v,z)) + \hat{\mu}(v,z)\right]\right] - \nu(v) \\
&= \mathbb{E}_{Z,A|V=v,\mathcal{W}^1}\left[\mathbb{E}_{Y|A=a,V=v,Z=z,\mathcal{W}^1}\left[\frac{\mathbb{I}\{A=a\}}{\hat{\pi}(v,Z)}(Y - \hat{\mu}(v,z))\right] + \hat{\mu}(v,Z)\right] - \nu(v) \\
&= \mathbb{E}_{Z,A|V=v,\mathcal{W}^1}\left[\frac{\mathbb{I}\{A=a\}}{\hat{\pi}(v,Z)}(\mathbb{E}_{Y|A=a,V=v,Z=z,\mathcal{W}^1}[Y] - \hat{\mu}(v,Z)) + \hat{\mu}(v,Z)\right] - \nu(v) \\
&= \mathbb{E}_{W|V=v,\mathcal{W}^1}\left[\frac{\mathbb{I}\{A=a\}}{\hat{\pi}(v,Z)}(\mu(v,Z) - \hat{\mu}(v,Z)) + \hat{\mu}(v,Z)\right] - \nu(v) \\
&= \mathbb{E}_{Z|V=v,,\mathcal{W}^1}\left[\mathbb{E}_{W|V=v,Z=z,\mathcal{W}^1}\left[\frac{\mathbb{I}\{A=a\}}{\hat{\pi}(v,Z)}(\mu(v,z) - \hat{\mu}(v,z)) + \hat{\mu}(v,z)\right]\right] - \nu(v) \\
&= \mathbb{E}_{Z|V=v,\mathcal{W}^1}\left[\frac{\mathbb{P}(A=a \mid V=v, Z=z)}{\hat{\pi}(v,Z)}(\mu(v,Z) - \hat{\mu}(v,Z)) + \hat{\mu}(v,Z)\right] - \nu(v) \\
&= \mathbb{E}_{Z|V=v,\mathcal{W}^1}\left[\frac{\pi(v,Z)}{\hat{\pi}(v,Z)}(\mu(v,Z) - \hat{\mu}(v,Z)) + \hat{\mu}(v,Z)\right] - \nu(v) \\
&= \mathbb{E}_{Z|V=v,\mathcal{W}^1}\left[\frac{\pi(v,Z)}{\hat{\pi}(v,Z)}(\mu(v,Z) - \hat{\mu}(v,Z)) + \hat{\mu}(v,Z) - \mu(v,Z)\right] \\
&= \mathbb{E}\left[\frac{(\mu(v,Z) - \hat{\mu}(v,Z))(\pi(v,Z) - \hat{\pi}(v,Z))}{\hat{\pi}(v,Z)} \mid V = v, \mathcal{W}^1\right]
\end{aligned}$$

Where the first line holds by definition of the error function $\hat{r}$ and the second line by iterated expectation. The third line uses the fact that conditional on $Z = z, V = v, A = a$, then the only randomness in W is Y (and therefore $\hat{\mu}$ is constant). The fourth line makes use of the ($\mathbb{I}\{A = a\}$) term to allow us to condition on only $A = a$ (since the term conditioning on any other $a' \neq a$ will evaluate to zero). The fifth line applies the definition of μ.
The sixth line again uses iterated expectation and the seventh makes use of the fact that conditional on Z, the only randomness now is in A and that $\mathcal{W}^1$ is an independent randomly sampled set. The seventh line applies the definition of $\pi(v, z) = \mathbb{P}(A = 1 \mid V = v, Z = z)$ which since $A \in \{0, 1\}$ is equal to $\mathbb{E}[A \mid V = v, Z = z]$. The eight line uses iterated expectation and the fact that $\mathcal{W}^1$ is an independent randomly sampled set to rewrite $\nu(v) = E_{Z|V=v,\mathcal{W}^1}[\mu(v, Z)]$. The ninth line rearranges the terms.

By Cauchy-Schwarz and the positivity assumption,

$$\hat{r}_{\mathrm{DR}}(v) \leq C\sqrt{\mathbb{E}[(\mu(v, Z) - \hat{\mu}(v, Z))^2 \mid V = v, \mathcal{W}^1]}\sqrt{\mathbb{E}[(\pi(v, Z) - \hat{\pi}(v, Z))^2 \mid V = v, \mathcal{W}^1]}$$

for a constant C.

Squaring both sides yields

$$\hat{r}^2_{\mathrm{DR}}(v) \leq C^2 \ \mathbb{E}[(\mu(v, Z) - \hat{\mu}(v, Z))^2 \mid V = v, \mathcal{W}^1] \ \mathbb{E}[(\pi(v, Z) - \hat{\pi}(v, Z))^2 \mid V = v, \mathcal{W}^1]$$

If $\hat{\pi}$ and $\hat{\mu}$ are estimated using separate training samples, then taking the expectation over the first-stage training sample $\mathcal{W}^1$ yields:

$$\mathbb{E}[\hat{r}^2_{\mathrm{DR}}(v)] \leq C^2 \ \mathbb{E}[(\mu(v, Z) - \hat{\mu}(v, Z))^2] \mid V = v] \ \mathbb{E}[(\pi(v, Z) - \hat{\pi}(v, Z))^2] \mid V = v]$$

Applying Theorem 1 of Kennedy [14] gets the pointwise bound:

$$\begin{aligned}\mathbb{E}\left[\Big(\hat{\nu}_{\mathrm{DR}}(v) - \nu(v)\Big)^2\right] &\lesssim \mathbb{E}\left[\Big(\tilde{\nu}(v) - \nu(v)\Big)^2\right] \\ &\quad + \mathbb{E}\Big[(\hat{\pi}(V, Z) - \pi(V, Z))^2 \mid V = v\Big]\mathbb{E}\Big[(\hat{\mu}(V, Z) - \mu(V, Z))^2 \mid V = v\Big]\end{aligned}$$

and the bound on integrated MSE of the DR approach:

$$\begin{aligned}&\mathbb{E}\left\|\hat{\nu}_{\mathrm{DR}}(v) - \nu\right\|^2 \lesssim \ \mathbb{E}\left\|\tilde{\nu}(v) - \nu(v)\right\|^2 \\ &+ \int_{\mathcal{V}} \mathbb{E}\Big[(\hat{\pi}(V, Z) - \pi(V, Z))^2 \mid V = v\Big]\mathbb{E}\Big[(\hat{\mu}(V, Z) - \mu(V, Z))^2 \mid V = v\Big]p(v)dv\end{aligned}$$

□

B.6 Efficient influence function for DR method

We provide the efficient influence function of the DR method. The efficient influence function indicates the form of the bias-correction term in the DR method. The efficient influence function $\phi(A, V, Z, Y)$ for parameter $\psi(V) := \mathbb{E}[Y^a \mid V] = \mathbb{E}[\mathbb{E}[Y \mid V, Z, A = a] \mid V]$ is

$$\phi(A, V, Z, Y) = \frac{\mathbb{I}\{A = a\}}{\pi(V, Z)}(Y - \mu(V, Z)) + \mu(V, Z) - \psi(V)$$

C Experimental details and additional results

In this section we present details on the experiments and present additional results. We present the random forests graphs omitted from the main paper, results on calibration-type curves that show where the errors are distributed, and experiments on our evaluation procedure.

C.1 Experimental details

More details on data-generating process We designed our data-generating process in order to simulate a real-world risk assessment setting. We consider both V and Z to be risk factors whose larger values indicate increased risk and therefore we construct μ to increase with V and Z. Our goal is to assess risk under the null (or baseline) treatment as per [8], and we construct π such that historically the riskier treatments were more likely to get the risk-mitigating treatment and the less risky cases were more likely to get the baseline treatment.

We now provide further details on the choices of coefficients and variance parameters. In the first set of experiments presented in the main paper, we simulate V_i from a standard normal, and in the uncorrelated setting (where $\rho = 0$) we also simulate Z_i from a standard normal. In the correlated setting, we sample Z_i from a normal with mean ρV_i and variance $1 - \rho^2$ so that the Pearson's correlation coefficient between V_i and Z_i is ρ and so that the variance in $Z_i = 1$. We simulate μ to be a sparse linear model in V and Z with coefficients of 1 when $\rho = 0$. When $\rho \neq 1$, the coefficients are set to $\frac{k_v}{k_v + \rho k_z}$ so that the L_1 norm of the ν coefficients equals k_v for all values of ρ. Without this adjustment, changing ρ would impact error by also changing the signal-to-noise ratio in ν. We simulate the potential outcome Y^a to be conditionally Gaussian and the choice of variance $\frac{1}{2n} \|\mu(V, Z)\|_2^2$ yields a signal-to-noise ratio of 2. The specification for ν follows from the marginalization of μ over Z. The propensity score π depends on the sigmoid of a sparse linear function in V and Z that uses coefficients $\frac{1}{\sqrt{k_v + k_z}}$ in order to satisfy our positivity condition.

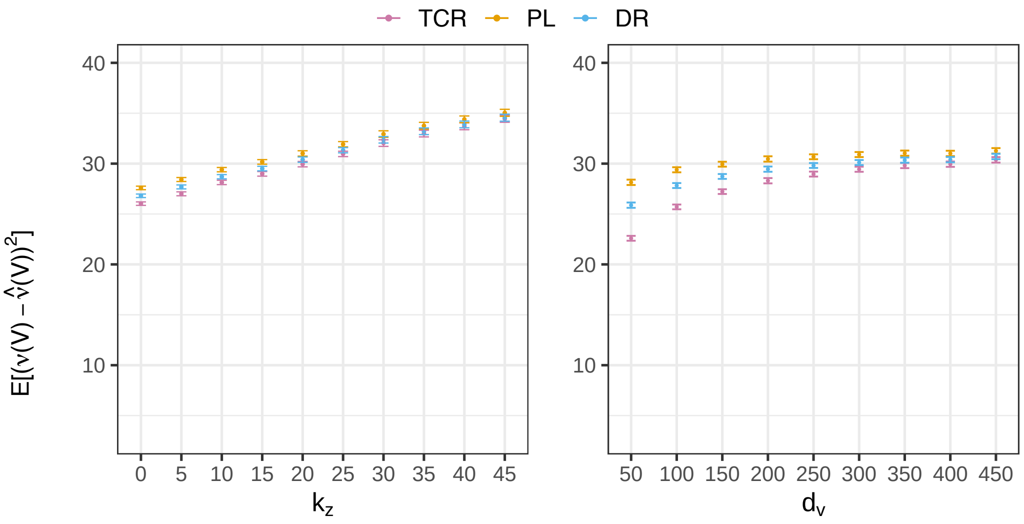
We use $d = 500$, $n = 1000$, $k_v = 25$, and $0 \leq k_z \leq 45$ to simulate a sparse high-dimensional setting with many measured variables in the training data, of which only 5%-15% are predictive of the outcomes. In one set of experiments, we vary the value of k_z to assess impact of various levels of confounding on performance. In other experiments, where we vary ρ or the dimensionality of V (d_{V}), we use $k_z = 20$ so that V has slightly more predictive power than the hidden confounders Z.

Hyperparameters Our LASSO presents are presented for cross-validated hyperparameter selection using the `glmnet` package in R. The random forests results use 1000 trees and default *mtry* and splitting parameters in the `ranger` package in R.

Training runs Defining a training run as performing a learning procedure such as LASSO, for a given hyperparameter selection and given simulation, the TCR method trains in one run, the PL method trains in two runs, and the DR method trains in three runs. For a given simulation, the exact number of runs depends on the hyperparameter tuning. Since we only ran random forests (RF) for the default parameters, the TCR method with RF trained in one run, the PL method with RF trained in two runs, and the DR method with RF trained in three runs. The LASSO results using `cv.glmnet` were tuned over ≤ 100 values of λ; the TCR method with LASSO trained in ≤ 100 runs, the PL method with LASSO trained in ≤ 200 runs, and the DR method with LASSO trained in ≤ 300 runs.

Sample size and error metrics For experiments in the main paper, we trained on $n = 1000$ datapoints. We test on a separate set of $n = 1000$ datapoints and report the estimated mean squared error (MSE) on this test set using the following formula:

$$\frac{1}{n} \sum_{i=1}^{n} (\nu(V_i) - \hat{\nu}(V_i))^2$$



(a) Random forests

(b) Random forests

Figure 3: **(a)** MSE as we vary k_z using random forests to learn $\hat{\pi}$, $\hat{\mu}$, $\hat{\nu}_{\mathrm{TCR}}$, $\hat{\nu}_{\mathrm{PL}}$, $\hat{\nu}_{\mathrm{DR}}$ for $\rho = 0$, $d_{\mathrm{V}} = 400$ and $k_v = 25$.
(b) MSE against d_{V} using random forests and $\rho = 0$, $k_v = 25$ and $k_z = 20$.
Error bars denote 95% confidence intervals.

Computing infrastructure We ran experiments on an Amazon Web Serivces (AWS) c5.12xlarge machine. This parallel computing environment was useful because we ran thousands of simulations. The traintime of each simulation, entailing the LASSO and RF experiments, took 1.8 seconds. In practice for most real-world decision support settings, our method can be used in standard computing environments; relative to existing predictive modeling techniques, our method will require $\leq 3X$ the current train time. Our runtime depends only on the regression technique used in the second stage and should be competitive to existing models.

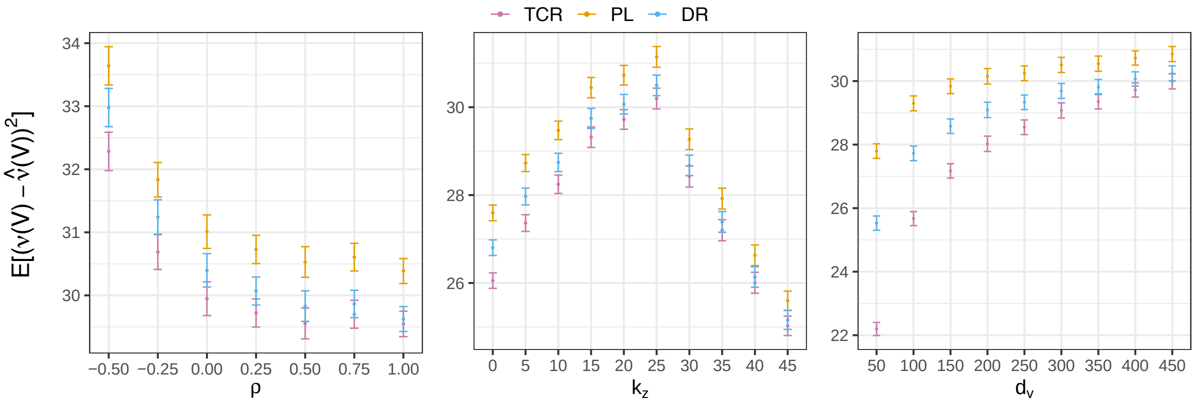
C.2 Random forest results

Figure 3 presents the results when using random forests for the first and second stage estimation in the uncorrelated V-Z setting. Figure 3a was provided in the main paper, and we include it here again for ease of reference. Figure 3b shows how method performance varies with d_{V}. At low d_{V}, the TCR method does significantly better than the two counterfactually valid approaches. This suggests that the estimation error incurred by the PL and DR methods outweighs the confounding bias of the TCR method.

C.3 Evaluation experiments

To empirically assess our proposed doubly-robust evaluation procedure, we generated one sample of training data with $n = 1000$, $d = 500$, $d_{\mathrm{V}} = 200$, $k_v = 25$, and $k_z = 30$ as well as a "ground-truth" test set with $n = 10,000$. We trained the TCR, PL, and DR methods on the training data and estimated their true performance on the large test set. The true prediction error

$$\frac{1}{n}\sum_{i=1}^{n}\left(Y_i^a - \hat{\nu}(V_i)\right)^2$$



(a) Random forests

(b) Random forests

(c) Random forests

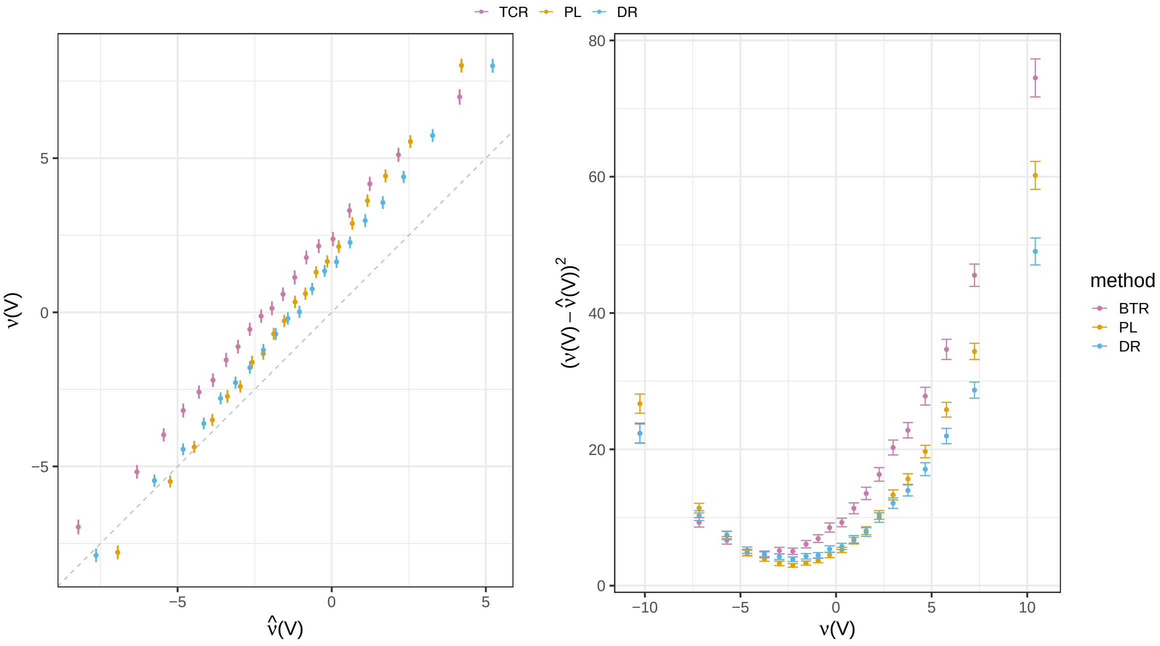
Figure 4: **(a)** MSE against correlation ρ_{V_i,Z_i} for $k_z = 20$, $k_v = 25$, and $d_{\rm V} = 400$. For all methods, error decreases with $\rho \leq 0.5$, at which point the error does not change with increasing ρ. **(b)** MSE as we increase k_z for $\rho = 0.25$, $k_v = 25$, and $d_{\rm V} = 400$. Compare to Figure 3a; the weak positive correlation reduces MSE, particularly for $k_v < i \leq k_z$ when V_i is only a correlate for the confounder Z_i but not a confounder itself. **(c)** MSE against $d_{\rm V}$ for $\rho = 0.25$, $k_z = 20$, and $k_v = 25$. As with the uncorrelated setting (3b), the DR and TCR methods are better able to take advantage of low $d_{\rm V}$ than the PL method.
Error bars denote 95% confidence intervals.

was 77.53, 74.12, and 72.68 respectively for the TCR, PL and DR methods. We then ran 100 simulations where we sampled a more realistically sized test set of $n = 2000$. In each simulation we performance the evaluation procedure to estimate prediction error on the observed data. 81% of the simulations correctly identified the DR procedure as having the lowest error, 14% suggested that the PL procedure had the lowest error and 5% suggested that the TCR had the lowest error.

For additional experimental results on using doubly-robust evaluation methods for predictive models, we recommend [8].

C.4 Calibration-styled analysis of the error

Above we analytically showed that in a standard risk assessment setting the TCR method underestimates risk. We empirically demonstrate this in Figure 5 where the calibration curve (Figure 5a) shows that TCR underestimates risk for all predicted values. Figure 5b plots the squared error against true risk $\nu(V)$, illustrating that errors are extremely large for high-risk individuals, particularly for the TCR model. This highlights a danger in using confounded approaches like the TCR model: they make misleading predictions about the highest risk cases. In high-stakes settings like child welfare screening, this may result in dangerously deciding to *not* investigate the cases where the child is at high risk of adverse outcomes [8]. The counterfactually valid PL and DR models mitigate this to some effect, but future work should investigate why the errors are still large on high-risk cases and propose procedures to further mitigate this.



(a) LASSO

(b) LASSO

Figure 5: (a) Calibration plot for LASSO regressions with $p = 400$, $q = 100$, $k_z = 20$ and $k_v = 25$. A well-calibrated model will track the dotted $y = x$ line. Our DR model is the best calibrated. As expected from its confounding bias, the TCR method underestimates risk for all predicted values. Interestingly the PL and DR methods also underestimate risk for higher predicted risk values.
(b) Squared error against true risk $\nu(V)$ for LASSO regressions with $p = 400$, $q = 100$, $k_z = 20$ and $k_v = 25$. All models have highest error on the riskiest cases (those with large values of $\nu(V)$); this is particularly pronounced for the TCR model, suggesting that the TCR model would make misleading predictions for the highest risk cases.