Stabilized direct learning for efficient estimation of individualized treatment rules

Kushal S. Shah¹ | Haoda Fu² | Michael R. Kosorok¹

¹Department of Biostatistics, University of North Carolina at Chapel Hill, North Carolina, USA
²Eli Lilly and Company, Lilly Corporate Center, Indianapolis, USA

Correspondence
Michael R. Kosorok, Department of Biostatistics, University of North Carolina at Chapel Hill, NC 27599, USA.
Email: kosorok@unc.edu

Abstract
In recent years, the field of precision medicine has seen many advancements. Significant focus has been placed on creating algorithms to estimate individualized treatment rules (ITRs), which map from patient covariates to the space of available treatments with the goal of maximizing patient outcome. Direct learning (D-Learning) is a recent one-step method which estimates the ITR by directly modeling the treatment–covariate interaction. However, when the variance of the outcome is heterogeneous with respect to treatment and covariates, D-Learning does not leverage this structure. Stabilized direct learning (SD-Learning), proposed in this paper, utilizes potential heteroscedasticity in the error term through a residual reweighting which models the residual variance via flexible machine learning algorithms such as XGBoost and random forests. We also develop an internal cross-validation scheme which determines the best residual model among competing models. SD-Learning improves the efficiency of D-Learning estimates in binary and multi-arm treatment scenarios. The method is simple to implement and an easy way to improve existing algorithms within the D-Learning family, including original D-Learning, Angle-based D-Learning (AD-Learning), and Robust D-learning (RD-Learning). We provide theoretical properties and justification of the optimality of SD-Learning. Head-to-head performance comparisons with D-Learning methods are provided through simulations, which demonstrate improvement in terms of average prediction error (APE), misclassification rate, and empirical value, along with a data analysis of an acquired immunodeficiency syndrome (AIDS) randomized clinical trial.

KEYWORDS
D-Learning, heteroscedasticity, individualized treatment rule, multi-arm treatments, precision medicine, statistical machine learning

1 | INTRODUCTION

Precision medicine is a framework at the intersection of statistics, machine learning, and causal inference, for leveraging patient heterogeneity to improve patient outcomes. Individualized treatment rules (ITR), attempting to maximize expected treatment benefit across a population, may recommend personalized actions based on patient demographic information, clinical biomarkers, or genetic data. One of the primary goals of precision medicine
has been formalized as decision support—the estimation of optimal and near-optimal regimes (Kosorok & Laber, 2019). In this vein, it is important to develop algorithms which work efficiently with the data at hand.

In recent years, an extensive literature has been developed in the area of estimating optimal ITRs. Traditionally, algorithms have fallen into one of the two categories: model-based versus policy-search approaches. Model-based approaches may also be considered “regression-based” or “indirect” in that they first model the conditional response of interest and then invert the relationship between patient covariates, treatment, and outcome to estimate an optimal rule (Kosorok & Moodie, 2016). Primary examples of model-based approaches are Q-Learning (Qian & Murphy, 2011) and A-Learning (Murphy, 2003). Q-Learning approaches model the outcome conditional on covariates, whereas A-Learning approaches model regret functions or contrast functions between treatments (Schulte et al., 2014). Policy-search approaches, on the other hand, maximize value functions directly instead of modeling the conditional mean. For this reason, they are also known as “value-search” or “direct-search” (Kosorok & Moodie, 2016). By sidestepping the modeling step and directly searching for an optimal rule among a class of policies, they may avoid model misspecification (Xiao et al., 2019). Many policy-search approaches have reframed the maximization of clinical outcomes as a weighted classification problem with the goal of minimizing weighted classification error, including the outcome-weighted learning (OWL) family of methods (Zhou et al., 2017; Zhang et al., 2020) and various tree-based extensions (Cui et al., 2017; Kallus, 2018).

A third category of algorithms to estimate ITRs also exists, consisting of methods which have attempted to combine the advantages of model-based and policy-search approaches. Often, these methods use the augmented inverse probability weighted estimator (AIPWE) within the classification framework. Such approaches are “doubly robust” in the sense that they enjoy greater protection against model misspecification and increased efficiency when both models are correctly specified (Zhang & Zhang, 2018; Zhao et al., 2019).

Tian et al. (2014) developed a “modified-covariate” approach for ITR estimation, which falls into the third category but is unique in that it maintains the regression-based framework to model the treatment–covariate interaction effect directly, without having to specify a main effect model or conditional mean outcome function. Later, Qi and Liu (2018) coined the term “Direct Learning” (D-Learning) for the Tian et al. (2014) method and extended it to nonlinear decision rules and multi-arm treatment settings. Key extensions to D-Learning are Angle-based Direct Learning (AD-Learning) (Qi et al., 2020), which improves D-Learning in the multi-arm treatment case, and Robust Direct Learning (RD-Learning) (Meng & Qiao, 2022), which achieves a double robustness property.

Now consider, for example, the “AIDS Clinical Trial Group Study 175” (ACTG175), a randomized clinical trial (RCT) which compared the effectiveness of four treatments in increasing CD4 cell counts in HIV-1 patients (Hammer et al., 1996). Previous studies have suggested that the response of change in CD4 cell count from these data may have skewed, heteroscedastic errors Xiao et al. (2019), Zhang et al. (2021), which we confirm in Section 5. In such a situation, when the variance of the clinical outcome is a function of the covariates (or treatment), the D-Learning family of estimators remains consistent for the optimal ITR, but gives each observation equal weight by default in model training. A reweighting approach which utilizes this error structure to prioritize observations with smaller expected outcome variance is beneficial because it can attain greater efficiency when estimating an ITR. This example motivates the ITR estimation approach of this paper.

In this paper, we propose Stabilized D-learning (SD-Learning), a method to increase the efficiency of D-Learning estimates in situations, where the variance of the error term is non-homogeneous and a function of the treatment and covariates. SD-Learning can be viewed as a special case of the framework of Liang and Yu (2020) with a single-index model. Liang and Yu (2020) found the efficient score for a semiparametric, and hence general, class of estimators of the decision function, but the estimation procedure does not lead to optimality. The SD-Learning methodology specializes on a smaller class of decision rules and achieves optimality within that class. These differences are highlighted in Section 2. From another perspective, SD-Learning may be considered an adaptation of feasible-weighted least squares (FWLS; Olive, 2017) to the precision medicine setting, where optimal ITR estimation with two or more treatments is of interest. Similarly to FWLS, SD-Learning is motivated by efficient estimation and controls on variance. We contribute to the existing literature in the following ways:

1. We bring the work of Tian et al. (2014), Qi and Liu (2018), Qi et al. (2020), and Meng and Qiao (2022) into a single framework such that the estimated parameters from either of these methods can be improved by a single-iteration update.

2. Our method applies concepts from weighted least squares (WLS) theory to increase the precision of ITR parameter estimation under heteroscedasticity. This entails a residual reweighting framework, where
residual variance is modeled through flexible machine learning methods. We develop an internal cross-validation scheme allowing for selection of an optimal model among methods such as XGBoost (Chen & Guestrin, 2016) and random forests (Breiman, 2001).

(3) We allow for even the multiple-treatment scenario ($K \geq 3$ treatments) to fit into the least-squares framework through a vectorization approach. As a result of this, parameter estimation has a simple implementation leading to a closed-form solution; hence, the algorithm is efficient and does not require iterative optimization techniques to solve.

(4) We show that SD-Learning parameter estimates are consistent, asymptotically normal in binary and multi-arm treatment scenarios under heterogeneous error, have greater efficiency than D-Learning estimates, and establish value function convergence bounds.

The rest of this paper is organized as follows. In Section 2, we introduce the methodology of SD-Learning. Specifically, Section 2.1 reviews recent developments in D-Learning and Section 2.2 introduces the mathematical motivation behind SD-Learning and outlines the reweighting solution. Section 2.3 extends the reweighting solution to scenarios with multi-arm treatments. In Section 2.4, the residual model fitting step of the method is described in greater detail, and a stepwise implementation of the method is delineated. In Section 3, theoretical results for SD-Learning including consistency, asymptotic normality, asymptotic efficiency, and value bounds are established for binary and multi-arm settings. Head-to-head simulations comparing SD-Learning to D-Learning, AD-Learning, and RD-Learning based on average prediction error (APE), misclassification rate, and empirical value are provided in Section 4, and value comparisons from analysis of the ACTG175 RCT data are made in Section 5. Concluding discussions and areas for future work are presented in Section 6.

2 | STABILIZED DIRECT LEARNING (SD-LEARNING)

Although SD-Learning works with observational data, for simplicity, we first consider an RCT setting to demonstrate the methodology. For $n$ patients, we observe independent realizations of the random triplet $(X, A, R)$. Patient covariates are represented by the $p$-dimensional vector $X \in \mathcal{X} \subset \mathbb{R}^p$, which includes an intercept. We start with the binary treatment scenario, $A \in \mathcal{A} = \{-1, 1\}$. Clinical outcome is represented by $R \in \mathbb{R}$, and it is assumed, without loss of generality, that larger $R$ corresponds to the better outcome. The probability of receiving treatment $a$, given covariates $x$, is represented by $\pi(a, x) = \text{Pr}(A = a | X = x)$. An ITR, $d(X) : \mathcal{X} \mapsto \mathcal{A}$, is a mapping from covariates to treatments. Let $\mathbb{1}(\cdot)$ represent the indicator function, $Z^T$ denote the transpose of matrix $Z$, and $P_n(\cdot)$ represent empirical average (e.g., $P_n(X) = n^{-1} \sum_{i=1}^n x_i$, where $x_1, ..., x_n$ are realizations of the random variable, $X$). Let $\text{Vec}(Z)$ represent vectorization (e.g., for matrix $Z = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$, $\text{Vec}(Z) = [a \ b \ c \ d]^T$).

Let $R^*(-1)$ and $R^*(1)$ represent potential outcomes that would have been observed had a patient received treatment $-1$ or $1$, respectively. From the framework of Rubin (1974), we make the usual assumptions for the precision medicine context (Kosorok & Moodie (2016), Hernán & Robins (2019), Kosorok & Laber (2019)): (i) stable unit treatment value assumption (SUTVA); $R = R^*(A)$, (ii) no unmeasured confounding: $A \perp \{R^*(-1), R^*(1)\} | X$, and (iii) positivity: $\pi(A, X) > c > 0, \forall A \in \mathcal{A}, X \in \mathcal{X}$. Prior to outlining the proposed SD-Learning methodology, we review key findings from the D-Learning family of methods.

2.1 | D-Learning background

It is known from Qian and Murphy (2011) that the expected response under an ITR, $d$, can be represented by the value function:

$$V(d) = E[R | A = d(X)] = E \left[ \frac{R \cdot \mathbb{1}[A = d(X)]}{\pi(A, X)} \right], \quad (1)$$

and we define an optimal ITR, $d^\text{opt}$, as the decision rule that maximizes the expected average response: $d^\text{opt}(\cdot) = \text{argmax}_{d \in D} V(d)$, where $D$ is a prespecified class of decision rules. Using the potential outcomes notation, $V(d) = E[R^*(d)] = \sum_{a \in \{-1, 1\}} E[R^*(a)]P[d(X) = a]$, representing the counterfactual population mean outcome under the ITR, $d$.

2.1.1 | D-Learning

In the two-arm setting, assume that the outcome can be expressed by:

$$R = m(X) + \delta(X)A + \eta, \quad (2)$$

where $m(X)$ and $\delta(X)$ are measurable functions representing the main and interaction effects, respectively, and $\eta$ is a mean-zero random error term. Equation (2) is a general multivariate regression setup for characterizing
interactions between treatment and covariates. Note the following:

\[ d^{\text{opt}}(X) = \text{sign}\{E(R|X, A = 1) - E(R|X, A = -1)\} \]

\[ = \text{sign}\{f^{\text{opt}}(X)\}, \quad (3) \]

\[ f^{\text{opt}}(X) = E\left\{ \frac{RA}{\pi(A, X)}|X\right\} = 2\delta(X). \quad (4) \]

Due to SUTVA and the no unmeasured confounders assumptions, \( f^{\text{opt}}(X) \) in Equation (3) is causally interpreted as the conditional average treatment effect (CATE), as outlined in Equation (2.3) of Jacob (2021). Positivity is necessary for \( d^{\text{opt}} \) to be optimal, as it ensures that every covariate–treatment combination has positive probability of being observed (Kosorok & Moodie, 2016; Schulte et al., 2014). Since the CATE is a contrast between the effects of two treatments (−1 and 1), the link between \( f^{\text{opt}}(X) \) and \( \delta(X) \) is intuitive because \( \delta(X)(1) - \delta(X)(-1) = 2\delta(X) \).

Tian et al. (2014) made the connection between the optimal ITR in Equation (3) and formulation of the optimal decision function in Equation (4), which forms the basis of D-Learning, as \( f^{\text{opt}}(X) \) can now be directly learned through a regression method of choice. Lemma 1 of Qi and Liu (2018) shows that an estimation framework for \( f^{\text{opt}}(X) \) in Equation (4) is:

\[ f^{\text{opt}}(X) \in \arg\min_{f} E\left[ \frac{\{RA - f(X)\}^2}{\pi(A, X)} \right]. \quad (5) \]

Considering the class \( \mathcal{F} = \{ f(X) = X^T \beta : \beta \in \mathbb{R}^p \} \) to approximate \( f^{\text{opt}}(X) \), the estimation problem can be solved with ordinary least squares (OLS) with or without regularization.

### 2.1.2 AD-Learning

Qi and Liu (2018) proposed pairwise D-Learning for the case where \( A \in \{1, 2, ..., K\} \). This was improved with AD-Learning (Qi et al., 2020), which uses the angle-based approach of Zhang and Liu (2014) to project treatment \( A \) into \( K \) simplex vertices defined in \( \mathbb{R}^{K-1} \). Let treatment \( A \) be represented by the vector \( u_A \in \mathbb{R}^{K-1} \):

\[ u_A = \begin{cases} \frac{1}{\sqrt{K-1}}1_{K-1}, & A = 1 \\ \sqrt{\frac{K}{K-1}}e_{A-1} - \frac{1 + \sqrt{K}}{\sqrt{(K-1)^2}}1_{K-1}, & 2 \leq A \leq K. \end{cases} \quad (6) \]

Here, \( e_i \) is a \((K-1)\)-dimensional vector of zeroes with 1 in the \( i \)th location. Let the random vector \( U \) be such that \( U \mid (X, A) \overset{d.s.}{=} u_A \). The working model is:

\[ R = \mu(X) + \sum_{k=1}^{K} \delta_k(X)\mathbb{I}(A = k) + \eta, \quad (7) \]

where \( \mu(X) \) is the main effect, \( \delta_k(X) \) is the interaction effect between the \( k \)th treatment and covariates, and \( \eta \) is the mean-zero random error. The contrast \( \delta_k(X) - \delta_j(X) \) can be causally interpreted as the CATE between treatments \( k \) and \( j \). The optimal ITR can then be expressed as:

\[ d^{\text{opt}}(X) = \arg\max_{k \in \{1, ..., K\}} E(R|X = x, A = k) \]

\[ = \arg\max_{k \in \{1, ..., K\}} u_k^T E\left\{ \frac{RU}{\pi(A, X)}|X\right\} \quad (8) \]

\[ = \arg\max_{k \in \{1, ..., K\}} u_k^T f^{\text{opt}}(X), \]

where \( f^{\text{opt}}(X) : \mathbb{R}^{p+1} \to \mathbb{R}^{K-1} \). As shown in Lemma 1 of Qi et al. (2020), for independent responses, this leads to an estimation problem for \( f^{\text{opt}}(X) \) in Equation (8) of the form:

\[ f^{\text{opt}}(X) \in \arg\min_{f \in \mathbb{R}^{K-1}} \frac{\{KRU - f(X)\}^T\{KRU - f(X)\}}{\pi(A, X)}, \quad (9) \]

which, in Lemma 2, is shown to be equivalent to the following estimation framework:

\[ f^{\text{opt}}(X) \in \arg\min_{f \in \mathbb{R}^{K-1}} \frac{1}{\pi(A, X)} \left( E\left\{ \frac{K}{K-1}R - UF(X) \right\} \right)^2. \quad (10) \]

### 2.1.3 RD-Learning

Meng and Qiao (2022) developed RD-Learning, which replaces the outcome \( r_i \) in D-Learning with the residual \( r_i - \hat{m}(x_i) \), where \( \hat{m}(X) \) is an estimator for the main effect, \( m(X) \) (similarly to Zhou et al. (2017)). This reduces the variance and leads to doubly robust estimation of the treatment effect in the sense that consistency is guaranteed if either the main effect model or propensity score model is correctly specified.

### 2.2 SD-Learning

For the binary treatment RCT setting, \( \pi(A, X) \) is known, and assuming Equation (2), the D-Learning estimation problem in Equation (5) induces the following working...
model:
\[ 2RA = f(X) + \epsilon, \]
(11)
showing that the estimation of \( f(X) \) can proceed without needing to model \( m(X) \). Assume that \( E(\epsilon|A,X) = 0 \) and \( \text{var}(\epsilon|A,X) = \sigma^2_0(A,X) \). Note that this error term is very general; it can be an arbitrary function of the treatment and covariates. In this case, the D-Learning estimator of the treatment effect is consistent, but due to the potential heteroscedasticity, it may lack efficiency as it gives each observation equal weight. Considering decision functions in \( F \), we propose a modified D-Learning objective function based on reweighting to gain efficiency:

\[
\hat{\beta}_n^D = \arg\min_{\beta \in \mathbb{R}^p} \frac{1}{n} \sum_{i=1}^{n} \frac{(2RA - X^T \beta)^2}{w(A,X)\pi(A,X)},
\]
(12)
where \( w(A,X) \) is an arbitrary set of weights which need to be specified and/or estimated. The following assumptions establish the basic conditions needed to find optimal weights. For all \( A \in \mathcal{A} \) and \( X \in \mathcal{X} \) almost surely:

**Assumption 1.** \( E(XX^T) \) is full rank and \( E\|X\|^2 < \infty \).

**Assumption 2.** \( 0 < c_1 \leq \sigma^2_0(A,X) \leq c_2 < \infty \) almost surely.

Assumption 1 imposes a finite second moment restriction and assumes nonsingularity of the covariates. Assumption 2 ensures that the true residual variance function is finite and nonzero (bounded above and below).

**Proposition 1.** Under Assumptions 1 and 2, setting
\[ w(A,X) = \frac{\sigma^2_0(A,X)}{\pi(A,X)} \]
minimizes the estimator of the asymptotic variance of Equation (12).

Proposition 1 offers a simple way to perform the reweighting. Let \( \hat{\beta}^D_n \) be a consistent estimate of \( \beta_0 \), which can be obtained by fitting a traditional D-Learning model (Qi and Liu, 2018). Since \( \epsilon = 2AR - X^T \beta \), \( \sigma^2_0(A,X) \) can be estimated by regressing \( (2AR - X^T \hat{\beta}_n^D)^2 \) on \( (A,X) \) through a parametric or nonparametric model. The resulting prediction function can be denoted as \( \hat{\sigma}^2_n(A,X) \). This procedure breaks down into the following implementation steps:

1. Obtain a D-Learning estimator:
\[
\hat{\beta}_n^D = \arg\min_{\beta \in \mathbb{R}^p} \frac{1}{n} \sum_{i=1}^{n} \frac{(2r_i a_i - X_i^T \beta)^2}{\pi(a_i,x_i)}.
\]
(13)
2. Regress the squared residuals from Step 1, \( (2AR - X^T \hat{\beta}_n^D)^2 \), on the treatment and covariates, \( (A,X) \), to obtain prediction function \( \hat{\sigma}^2_n(A,X) \).
3. Find \( \hat{\beta}^S_n \) using:
\[
\hat{\beta}^S_n = \arg\min_{\beta \in \mathbb{R}^p} \frac{1}{n} \sum_{i=1}^{n} \frac{\pi(a_i,x_i)(2r_i a_i - x_i^T \beta)^2}{\hat{\sigma}^2_n(a_i,x_i)}.
\]  
(14)
Thus, the SD-Learning estimator for the binary treatment case is formulated as a least-squares problem, reweighted by the inverse of the estimated residual variance. A procedure for obtaining an improved estimate of the parameters has therefore been provided for binary D-Learning in the case of heteroscedasticity. The same reweighting framework can be used in the case of RD-Learning, where the only differences are that a model for the main effect, \( m(X) \), must be estimated, and the augmented outcome becomes \( R^* = R - m(X) \).

### 2.3 Extension of SD-Learning to multiple treatments

Now, we expand the treatment space to \( K \) treatments, indexed as \( A \in \{1,2,...,K\} \). Let \( u_A \in \mathbb{R}^{K-1} \) be defined as per Equation (6). Assuming Equation (7), the AD-Learning estimation problem in Equation (10) induces the following working model:

\[
\frac{K}{K-1} R = U^T f(X) + \epsilon. \tag{15}
\]

We use this working model under the same scenario as Section 2.2: \( E(\epsilon|A,X) = 0 \) and \( \text{var}(\epsilon|A,X) = \sigma^2_0(A,X) \). The class of linear decision functions is defined as \( F = \{f(X) = B^T X : B \in \mathbb{R}^{p(K-1)}\} \).

Adding an arbitrary weight term, \( w(A,X) \), in the denominator, similarly to the binary case, we propose the SD-Learning objective function as a modified version of AD-Learning:

\[
\hat{\beta}^S_n = \arg\min_{\beta \in \mathbb{R}^{K-1}} \frac{1}{w(A,X)\pi(A,X)} \left( \frac{K}{K-1} R - U^T B^T X \right)^2.
\]
(16)
Again, \( w(A,X) \) must be optimally chosen. We can reframe this objective function so that it is easier to optimize. Using the identity \( \text{Vec}(ABC) = (C^T \otimes A) \text{Vec}(B) \), where \( \otimes \) denotes the Kronecker product, and the fact that \( U^T B^T X \).
is a scalar:
\[
U^\top B^\top X = \text{Vec}(U^\top B^\top X) = (X^\top \otimes U^\top) \text{Vec}(B^\top) = X^\top B_s,
\]
where \(X_s = (X^\top \otimes U^\top)^\top\) and \(B_s = \text{Vec}(B^\top)\). This allows for an equivalent reformulation of the SD-Learning estimation problem:
\[
\tilde{B}_n^S = \arg\min_{B \in \mathbb{R}^{p(K-1)}} P_n \left\{ \frac{1}{\pi(A,X)} \left( \frac{K}{K-1} R - X^\top B_s \right)^2 \right\}.
\]
Note that \(B_s\) and \(X_s\) in Equation (18) are vectors in \(\mathbb{R}^{p(K-1)}\), unlike \(B\) and \(X\) in Equation (16), which are a matrix in \(\mathbb{R}^{p(K-1)}\) and vector in \(\mathbb{R}^p\), respectively. \(w(A,X)\) can now be optimized in a fashion akin to the binary SD-Learning case:

**Proposition 2.** Under Assumption 1 for \(X_s\) instead of \(X\) and Assumption 2, setting \(w(A,X) = \pi(A,X)\) minimizes the estimator of the asymptotic variance of Equation (18).

Note that these are the same weights as found in Proposition 1 for the binary treatment case. Having found the optimal weights, \(w(A,X)\), we switch back to non-vectorized notation (using \(U^\top B^\top X\) instead of the equivalent \(X^\top B_s\)). \(\sigma^2_{\hat{S}}(A,X)\) can be estimated by re-estimating \(\{K/K-1 - u^\top (\tilde{B}_n^AD)^\top X\}^2\) on \((A,X)\) through a parametric or nonparametric model, with the estimate denoted by \(\hat{\sigma}^2_{\hat{S}}(A,X)\). Let \(\tilde{B}_n^AD\) represent a consistent estimate of \(B_0\), obtained via an AD-Learning model (Qi et al. (2020)). The implementation of this procedure is as follows:

1. Obtain an AD-Learning estimator:
\[
\tilde{B}_n^AD = \arg\min_{B \in \mathbb{R}^{p(K-1)}} \frac{1}{n} \sum_{i=1}^{n} \frac{1}{\pi(u_i,x_i)} \left( \frac{K}{K-1} r_i - u_i^\top B_s^\top x_i \right)^2.
\]

2. Regress the squared residuals from Step 1, \(\{K/K-1 - u^\top (\tilde{B}_n^AD)^\top X\}^2\), on \((A,X)\), to obtain prediction function \(\hat{\sigma}^2_{\hat{S}}(A,X)\).

3. Find \(\tilde{B}_n^S\) using:
\[
\tilde{B}_n^S = \arg\min_{B \in \mathbb{R}^{p(K-1)}} \frac{1}{n} \sum_{i=1}^{n} \frac{1}{\hat{\sigma}^2_{\hat{S}}(u_i,x_i)} \left( \frac{K}{K-1} r_i - u_i^\top B_s^\top x_i \right)^2.
\]

Thus, SD-Learning in the multi-arm scenario remains a least-squares problem, and under non-homogeneous error structures, provides an increased-efficiency estimation approach through the angle-based framework (refer to Theoretical Results in Section 3).

As per covariate dimensionality and sparsity assumptions, the OLS steps of the implementation (Steps 1 and 3) in the binary or multi-arm case can be replaced with LASSO, Ridge, Elastic Net, or other regularized least-squares techniques. Detailed proofs of Propositions 1 and 2 can be found in Web Appendix A, and the extension of SD-Learning to observational data can be found in Web Appendix C of the Supporting Information.

We note that Liang and Yu (2020) used the same working model in developing a semiparametric efficiency framework for a large class of estimators \(f(X) = g(\beta^\top X)\), where \(g\) is an arbitrary function, leading to an efficient score function which also includes inverse variance estimates. However, due to the many conditional expectations involved in the score function, the inverse variance terms are difficult to estimate directly. As a result of this, their actual estimation procedure ignores the variance terms and reduces to D-Learning of Tian et al. (2014) if a single-index model is used and \(E(\epsilon | X) = 0\) is assumed. In this case, if a linear decision function is assumed, the leading variance term can be added back and the resulting efficient score function can be solved with the SD-Learning algorithm. Thus, SD-Learning enhances the estimation procedure of Liang and Yu (2020) in the linear decision function case. Additionally, SD-Learning allows the error term to depend on \(A\) as well as \(X\), which is often encountered in practice. Similarly, Mo and Liu (2021) developed efficient Learning (E-Learning), a semiparametric approach incorporating an inverse variance term, but this method necessitates the specification of a main effect model. Moreover, multi-arm treatment estimation in E-Learning no longer maintains the least-squares framework, hence requiring an accelerated proximal gradient method to solve. Compared to both methods, we specialize to linear decision rules, and within this class, achieve (1) the optimal estimator and (2) an estimation procedure which reaches a closed-form solution instead of requiring optimization techniques.

### 2.4 Residual model fitting

The residual modeling step in the implementation of SD-Learning is important, with various parametric and nonparametric options. We propose LASSO, random forest, and/or tree-based XGBoost for residual modeling in order to include a diversity of approaches through parametric assumptions, bagging, and/or boosting, respectively. Additionally, all three methods were chosen for their speed, LASSO and XGBoost for their ability to
TABLE 1  Detailed overview of the SD- and SABD-Learning algorithms

| (1)  | Initial Estimate: Obtain a consistent estimate, $\hat{\beta}^S_0$, of the parameters of the decision function through D-Learning in the binary treatment case or AD-Learning in the multi-arm treatment case (unweighted). |
| (2)  | Hyperparameter Tuning: Obtain $Z = (2 AR - X' \hat{\beta}^S_0)^2$ as the squared residuals from Step (1). Find optimal LASSO (L), random forest (RF), XGBoost (XG), and/or SuperLearner (SL) parameters for predicting $Z$ from treatment and covariates, $(A, X)$, resulting in candidate prediction functions $\hat{\sigma}^2_0(A, X)$, $\hat{\sigma}^2_{\text{RF}}(A, X)$, $\hat{\sigma}^2_{XG}(A, X)$, and $\hat{\sigma}^2_{\text{SL}}(A, X)$. |
| (3)  | Internal $K$-Fold CV: Randomly partition the original sample into $K$ equal-sized (or nearly equal) training folds and let $z_{ij}$, $a_{ij}$, and $r_{ij}$ represent the squared residual, treatment, and outcome, respectively, corresponding to the $i$th observation in the $j$th testing fold (of size $n_j$), where $i \in [1, \ldots, n]$ and $j \in [1, \ldots, K]$. For method $m \in \{L, RF, XG, SL\}$, determine average test set error as $\text{MSE}_m = \frac{1}{K} \sum_{j=1}^{K} \left[ \frac{1}{n_j} \sum_{i=1}^{n_j} \left( z_{ij} - \hat{\sigma}^2_{m_j}(a_{ij}, x_{ij}) \right)^2 \right]$, where $\hat{\sigma}^2_{m_j}(A, X)$ represents the prediction function resulting from method $m$ when trained on training data from fold $j$. Pick the method with the lowest average error, $m_{\text{opt}} = \arg\min_m \{ \text{MSE}_m \}$. |
| (4)  | Obtaining Weights: Using $m_{\text{opt}}$ from Step (3), obtain the predicted squared residual for each of $n$ observations, $\hat{\sigma}^2_{m_{\text{opt}}}(A, X)$. |
| (5)  | Reweighted Estimate: Using $\hat{\sigma}^2_{m_{\text{opt}}}(A, X)$ from Step (4) as weights, obtain stabilized parameter estimates through SD-Learning as in Equation (14) or SABD-Learning as in Equation (20). |

handle sparsity (Zhang & Huang, 2008; Fauzan & Murfi, 2018; Chen & Guestrin, 2016), and in the case of random forests and XGBoost, flexibility. They also have a relatively low number of hyperparameters to tune, making their implementation easier. A SuperLearner algorithm can also be used, which combines candidate parametric and nonparametric methods to find an optimal combination which minimizes cross-validated risk. The SuperLearner has been shown to perform asymptotically as well as or better than any of the constituent candidate learners (van der Laan et al., 2007).

Instead of picking one of the above residual modeling algorithms directly, we propose to first compare them using an internal cross-validation step. After squared residuals from the initial D-Learning fit, $Z = (2 AR - X' \hat{\beta}^S_0)^2$, are obtained, each algorithm is fit by regressing $Z$ against treatment and covariates, $(A, X)$. After the hyperparameters for each algorithm are tuned, they can be compared using $K$-fold cross-validation with mean squared error (MSE) as the evaluation metric. This results in finding the best residual modeling method. We describe the algorithm in detail in Table 1.

3 | THEORETICAL RESULTS

In this section, we establish the theoretical properties of SD-Learning for settings with fixed dimension, $p$. We establish consistency and asymptotic normality of the SD-Learning estimator, along with bounds for the empirical value function. Detailed proofs for all theorems and remarks can be found in Web Appendix B of the Supporting information. We state two additional assumptions below and will delineate which assumptions are needed for each theorem.

**Assumption 3.** $\hat{\sigma}^2_0(A, X)$ is uniformly consistent for $\sigma^2_0(A, X)$. That is, $\left| \left| \hat{\sigma}^2_0(A, X) - \sigma^2_0(A, X) \right| \right|_{\infty, (A, X)}^p \to 0$, where $\left| \left| \cdot \right| \right|_{\infty, (A, X)}$ represents the uniform norm over $(A, X)$, and $(A, X)$ is in a bounded set. In the case of observational data, we also require $\left| \left| \hat{\pi}_n(A, X) - \pi_0(A, X) \right| \right|_{\infty, (A, X)}^p \to 0$.

**Assumption 4.** $\forall \gamma > 0, \exists \zeta$ which is P-Donsker such that $\Pr\left\{ \frac{X \zeta}{\hat{\sigma}^S_0(A, X)} \in \mathbb{G} \right\} > 1 - \gamma, \forall n$ large enough.

In Remark 1, we propose two estimation methods for $\hat{\sigma}^2_0(A, X)$ and provide justification that they satisfy Assumptions 3 and 4. Then, in Theorem 1, we establish consistency of the SD-Learning estimator in the binary treatment setting, and with consistency established, asymptotic normality of the estimator is shown in Theorem 2.

**Remark 1.** Estimating $\sigma^2_0(A, X)$ by regressing squared residuals, $Z$, against treatment and covariates, $(A, X)$, with (1) linear regression with arbitrary features and (2) random forests satisfies Assumptions 3 and 4.

**Theorem 1.** If Assumptions 1–3 are met, $\hat{\beta}^S_n \overset{P}{\to} \beta_0$.

**Theorem 2.** Let $Y^* = 2 A Y$ and $E(Y^*|X) = X \beta_0$. Denote $U_0 = E \left\{ \frac{XX^\top}{\sigma^2_0(A, X)} \right\}$. If Assumption 4 is additionally met, $\sqrt{n}(\hat{\beta}^S_n - \beta_0)$ is asymptotically normal with variance $U_0^{-1}$.

$\hat{\beta}^S_n$ achieves the lower bound of the asymptotic variance shown in Proposition 1, and is thus the optimal estimator for $\beta_0$ among the weighted choices of Equation (12).
Thus, we have achieved consistency, convergence, and asymptotic efficiency.

Since Section 2.3 unifies the framework between SD-Learning in binary versus multi-arm treatment scenarios, the extension of Theorems 1 and 2 to multi-arm treatment is natural. This development is outlined in Theorem 3:

**Theorem 3.** Let \( X = (X^T \otimes U^T)^T, \quad B_s = \text{vec}(B^T), \) \( Y^* = \frac{K}{K-1} R \) where \( A \in \{1, 2, \ldots, K\}, \) and \( E(Y^*|X_s) = X_s^T B_s. \)

Denote \( U_0 = E\left\{ \frac{X_i X_i^T}{\sigma_0(A, X)} \right\}. \) Under Assumption 1 for \( X, \)

instead of \( X \) and Assumptions 2 and 3, \( \hat{B}_s \xrightarrow{p} B_s. \) Moreover, if Assumption 4 is additionally met, \( \sqrt{n}(\hat{B}_s - B_s) \) is asymptotically normally distributed with variance \( U_0^{-1}. \)

Thus, all results established for the binary treatment case extend to the multi-arm setting. Combining the asymptotic normality result of Theorem 3 with Theorem 1 of Qi et al. (2020) which shows that:

\[
V(d^\text{opt}) - V(\hat{d}_n) \leq \frac{2K(K-1)}{1-C(K)} \left( E\left\| f^\text{opt} - \hat{f}_n \right\|_2^2 \right)^{1/2},
\]

where constant \( C(K) \) only depends on \( K, \sqrt{n}\)-convergence of \( V(\hat{d}_n) \) to \( V(d^\text{opt}) \) is established.

### 4 | NUMERICAL RESULTS: SIMULATION STUDIES

We perform head-to-head comparisons between SD-Learning and the D-Learning family of methods (D-Learning, AD-Learning, and RD-Learning), while noting that D-Learning has been extensively compared to other precision medicine methods in existing literature (Wang et al., 2020; Qi et al., 2020; Mo & Liu, 2021). In all simulations, LASSO is used to obtain estimates of the decision function parameters (Steps 1 and 5 of Table 1), and we use internal cross-validation to pick between LASSO, random forest, and XGBoost at the intermediate residual modeling steps (Steps 2 and 3 of Table 1). Methods are evaluated based on three criteria:

1. **Average prediction error:** Coefficient accuracy based on MSE of true versus predicted decision functions. For binary simulation settings, \( \text{APE} = n^{-1} \sum_{i=1}^{n} (\beta_0^i - \hat{\beta}_0^i)^2; \) for multi-arm settings, \( \text{APE} = n^{-1} \sum_{i=1}^{n} \sum_{k=1}^{K} \delta_k(x_i) u_k - \sum_{k=1}^{K} \delta_k(x_i) \hat{u}_k \).

2. **Misclassification rate:** percent incorrect treatment assignment.

3. **Empirical value:** \( \hat{\nu}(d) = \frac{P_n[R \cdot \mathbb{1}(A = d(X))/\pi(A, X)]}{P_n[\mathbb{1}(A = d(X))/\pi(A, X)]}. \)

Better performance corresponds to lower APE, lower misclassification rate, and higher empirical value. In all simulations, performance based on these criteria is determined on a test dataset with 10,000 observations. 100 replications of each simulation setting are performed.

We compare all binary methods with \( p = \{30, 60, 120\} \) and multi-arm methods with \( p = \{20, 40, 60\}. \) Simulated observations are independent with continuous covariates generated according to a \( U[-1,1] \) distribution. To allow for heteroscedasticity, the outcome is generated according to the working model in Equation (2) for binary treatments and Equation (7) for multi-arm treatments, but with \( \eta \) generated according to \( \sigma_0(X) \cdot Z, \) where \( Z \sim N(0, 1) \) and \( \sigma_0(X) > 0. \) Here, non-constant \( \sigma_0(X) \) introduces heteroscedasticity.

#### 4.1 Binary treatment simulations

We compare SD-Learning and D-Learning with four simulation settings, where \( n = 200: \)

1. \( m(X) = 1 + 2X_1 + X_2 + 0.5X_1; \delta(X) = 0.5(0.9 - X_1); \sigma_0^2(X) = 1; \pi(A = 1, X) = 0.5 \) (RCT).

2. \( m(X) = 1 + 12X_1 + 6X_2 + 3X_3; \delta(X) = 4X_1; \sigma_0^2(X) = 0.25 + (X_2 + 1)^2; \pi(A = 1, X) = 0.25 + 0.5 \cdot 1(X_2 > 0). \)

3. \( m(X) = 1 + 10X_1 + 10X_2 + 20X_3 + 5X_4; \delta(X) = 4(0.3 - X_1 - X_2); \sigma_0^2(X) = 1 + 4X_3^2; \pi(A = 1, X) = 0.5 \) (RCT).

4. \( m(X) = 1 + 10X_1 + 6X_1^2 - 6X_2^2 + 10X_3; \delta(X) = 2X_2^2 + 1.5X_1 + 3X_4; \sigma_0^2(X) = 0.25 + (0.3 - X_1 - X_2)^2; \pi(A = 1, X) = 0.5 \) (RCT).

Scenario (1) is very similar to the third linear decision boundary scenario in Qi and Liu (2018), and has homoscedastic error. In this case, an intermediate residual reweighting step is not necessary since the optimal weights are \( \omega(X) = 1 \) by design, which is the default for D-Learning. Scenarios (2)-(4) introduce heteroscedasticity through the error term. Scenarios (2) and (3) meet linear decision function assumptions, and in both, the interaction effect has variables in common with the main effect but not with the error function. Specifically, Scenario (2) is
an observational data setting, where we model the propensity score through random forest for binary classification. Finally, Scenario (4) is a nonlinear decision boundary scenario, where SD-Learning is misspecified. This scenario will help test the robustness of SD-Learning in situations, where the true parameters of the decision function cannot be consistently estimated.

Figure 1 shows the APE for all four scenarios, and Table 2 contains misclassification rates and the estimated empirical value function on the test dataset. Since Scenario (1) has homoscedastic error, D-Learning performs “optimally” in the sense of correctly specified weights; hence, stabilizing the D-Learning estimates is technically not needed. However, SD-Learning still performs similarly to D-Learning, with comparable APE and only slightly lower misclassification rate and empirical value. For Scenarios (2) and (3), SD-Learning outperforms D-Learning because of the heterogeneous error structure. SD-Learning prioritizes observations with smaller expected outcome variance, and therefore estimates parameters more efficiently, as shown by the lower APEs in Figure 1. This results in better classification and therefore, higher empirical value. In Scenario (4), although the true decision boundary is nonlinear, SD-Learning’s flexible modeling of heteroscedasticity gives it an advantage over D-Learning.

We also perform simulations to show the advantage of stabilizing the estimates of RD-Learning. For these simulations, n = 100. For the main effect modeling step, LASSO was used, but nonparametric methods may also be used as per Meng and Qiao (2022). The simulation settings for SRD-Learning versus RD-Learning are as follows:

\[
m(X) = 1 + 10X_1 + 10X_2 + 20X_3 + 20X_5 + 10X_1X_2; \\
\delta(X) = 1.25X_3 + 2.5X_4; \\
\sigma_0^2(X) = 1 + 0.1(1 + X_1^2); \\
\pi(A = 1, X) = 0.5 \text{(RCT)}.\]
Both scenarios have heterogeneous error. Scenario (5) has a true linear decision function, whereas Scenario (6) has a nonlinear decision function along with a nonlinear cosine term in the main effect. Figure 1 shows the APE for both scenarios, and Table 2 contains misclassification rates and the estimated empirical value function on the test dataset. Although RD-Learning is already robust in the sense that the main effect is removed before model fitting, the reweighing step of SRD-Learning adds efficiency in situations with heteroscedasticity—even in the presence of a misspecified decision function and nonlinearity in the main effect.

## 4.2 Multi-arm treatment simulations

We compare SABD-Learning and AD-Learning with two multi-arm treatment scenarios under heteroscedasticity, setting $K = 4$ and $n = 200$ in both:

**Scenario (7)**

\[
m(X) = 1 + 2X_1 + 2X_2; \\
\sigma^2_0(X) = .25 + 0.2(1.5 - X_2)^2; \\
\delta(X) \begin{cases} 
.75 + 1.5X_1 + 1.5X_2 + 1.5X_3 + 1.5X_4; \\
.75 + 1.5X_1 - 1.5X_2 - 1.5X_3 + 1.5X_4; \\
.75 + 1.5X_1 - 1.5X_2 + 1.5X_3 - 1.5X_4; \\
.75 - 1.5X_1 + 1.5X_2 - 1.5X_3 + 1.5X_4; \\
.25; \\
0.25 \\
0.25 \\
0.25 \\
0.25 
\end{cases} \quad A \begin{cases} 
1 \\
2 \\
3 \\
4 
\end{cases} \\
\pi(A = 1, X) = 0.5 \quad (RCT).
\]

**Scenario (8)**

\[
m(X) = 1 + X_2 + 3X_3 + 2X_4; \\
\sigma^2_0(X) = .25 + 2X_2 \times 1(X_2 > 0) + X_3 \times 1(X_3 > 0, A = 2); \\
\delta(X) \begin{cases} 
0.5 + 2X_1 + X_2 + X_3; \\
0.25 + 1(X_1 < 0) + 0.4 \times 1(X_1 > 0) \\
1 + X_1 - X_2 - X_3; \\
0.25 + 1(X_1 < 0) + 0.2 \times 1(X_1 > 0) \\
1.5 + 3X_1 - X_2 + X_3; \\
0.25 + 1(X_1 < 0) + 0.2 \times 1(X_1 > 0) \\
1 - X_1 - X_2 + X_3; \\
0.25 + 1(X_1 < 0) + 0.2 \times 1(X_1 > 0) 
\end{cases} \quad A \begin{cases} 
1 \\
2 \\
3 \\
4 
\end{cases} \\
\pi(A = 1, X) = 0.25 \quad (RCT).
\]

Both scenarios meet linear assumptions for the treatment–covariate interaction effects. In Scenario (7), heterogeneous error is a quadratic function of the covariate $X_2$, but in Scenario (8), it is a spline function of $X_2$, $X_3$, and $X_4$ dependent on treatments $A = 1$ and $A = 2$. Scenario (8) is additionally an observational data setting, with the propensity score modeled through random forest for multiclass classification. Figure 1 reports APE for both scenarios, and Table 3 displays misclassification and value results. Reweighting in the multi-arm scenario under heteroscedasticity also improves efficiency, resulting in lower APEs and misclassification rates, and higher empirical values. SABD-Learning improves the performance of AD-Learning when an error structure can be learned and utilized to obtain decision rules built by favoring observations which are more likely to represent signal than noise.

## 5 DATA ANALYSIS: AIDS CLINICAL TRIAL

We observe that in the ACTG175 data of 2139 patients, heteroscedasticity is present through covariates and treatments; this analysis is detailed in Web Appendix C of the Supporting information. Hence, we apply SD-Learning to the data, which may benefit from a reweighing approach. This double-blinded study evaluated monotherapy versus combination approaches to increasing CD4 cell counts in HIV-1-infected patients with initial cell counts of 200–500 cells/mm$^3$ (Hammer et al., 1996). AIDS-defining illnesses have been shown to decrease as CD4 cell count increases (Mocroft et al., 2013), so larger increases in CD4 cell count are preferable.

Patients were randomly assigned to one of four daily regimens with equal probability:

1. 600 mg zidovudine (Z);
2. 600 mg zidovudine + 400 mg didanosine (ZD);
3. 600 mg zidovudine + 2.25 mg zalcitabine (ZZ);
4. 400 mg didanosine (D).

The outcome we use for this analysis is the change in CD4 cell count from baseline to 20 weeks, as done in Qi and Liu (2018) and Qi et al. (2020). Twelve covariates are selected as per Fan et al. (2017); five continuous: weight (kg), age (years), Karnofsky score (0–100), baseline CD4 count (cells/mm$^3$), baseline CD8 count (cells/mm$^3$); and seven binary: hemophilia (1=yes, 2=no), homosexual activity (1=yes, 0=no), history of intravenous drug use (1=yes, 0=no), race (1=non-white, 0=white), gender (1=male, 0=female), antiretroviral
TABLE 2  Mean empirical value and misclassification rate, along with standard error of the mean (SEM), for four binary D- vs. SD-Learning simulations and two binary RD- vs. SRD-Learning simulations for 30, 60, and 120 covariates

| Scenario 1 | p = 30 | Value | Misclass. | Value | Misclass. | Value | Misclass. |
|------------|--------|-------|-----------|-------|-----------|-------|-----------|
| D-Learning | 1.43 (0.01) | 0.07 (0.01) | 1.43 (0.01) | 0.07 (0.01) | 1.44 (0.01) | 0.08 (0.01) |
| SD-Learning | 1.42 (0.01) | 0.08 (0.01) | 1.42 (0.01) | 0.09 (0.01) | 1.43 (0.01) | 0.09 (0.01) |

| Scenario 2 | p = 60 | Value | Misclass. | Value | Misclass. | Value | Misclass. |
|------------|--------|-------|-----------|-------|-----------|-------|-----------|
| D-Learning | 2.46 (0.01) | 0.14 (0.01) | 2.27 (0.01) | 0.14 (0.01) | 2.17 (0.01) | 0.15 (0.01) |
| SD-Learning | 2.48 (0.01) | 0.13 (0.01) | 2.30 (0.01) | 0.13 (0.01) | 2.21 (0.01) | 0.14 (0.01) |

| Scenario 3 | p = 120 | Value | Misclass. | Value | Misclass. | Value | Misclass. |
|------------|---------|-------|-----------|-------|-----------|-------|-----------|
| D-Learning | 2.64 (0.09) | 0.29 (0.01) | 2.40 (0.09) | 0.32 (0.01) | 2.16 (0.08) | 0.36 (0.01) |
| SD-Learning | 2.82 (0.08) | 0.26 (0.01) | 2.56 (0.09) | 0.30 (0.01) | 2.21 (0.08) | 0.35 (0.01) |

| Scenario 4 | p = 20 | Value | Misclass. | Value | Misclass. | Value | Misclass. |
|------------|--------|-------|-----------|-------|-----------|-------|-----------|
| D-Learning | 1.81 (0.05) | 0.31 (0.01) | 1.91 (0.05) | 0.32 (0.01) | 1.90 (0.04) | 0.33 (0.01) |
| SD-Learning | 2.08 (0.03) | 0.24 (0.01) | 2.02 (0.05) | 0.29 (0.01) | 1.99 (0.04) | 0.31 (0.01) |

| Scenario 5 | p = 40 | Value | Misclass. | Value | Misclass. | Value | Misclass. |
|------------|--------|-------|-----------|-------|-----------|-------|-----------|
| RD-Learning | 1.94 (0.03) | 0.20 (0.01) | 1.91 (0.05) | 0.27 (0.01) | 1.43 (0.05) | 0.30 (0.01) |
| SRD-Learning | 1.98 (0.03) | 0.18 (0.01) | 2.02 (0.04) | 0.24 (0.01) | 1.55 (0.05) | 0.27 (0.01) |

| Scenario 6 | p = 60 | Value | Misclass. | Value | Misclass. | Value | Misclass. |
|------------|--------|-------|-----------|-------|-----------|-------|-----------|
| RD-Learning | 6.16 (0.04) | 0.21 (0.01) | 5.94 (0.04) | 0.23 (0.01) | 5.65 (0.05) | 0.25 (0.02) |
| SRD-Learning | 6.24 (0.04) | 0.19 (0.01) | 5.97 (0.04) | 0.23 (0.01) | 5.70 (0.05) | 0.24 (0.01) |

Note: The best-performing method for each category is bolded.

TABLE 3  Mean empirical value and misclassification rate, along with standard error of the mean (SEM), for two multi-arm scenarios comparing AD-Learning to SABD-Learning. All simulations are repeated with 20, 40, and 60 covariates

| Scenario 7 | p = 20 | Value | Misclass. | Value | Misclass. | Value | Misclass. |
|------------|--------|-------|-----------|-------|-----------|-------|-----------|
| AD-Learning | 3.39 (0.02) | 0.34 (0.01) | 3.23 (0.03) | 0.39 (0.01) | 3.19 (0.03) | 0.42 (0.01) |
| SABD-Learning | 3.49 (0.01) | 0.30 (0.01) | 3.39 (0.02) | 0.33 (0.01) | 3.39 (0.02) | 0.35 (0.01) |

| Scenario 8 | p = 40 | Value | Misclass. | Value | Misclass. | Value | Misclass. |
|------------|--------|-------|-----------|-------|-----------|-------|-----------|
| AD-Learning | 2.71 (0.03) | 0.52 (0.01) | 2.68 (0.03) | 0.56 (0.01) | 2.67 (0.03) | 0.58 (0.01) |
| SABD-Learning | 2.86 (0.02) | 0.47 (0.01) | 2.86 (0.02) | 0.50 (0.01) | 2.88 (0.02) | 0.51 (0.01) |

Note: The best-performing method for each category is bolded.

history (1=experienced, 0=naive), and symptomatic status (1=symptomatic, 0=asymptomatic). For all comparisons, LASSO is used to obtain estimates of the decision function parameters. LASSO, random forest, and XGBoost are tuned and used for the residual modeling step of SD- and SABD-Learning, with the optimal method picked by internal cross-validation.

5.1 Binary scenario

We compare the performance of D-Learning and its corresponding stabilized version, SD-Learning, for each pairwise set of treatments from the four choices. We randomly split the data into a training set of \( n \) observations, using the rest of the observations as testing data. \( n \) is selected to be 100, 200, 400, and 800. For generating empirical value estimates, \( \hat{V}(d) \), we perform Monte Carlo cross-validation (repeated random subsampling) with 1,000 iterations at each \( n \). The corresponding binary treatment results are shown in Table 4.

In terms of empirical value, SD-Learning improves upon the performance of D-Learning in most scenarios, especially for comparisons involving treatment ZD. SD- and D-Learning perform approximately equally well for ZZ versus D, with empirical values differing by less than 0.10 in all cases. For pairwise comparisons of Z versus ZZ and Z versus D, both methods eventually converge upon
TABLE 4  Empirical value estimates for binary AIDS data scenarios comparing performance of D- and SD-Learning on each pairwise set of treatments (Z, ZD, ZZ, D) at varying sample sizes of training data (n = 100, 200, 400, 800)

|       | D-Learning | SD-Learning | Z vs. ZD | n = 100 | D-Learning | SD-Learning |
|-------|------------|-------------|----------|---------|------------|-------------|
| ZD vs. ZZ | n = 100 | 49.12 (0.31) | 49.24 (0.30) | n = 100 | 51.75 (0.21) | 52.18 (0.20) |
|       | n = 200 | 51.76 (0.21) | 52.12 (0.20) | n = 200 | 52.92 (0.15) | 53.43 (0.14) |
|       | n = 400 | 51.76 (0.21) | 52.12 (0.20) | n = 400 | 53.48 (0.16) | 53.90 (0.16) |
|       | n = 800 | 52.92 (0.15) | 53.43 (0.14) | n = 800 | 53.84 (0.35) | 54.30 (0.35) |
| ZD vs. D | n = 100 | 48.67 (0.29) | 48.92 (0.28) | n = 100 | 15.62 (0.27) | 15.62 (0.27) |
|       | n = 200 | 50.59 (0.23) | 51.02 (0.23) | n = 200 | 17.96 (0.14) | 18.10 (0.15) |
|       | n = 400 | 52.88 (0.19) | 53.28 (0.19) | n = 400 | ZZ for over 99% of patients. |
|       | n = 800 | 55.99 (0.33) | 56.42 (0.33) | n = 800 | ZZ for over 99% of patients. |
| ZZ vs. D | n = 100 | 23.57 (0.13) | 23.64 (0.13) | n = 100 | 24.57 (0.19) | 24.55 (0.19) |
|       | n = 200 | 23.89 (0.13) | 23.88 (0.14) | n = 200 | D for over 99% of patients. |
|       | n = 400 | 24.52 (0.15) | 24.57 (0.15) | n = 400 | D for over 99% of patients. |
|       | n = 800 | 25.55 (0.25) | 25.48 (0.25) | n = 800 | D for over 99% of patients. |

Notes: At each sample size, results are averaged from 1000 replications, and corresponding standard error of the mean (SEM) is shown. The best-performing method at each level of sample size is bolded. When both methods converge upon recommending a single treatment (in over 99% of patients across all replications), the treatment is specified instead of the (nearly identical) value estimates.

TABLE 5  Empirical value estimates for multi-arm AIDS data scenarios comparing the performance of AD- and SABD-Learning in selecting among four treatments simultaneously

|       | AD-Learning | SABD-Learning |
|-------|-------------|---------------|
| n = 100 | 43.27 (0.43) | 43.98 (0.43) |
| n = 200 | 47.19 (0.37) | 48.29 (0.35) |
| n = 400 | 50.57 (0.25) | 51.74 (0.22) |
| n = 800 | 53.35 (0.18) | 54.25 (0.17) |
| n = 1200| 54.27 (0.23) | 55.24 (0.23) |

Notes: Varying sample sizes of the training data were chosen to be n = 100, 200, 400, 800, and 1200. At each sample size, results are averaged from 1000 replications, and the corresponding standard error of the mean (SEM) is shown. The best-performing method at each level of sample size is bolded.

5.2 Multi-arm treatment scenario

We now compare the performance of AD- and SABD-Learning at varying sample sizes while considering all four treatments simultaneously. We randomly split the data into training sets of n = 100, 200, 400, 800, and 1200, using the rest of the observations as testing data. The procedure is repeated 1000 times for each value of n. All results are shown in Table 5.

SABD-Learning has a distinct advantage over AD-Learning at all observed training data sample sizes. The stabilization step weights patients differentially based on predicted squared residual from the initial AD-Learning step. This results in more efficiently estimated treatment rules and therefore higher values, even in scenarios with very low sample sizes.

6 DISCUSSION

In this paper, we propose SD-Learning, which boosts the efficiency of D-Learning in a wide range of scenarios with more general error functions, thus enhancing the utility of D-Learning on datasets which may be encountered in practice. The performance of SD-Learning relies on sufficiently modeling residuals from an initial D-Learning fit, which can be achieved through a variety of parametric or nonparametric methods. When the true residual variance is homogeneous, SD-Learning reduces to D-Learning. Our results suggest that under this homogeneous error, SD-Learning pays a minor efficiency price, but under heterogeneous error, it can offer substantial efficiency gains. Additionally, SD-Learning parameter estimates are asymptotically normal when OLS is used for the estimation steps, allowing post-modeling inference even in the multi-arm treatment scenario.

The implementation benefits of SD-, SRD-, and SABD-Learning lie in the fact that they are straightforward to use and can simply be stacked on top of D-, RD-, and AD-Learning which have been shown to perform well in a multitude of settings. Additionally, our methodology even in the multi-arm treatment setting (SABD-Learning) remains estimable with a least-squares framework and
closed-form solution, not requiring the use of optimization algorithms.

For practical use, we recommend SRD-Learning, since Meng and Qiao (2022) showed that incorporation of the mean outcome model results in protection against incorrect specification of the propensity score model. This would be especially helpful in the case of observational data. In general, unless there is prior evidence of homoscedasticity, we recommend SD-Learning methods because they involve reweighting, which is a simple add-on that increases the efficiency of already-proven methods of Qi and Liu (2018) and Tian et al. (2014). We also suggest considering a variety of nonparametric prediction algorithms for the residual modeling step in order to gain an understanding of the heteroscedasticity structure of the dataset at hand. Random forests appeared to work well under a wide variety of scenarios, with the number of trees as the most important tuning parameter.

Several future extensions of this work are possible. Theoretical results in Section 3, including Remark 1 for random forests, can be extended to the case where covariate dimension, $p$, increases with $n$. A natural methodological extension would be to allow for SD-Learning to estimate nonlinear decision rules using Reproducing Kernel Hilbert Space (RKHS) techniques. Additionally, SD-Learning may be broadened to work for binary and survival outcomes under heteroscedasticity. As proposed in this paper, SD-Learning assumes independence (but not identically distributed errors) between observations. It would also be valuable to use SD-Learning in scenarios with correlation between observations. This would be akin to feasible generalized least squares (FGLS; Olive, 2017) for ITR estimation.

ACKNOWLEDGMENTS

We thank the reviewers for their valuable comments, which led to an improved paper. We also thank Marissa Ashner and Kyle Grosser for providing very helpful feedback.

DATA AVAILABILITY STATEMENT

The ACTG175 dataset used in this paper is publicly available via the R package speff2trial (Version 1.0.5), which can be accessed through CRAN at https://cran.r-project.org/web/packages/speff2trial/index.html.

ORCID

Kushal S. Shah https://orcid.org/0000-0002-0982-4386
Haoda Fu https://orcid.org/0000-0002-0538-4651
Michael R. Kosorok https://orcid.org/0000-0002-6070-9738

REFERENCES

Breiman, L. (2001) Random forests. Machine Learning, 45(1), 5–32.
Chen, T. & Guestrin, C. (2016) XGBoost: a scalable tree boosting system. Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, New York, NY, USA. Association for Computing Machinery.
Cui, Y., Zhu, R. & Kosorok, M. (2017) Tree based weighted learning for estimating individualized treatment rules with censored data. Electronic Journal of Statistics, 11(2), 3927–3953.
Fan, C., Lu, W., Song, R. & Zhou, Y. (2017) Concordance-assisted learning for estimating optimal individualized treatment regimes. Journal of the Royal Statistical Society, Series B, 79(5), 1565–1582.
Fauzan, M.A. & Murfi, H. (2018) The accuracy of XGBoost for insurance claim prediction. International Journal of Advances in Soft Computing and its Applications, 10(2), 13.
Hammer, S.M., Katzenstein, D.A., Hughesm, M.D., Gundacker, H., Schooley, R.T., Haubrich, R.H. et al. (1996) A trial comparing nucleoside monotherapy with combination therapy in HIV-infected adults with CD4 cell counts from 200 to 500 per cubic millimeter. New England Journal of Medicine, 335(15), 1081–1090.
Hernán, M.A. & Robins, J.M. (2019) Causal inference: what if. CRC Press.
Jacob, D. (2021) CATE meets ML: the conditional average treatment effect and machine learning. arXiv:2104.09935.
Kallus, N. (2018) Balanced policy evaluation and learning. Advances in neural information processing systems, Vol. 31, pp. 8909–8920.
Kosorok, M.R. & Laber, E.B. (2019) Precision medicine. Annual Review of Statistics and its Application, 6, 263–286.
Kosorok, M.R. & Moodie, E.E.M. (2016) Adaptive treatment strategies in practice: planning trials and analyzing data for personalized medicine. ASA–SIAM statistics and applied probability. Philadelphia, PA: Society for Industrial and Applied Mathematics.
Liang, M. & Yu, M. (2020) A semiparametric approach to model effect modification. Journal of the American Statistical Association, 1–13.
Meng, H. & Qiao, X. (2022) Augmented direct learning for conditional average treatment effect estimation with double robustness. Electronic Journal of Statistics, 16(1), 3523–3560.
Mo, W. & Liu, Y. (2021) Efficient learning of optimal individualized treatment rules for heteroscedastic or misspecified treatment-free effect models. Journal of the Royal Statistical Society, Series B, 84(2), 440–472.
Mocroft, A., Furrer, H.J., Miro, J.M., Reissm, P., Mussini, C., Kirk, O. et al. (2013) The incidence of AIDS-defining illnesses at a current CD4 count greater than 200 cells/microliter in the post-combination antiretroviral therapy era. Clinical Infectious Diseases, 57(7), 1038–1047.
Murphy, S.A. (2003) Optimal dynamic treatment regimes. Journal of the Royal Statistical Society, Series B, 65(2), 331–355.
Olive, D.J. (2017) WLS and generalized least squares. Linear Rr (pp. 163–173). Cham: Springer International Publishing.
Qi, Z., Liu, D., Fu, H. & Liu, Y. (2020) Multi-armed angle-based direct learning for estimating optimal individualized treatment rules with various outcomes. Journal of the American Statistical Association, 115(530), 678–691.
Qi, Z. & Liu, Y. (2018) D-learning to estimate optimal individual treatment rules. Electronic Journal of Statistics, 12(2), 3601–3638.
Qian, M. & Murphy, S.A. (2011) Performance guarantees for individualized treatment rules. Annals of Statistics, 39(2), 1180–1210.
Rubin, D.B. (1974) Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of Educational Psychology*, 66(5), 688–701.

Schulte, P.J., Tsiatis, A.A., Laber, E.B. & Davidian, M. (2014) Q- and A-learning methods for estimating optimal dynamic treatment regimes. *Statistical Science*, 29(4), 640–661.

Tian, L., Alizadeh, A.A., Gentles, A.J. & Tibshirani, R. (2014) A simple method for estimating interactions between a treatment and a large number of covariates. *Journal of the American Statistical Association*, 109(508), 1517–1532.

van der Laan, M.J., Polley, E.C. & Hubbard, A.E. (2007) Super learner. *Statistical Applications in Genetics and Molecular Biology*, 6(1), 1–23.

Wang, D., Fu, H. & Loh, P.-L. (2020) Boosting algorithms for estimating optimal individualized treatment rules. Technical Report. arXiv:2002.00079.

Xiao, W., Zhang, H.H. & Lu, W. (2019) Robust regression for optimal individualized treatment rules. *Statistics in Medicine*, 38(11), 2059–2073.

Zhang, B. & Zhang, M. (2018) C-learning: a new classification framework to estimate optimal dynamic treatment regimes. *Biometrics*, 74(3), 891–899.

Zhang, C., Chen, J., Fu, H., He, X., Zhao, Y.-Q. & Liu, Y. (2020) Multicategory outcome weighted margin-based learning for estimating individualized treatment rules. *Statistica Sinica*, 30, 1857–1879.

Zhang, C. & Liu, Y. (2014) Multicategory angle-based large-margin classification. *Biometrika*, 101(3), 625–640.

Zhang, C.-H. & Huang, J. (2008) The sparsity and bias of the Lasso selection in high-dimensional linear regression. *The Annals of Statistics*, 36(4), 1567–1594.

Zhang, J., Troxel, A.B. & Petkova, E. (2021) Robust index of confidence weighted learning for optimal individualized treatment rule estimation. *Stat*, 10(1), e374.

Zhao, Y., Laber, E.B., Ning, Y., Saha, S. & Sands, B.E. (2019) Efficient augmentation and relaxation learning for individualized treatment rules using observational data. *Journal of Machine Learning Research*, 20, 48.

Zhou, X., Mayer-Hamblett, N., Khan, U. & Kosorok, M.R. (2017) Residual weighted learning for estimating individualized treatment rules. *Journal of the American Statistical Association*, 112(517), 169–187.

**SUPPORTING INFORMATION**

Web Appendices referenced in Sections 2, 3, and 5, and R code to implement the proposed methods, are available with this paper at the Biometrics website on Wiley Online Library.

---

**How to cite this article:** Shah, K.S., Fu, H., & Kosorok, M.R. (2023) Stabilized direct learning for efficient estimation of individualized treatment rules. *Biometrics*, 79, 2843–2856. [https://doi.org/10.1111/biom.13818](https://doi.org/10.1111/biom.13818)