Hybrid Meta-learners for Estimating Heterogeneous Treatment Effects

Zhongyuan Liang
UC Berkeley, UCSF

Lars van der Laan
University of Washington

Ahmed Alaa
UC Berkeley, UCSF

Abstract

Estimating conditional average treatment effects (CATE) from observational data involves modeling decisions that differ from supervised learning, particularly concerning how to regularize model complexity. Previous approaches can be grouped into two primary "meta-learner" paradigms that impose distinct inductive biases. *Indirect* meta-learners first fit and regularize separate potential outcome (PO) models and then estimate CATE by taking their difference, whereas *direct* meta-learners construct and directly regularize estimators for the CATE function itself. Neither approach consistently outperforms the other across all scenarios: indirect learners perform well when the PO functions are simple, while direct learners outperform when the CATE is simpler than individual PO functions. In this paper, we introduce the *Hybrid Learner* (H-learner), a novel regularization strategy that interpolates between the direct and indirect regularizations depending on the dataset at hand. The H-learner achieves this by learning intermediate functions whose difference closely approximates the CATE without necessarily requiring accurate individual approximations of the POs themselves. We demonstrate that intentionally allowing suboptimal fits to the POs improves the bias-variance tradeoff in estimating CATE. Experiments conducted on semi-synthetic and real-world benchmark datasets illustrate that the H-learner consistently operates at the Pareto frontier, effectively combining the strengths of both direct and indirect meta-learners. **Code:** https://github.com/AlaaLab/H-learner

Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300.

1 INTRODUCTION

We consider the problem of estimating conditional average treatment effects (CATE) from observational data (Johansson et al., 2016; Alaa and Van Der Schaar, 2017; Shalit et al., 2017; Wager and Athey, 2018; Künzel et al., 2019; Kennedy, 2020). Let $\mathcal{D} = \{(X_i, T_i, Y_i)\}_{i=1}^n$ denote the observational dataset, where $X_i \in \mathcal{X}$ is a feature vector, $T_i \in \{0, 1\}$ is a binary treatment indicator and $Y_i \in \mathbb{R}$ is the observed outcome of interest. Under the Neyman-Rubin potential outcomes framework (Neyman, 1923), let $Y_i(1)$ and $Y_i(0)$ denote the potential outcomes (POs) for individual i under treatment and control, respectively. The *fundamental problem of causal inference* is that we only observe one of the POs for each individual, i.e., $Y_i = T_iY_i(1)+(1-T_i)Y_i(0)$. We denote $\mu_0(x) = \mathbb{E}[Y(0) \mid X = x]$ and $\mu_1(x) = \mathbb{E}[Y(1) \mid X = x]$ as the expected POs under control and treatment. Our goal is to estimate the CATE function:

$$\tau(x) = \mathbb{E}[Y(1) - Y(0) \mid X = x], \tag{1}$$

using the dataset $\mathcal{D}$. This estimand is identifiable from the sample $\mathcal{D}$ under the following assumptions: *(1) Consistency:* $Y_i = Y_i(t)$, *(2) Unconfoundedness:* $(Y(0), Y(1)) \perp\!\!\!\perp T \mid X$, and *(3) Positivity:* $\forall x \in \mathcal{X}$, $0 < \pi(x) < 1$, where $\pi(x) = \mathbb{P}(T = 1 \mid X = x)$ is the propensity score (Rubin, 1974).

Learning the CATE function differs from standard supervised learning in several important ways. As noted by Curth and van der Schaar (2021), two key distinctions stand out: (1) *covariate shift* induced by confounding, where the feature distributions differ across treatment and control groups, i.e., $\mathbb{P}(X \mid T = 1) \neq \mathbb{P}(X \mid T = 0)$, and (2) the *inductive biases* associated with the goal of estimating a difference between two outcome functions, $\tau(x) = \mu_1(x) - \mu_0(x)$, rather than the individual POs, which introduces unique considerations in how we design and regularize the models for $\mu_0(x)$, $\mu_1(x)$, and $\tau(x)$. In this paper, we focus on the second aspect and propose a new strategy for regularizing CATE models.

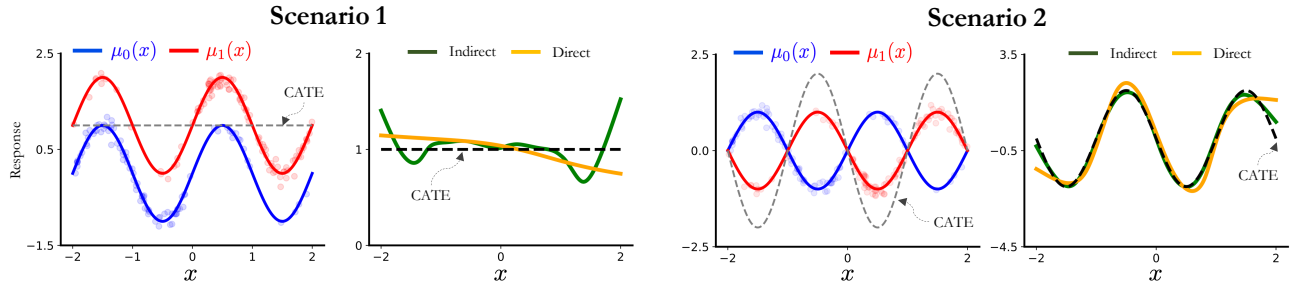

Figure 1: **Impact of inductive biases on CATE estimation.** (a) When CATE is simpler than POs, indirect learners introduce spurious heterogeneity due to independent regularization. (b) When CATE is as complex as POs, indirect learners perform better by accurately modeling each outcome surface.

To understand how regularization affects learning of CATE, consider the following illustrative example, with variants of this example appear in the literature (Künzel et al., 2019; Kennedy, 2020; Curth and van der Schaar, 2021). We define the POs as:

$$\mu_0(x) = \sin(\omega x), \quad \mu_1(x) = \sin(\omega x + \Delta) + \beta. \tag{2}$$

Here, the control and treatment outcome functions share the same sinusoidal shape and frequency, and differ only by a phase shift Δ and an amplitude shift β. As a result, both $\mu_0(x)$ and $\mu_1(x)$ have comparable complexity, i.e., both are equally easy—or difficult—to learn. However, the complexity of the resulting treatment effect function $\tau(x) = \mu_1(x) - \mu_0(x)$ can vary significantly depending on the values of Δ. We highlight this by considering the following two scenarios:

- **Scenario 1:** If $\Delta = 0$, the phase alignment implies the treatment effect is constant across all units: $\tau(x) = \mu_1(x) - \mu_0(x) = \beta, \forall x \in \mathbb{R}$. In this case, $\tau(x)$ is much simpler than either PO, both of which are nonlinear. This scenario is common in medicine, where prognostic factors (predicting outcomes) may be complex, while predictive factors (modulating treatment effects) are simpler or fewer.

- **Scenario 2:** If $\Delta > 0$, the treatment effect becomes heterogeneous: $\tau(x) = \mu_1(x) - \mu_0(x) = \beta + (\sin(\omega x + \Delta) - \sin(\omega x))$. Now, $\tau(x)$ inherits the full complexity of the POs. Though the individual POs have not changed in complexity, the induced CATE function is substantially more complex.

This example illustrates that the intrinsic difficulty in learning the treatment effect depends not just on the complexity of the POs, but also on the relationship between them. As such, choosing how to regularize CATE estimators requires careful consideration of which function—outcome or effect—can be better estimated in the given domain.

Künzel et al. (2019) coined the notion of "meta-learners" to describe a family of model-agnostic strategies for learning CATE. Broadly, these strategies fall into two categories (Curth and van der Schaar, 2021):

Indirect meta-learners use the datasets $\mathcal{D}_0 = \{(X_i, Y_i) : T_i = 0\}$ and $\mathcal{D}_1 = \{(X_i, Y_i) : T_i = 1\}$ to obtain intermediate models of the PO functions $\widehat{\mu}_t, t \in \{0, 1\}$, and then set $\widehat{\tau}(x) = \widehat{\mu}_1(x) - \widehat{\mu}_0(x)$.

Direct meta-learners target the CATE function directly by constructing a *pseudo-outcome* Y_φ based on some nuisance parameter φ (e.g., estimates of the propensity score $\widehat{\pi}(x)$ and the POs $\widehat{\mu}_t$, $t \in \{0, 1\}$), and then fit a model $\widehat{\tau}(x)$ for the CATE function using the constructed dataset $\{(X_i, Y_{\varphi,i})\}$.

Figure 1 shows how direct and indirect learners perform under both scenarios. In Scenario 1, where the CATE function is simpler than the PO, the direct learner outperforms the indirect approach. This is because indirect learners separately fit and regularize the PO models, and each model make errors in different regions of the covariate space. When the two models are subtracted to estimate the treatment effect, they can compound in unpredictable ways, introducing spurious heterogeneity into the CATE function. This phenomenon is known as "regularization-induced confounding" (Hahn et al., 2018). Conversely, in Scenario 2, where the CATE function is as complex as the POs, direct learners tend to underperform, as the benefit of targeting a simpler effect function no longer holds. In practice, however, it is difficult to know in advance which strategy will perform better, as their relative performance also depends on other factors underlying data generation processes (DGPs), including sample sizes, treatment imbalance, the degree of confounding and overlap.

To address these limitations, we propose the *Hybrid learner* (H-learner), a novel meta-learner regularization strategy for CATE estimation that generalizes both direct and indirect meta-learners. Our strategy formulates a proximal indirect learning task, where the model learns two intermediate functions, f_0 and f_1, which are not necessarily optimal estimators of the POs but are

trained such that their difference accurately approximates the treatment effect. Regularization is applied by weighting two competing objectives: encouraging f_0 and f_1 to closely approximate the true POs, and ensuring that their difference $f_1 - f_0$ aligns with a pseudo-outcome-based guess of CATE.

In the remainder of the paper, we describe the proposed H-learner regularization strategy and place it in the context of existing meta-learning paradigms. Our method is model-agnostic and compatible with a wide range of machine learning (ML) architectures. In particular, we make the following contributions:

1. We characterize the performance of meta-learners by identifying key factors of the underlying DGPs and explicitly linking them to the inductive biases introduced by their regularization strategies.
2. We introduce the H-learner as a unified approach that combines direct and indirect regularization and theoretically demonstrate that it achieves lower prediction risk by balancing the bias–variance trade-off inherent in existing meta-learners.
3. We empirically show that the H-learner consistently achieves state-of-the-art performance, remains robust across diverse DGP characteristics, and lies on the Pareto frontier of common benchmarks.

2 META-LEARNERS

In this section, we illustrate representative examples of indirect and direct meta-learners from the literature, and discuss the implicit inductive biases and regularization they impose.

2.1 Indirect Meta-learners

Given an observational dataset partitioned by the two treatment groups, $\mathcal{D}_0$ and $\mathcal{D}_1$, an indirect meta-learner estimates the PO models μ_0 and μ_1 by independently fitting outcome models on each group by minimizing a regularized empirical risk of the form:

$$\sum_{i=1}^{n} \ell\big(Y_i, \mu_{T_i}(X_i; \theta_{T_i})\big) + \lambda\, \mathcal{R}(\theta_0, \theta_1), \tag{3}$$

where ℓ is a supervised loss function (e.g., squared error), θ_0 and θ_1 are the parameters of the models for the control and treatment groups respectively, and $\mathcal{R}(\cdot)$ is a regularization term (e.g., ℓ_2 norm, shared representation penalty, or complexity constraint). The final CATE estimate is then given by the difference between these plug-in estimates of the PO functions: $\widehat{\tau}(x) = \mu_1(x; \widehat{\theta}_1) - \mu_0(x; \widehat{\theta}_0)$.

Several well-known methods fall under the category of indirect meta-learners. The *T-learner* fits two separate models $\mu_0(x; \theta_0)$ and $\mu_1(x; \theta_1)$ independently using the datasets $\mathcal{D}_0$ and $\mathcal{D}_1$, and estimates CATE as their difference. Another architecture is the *S-learner*, which fits a single model $\mu(x, t; \theta)$ by treating the treatment indicator as an additional feature, then estimating CATE by toggling this feature t, i.e., $\mu(x, 1; \widehat{\theta}) - \mu(x, 0; \widehat{\theta})$. Examples of these models include Bayesian additive regression trees (BARTs), regression trees and representation-based methods such as TARNet, which learn a shared latent representation of the features before fitting separate heads for each treatment arm (Hill, 2011; Athey and Imbens, 2016; Shalit et al., 2017).

However, because indirect learners optimize and regularize μ_0 and μ_1 independently, their inductive bias favors simpler functions that best fit $\mathcal{D}_0$ and $\mathcal{D}_1$. As a result, taking their difference can yield biased estimates and implicitly induce spurious treatment effect heterogeneity, particularly under DGPs where the CATE function is simpler than the POs (Hahn et al., 2018, 2020). This occurs because the models are not regularized with respect to the difference that defines the CATE, i.e., $\tau(x) = \mu_1(x) - \mu_0(x)$. Curth and van der Schaar (2021) further points out that this inductive bias not only risks introducing artefactual heterogeneity but also contradicts how treatment effect heterogeneity should be approached from a scientific standpoint—where the default (null) hypothesis is often that treatment effects are homogeneous (Crump et al., 2008; Ballman, 2015).

2.2 Direct Meta-learners

Direct meta-learners are typically implemented as two-stage procedures. In the first stage, a nuisance parameter φ—such as the propensity score or outcome regression—is estimated and used to construct a pseudo-outcome Y_φ. In the second stage, a single model is trained to estimate $\tau(x)$ using the pseudo-labeled dataset $\{(X_i, Y_{\varphi,i})\}$. The pseudo-outcome is designed so that it depends only on observables and satisfies, either exactly or approximately, the following condition:

$$E[Y_\varphi | X = x] = E[Y(1) - Y(0) | X = x],\ \forall x \in \mathcal{X}. \tag{4}$$

Once the pseudo-outcomes are constructed, the CATE is estimated via empirical risk minimization:

$$\sum_{i=1}^{n} \ell\big(Y_{\varphi,i}, \tau(X_i; \theta)\big) + \lambda\, \mathcal{R}(\theta), \tag{5}$$

where ℓ is a loss function and $\mathcal{R}(\theta)$ is a regularization applied directly to the CATE function. The general framework described above captures various methods in literature, including the X-learner, the Inverse propensity weighted (IPW) learner and the Doubly-robust (DR) learner (Table 1). When the propensity score is known, the IPW and DR pseudo-outcomes satisfy (4) exactly. The regularization strategy underlying direct meta-learners differs fundamentally from that of

indirect learners: rather than independently regularizing the PO models, direct learners apply training and regularization directly to the CATE function.

However, the effectiveness of this approach hinges on the quality of the pseudo-outcome transformation, i.e., how accurately it reflects the CATE function, and it is largely determined by factors underlying the DGP. When the CATE surface is higher-dimensional than the underlying POs, direct learners often yield poorer estimates as they target a harder function. Imbalanced propensities can further worsen pseudo-outcome constructions—such as those based on inverse propensity—which suffer from high variance and degrade performance (Curth and Van der Schaar, 2021).

	Pseudo-outcome
IPW-learner (Horvitz and Thompson, 1952)	$Y_\varphi = \frac{T-\pi(X)}{\pi(X)(1-\pi(X))}Y$
X-learner (Künzel et al., 2019)	$Y_\varphi = T(Y-\hat{\mu}_0(X)) + (1-T)(\hat{\mu}_1(X)-Y)$
DR-learner (Kennedy, 2020)	$Y_\varphi = \frac{T-\pi(X)}{\pi(X)(1-\pi(X))}(Y-\hat{\mu}_T(X)) + \hat{\mu}_1(X) - \hat{\mu}_0(X)$

Table 1: Direct meta-learners based on pseudo-outcome regression.

3 HYBRID META-LEARNERS

The key idea behind the proposed H-learner is to train a pair of models $f_0, f_1 \in \mathcal{F}$, where the CATE is estimated as their difference, $f_1(x) - f_0(x)$. However, unlike traditional indirect learners, which train f_0 and f_1 solely to minimize empirical risk with respect to the POs, the H-learner jointly optimizes these models to balance both accurate prediction of the POs and the accuracy of their implied CATE. As a result, the optimal finite-sample solutions $\hat{f}_0, \hat{f}_1 \in \mathcal{F}$ may not correspond to the best fits for the POs individually, but may instead favor a pair whose difference yields a more accurate estimate of the implied treatment effect function. Similar to direct meta-learners, the H-learner follows a two-stage procedure outlined below (Figure 2):

Stage 1: Construct a pseudo-outcome using a subset of the observational dataset $\mathcal{D}$ based on any of the transformations in Table 1. For instance, the X-learner pseudo-outcome can be constructed as:

$$Y_{\varphi,i} = T_i(Y_i - \hat{\mu}_0(X_i)) + (1-T_i)(\hat{\mu}_1(X_i) - Y_i), \quad (6)$$

where $\hat{\mu}_0$ and $\hat{\mu}_1$ are initial estimates of the POs for the control and treatment groups, respectively. These initial estimates can be obtained through any indirect learner, e.g., the T-learner described earlier.

Stage 2: Fit two intermediate functions $f_0, f_1 \in \mathcal{F}$ by jointly minimizing the empirical risk of each function in predicting the observed (factual) outcomes, along with the empirical risk of their difference in approximating the pseudo-outcome constructed in **Stage 1**.

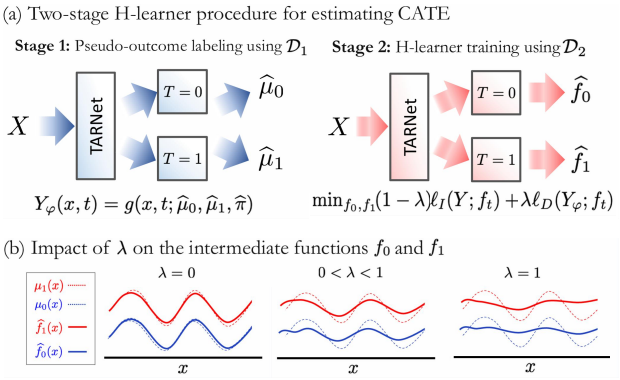

Figure 2: Pictorial depiction of the H-learner.

Formally, the H-learner loss function balances both POs prediction and treatment effect estimation, and is given by:

$$\min_{f_0,f_1} \sum_{i=1}^{n} (1-\lambda) \underbrace{\ell(Y_i, f_{T_i}(X_i))}_{\text{Indirect component } \ell_I} + \lambda \underbrace{\big((f_1(X_i) - f_0(X_i)) - Y_{\varphi,i}\big)^2}_{\text{Direct component } \ell_D}, \quad (7)$$

where $0 \leqslant \lambda \leqslant 1$, and we obtain the final CATE estimator as $\hat{\tau}(x) = \hat{f}_1(x) - \hat{f}_0(x)$.

The H-learner loss can be viewed as a regularized version of the indirect learning framework, with a regularization term that encourages the difference between the two intermediate models to align with a pseudo-outcome. Alternatively, it can be interpreted as a generalized regularization strategy that interpolates between direct learners ($\lambda = 1$) and indirect learners ($\lambda = 0$) while adaptively balancing the influence of each. In the following sections, we demonstrate that the optimal value of λ often depends on several factors of the underlying DGP, but in most cases, lies strictly within $\lambda \in (0,1)$ rather than at the boundary values. This reflects the benefit of leveraging both regularization strategies rather than relying on either one alone. Practically, we treat λ as a hyperparameter and adaptively tune its value by minimizing the validation loss. Details of this validation procedure are provided in Appendix A.

The H-learner procedure is compatible with any existing CATE estimation architecture that provides explicit estimates of the POs. In our implementation, we use the TARNet architecture both to estimate the nuisance components required for constructing pseudo-outcomes and to fit the intermediate models used to derive the final CATE estimate as illustrated in Figure 2(a). Further implementation details with a pseudo-algorithm can be found in the Appendix A.

4 RELATED WORK

4.1 Inconsistent Performance of Indirect and Direct Learners

The divergence in performance between indirect and direct learners has been consistently observed across settings in the literature. In particular, prior work has shown that no single meta-learner consistently outperforms others across all settings (Künzel et al., 2019). Both theoretical and empirical studies show that their relative performance varies substantially with the sparsity and smoothness of the response surfaces (Curth and Van der Schaar, 2021). Acharki et al. (2023) extended the comparison of meta-learners to the non-binary treatment setting, theoretically highlighting the bias–variance trade-off in the multi-treatment setting. These inconsistencies highlight the importance of developing a more robust approach, as the true DGP is often unknown in practice, and relying on either method can result in unreliable performance.

4.2 Regularization Strategies for CATE Estimations

Several other regularization strategies have been proposed to mitigate "regularization-induced confounding". *Weight regularization* penalizes discrepancies between the weights of the outcome heads to encourage parameter sharing (Hahn et al., 2018). *Reparameterization* implicitly introduces regularization by modeling $\mu_1(x) = \mu_0(x)+\tau(x)$, allowing the treatment effect $\tau(x)$ to be predicted directly by one head as an offset from $\mu_0(x)$ (Imai and Ratkovic, 2013; Curth and van der Schaar, 2021). *Structural regularization* was proposed with FlexTENet, a specialized neural architecture that enforces regularization through flexible feature sharing across layers (Curth and van der Schaar, 2021).

Comparison with H-learner: H-learner formulation easily reveals the limitations of these regularization strategies. Penalizing weight differences between f_0 and f_1, or enforcing shared layers when implementing these functions with neural networks, effectively reduces the output difference $f_1(x) - f_0(x)$, which is equivalent to applying H-learner with $Y_\varphi = 0$. This reveals their key weakness: such regularization imposes a strong inductive bias toward scenarios with small treatment effects. As a result, naively applying these approaches can degrade performance when this assumption is violated.

5 THEORETICAL ANALYSIS

In this section, we analyze the statistical properties of the H-learner. We focus on the linear setup, where the H-learner admits a closed-form solution. We then characterize conditions under which H-learner achieves strictly lower prediction risk than either estimator alone. Extending to nonlinear function classes does not yield a tractable closed form, so we defer to experiments in Section 6 for empirical validation. All proofs are deferred to Appendix B.

5.1 Closed-Form Solution of the H-Learner in Linear Models

Throughout this section, we use $X \in \mathbb{R}^{n\times d}$ to denote the covariate matrix and $X_1 \in \mathbb{R}^{n_1\times d}$ and $X_0 \in \mathbb{R}^{n_0\times d}$ denote the submatrices of X corresponding to treated and control groups. Let $\tau(x)$ denote the true CATE, and define $\theta^\star = \arg\min_\theta \mathbb{E}\big[(x^\top\theta - \tau(x))^2\big]$ as the true treatment effect parameter in the linear $\tau(x)$ case, or the best linear projection of $\tau(x)$ onto the span of x in the nonlinear case over the feature space. We denote by $\hat\theta_{\text{ind}}$, $\hat\theta_{\text{dir}}$, and $\hat\theta_H$ the OLS parameter estimates produced by the indirect learner, direct learner, and H-learner, respectively. For ease of notation, let $G_1 = X_1^\top X_1$, $G_0 = X_0^\top X_0$, $G = X^\top X$.

Theorem 5.1 *(Exact H-learner).* Under the linear setup, the H-learner OLS estimator admits a closed-form solution given by a matrix-weighted combination of the indirect and direct OLS estimators.

$$\hat\theta_H = (I - W)\hat\theta_{\text{ind}} + W\hat\theta_{\text{dir}},$$

where $W = \lambda A\big[(1-\lambda)I+\lambda A\big]^{-1}$, $A := (G_1^{-1}+G_0^{-1})\,G$.

Theorem 5.1 shows that the H-learner forms a matrix-weighted linear combination with a weight that depends on the regularization parameter λ and the geometry encoded in the matrix A. When $\lambda = 0$, the H-learner reduces to the indirect learner, and when $\lambda = 1$, it equals the direct learner. In general, this weighting operates in a direction-wise manner across the feature space. Let $\{\mu_i\}_{i=1}^d$ be the eigenvalues of A. Then the eigenvalues of W are $\left\{\frac{\lambda\mu_i}{(1-\lambda)+\lambda\mu_i}\right\}_{i=1}^d$, which are strictly increasing in both λ and μ_i. This highlights the adaptivity of the H-learner: Under poor overlap or in unbalanced settings with a small treatment arm, the eigenvalues of A tend to increase, as the Gram matrix G_1 or G_0 corresponding to the smaller treatment group becomes ill-conditioned. Consequently, the eigenvalues of W increase, shifting more weight toward the direct approach, where indirect learners are more susceptible to bias induced by separate regularization.

We next analyze how this interpolation translates into improved prediction risk through a bias–variance decomposition.

5.2 Bias–Variance Analysis of the H-Learner

Let $e_{\text{ind}} = \hat{\theta}_{\text{ind}} - \theta^\star$ and $e_{\text{dir}} = \hat{\theta}_{\text{dir}} - \theta^\star$ denote the estimation errors of the indirect and direct learners, respectively. We then define their biases $b_{\text{ind}} = \mathbb{E}[e_{\text{ind}}]$ and $b_{\text{dir}} = \mathbb{E}[e_{\text{dir}}]$, and their covariance matrices $\Sigma_{\text{ind}} = \text{Var}(e_{\text{ind}})$, $\Sigma_{\text{dir}} = \text{Var}(e_{\text{dir}})$, $\Sigma_{\text{ind,dir}} = \text{Cov}(e_{\text{ind}}, e_{\text{dir}})$. The mean squared error (MSE) can be expressed as $\text{MSE}_{\text{ind}} = \mathbb{E}[\|e_{\text{ind}}\|_2^2]$ and $\text{MSE}_{\text{dir}} = \mathbb{E}[\|e_{\text{dir}}\|_2^2]$.

To facilitate the analysis, we start by focusing on the scalar path $W = \omega I$, which leads to closed-form conditions under which the H-learner outperforms both endpoints. Since $\min_W \text{MSE}_H(W) \leqslant \min_{\omega\in[0,1]} \text{MSE}_H(\omega I)$, any improvement achieved along this scalar path provides a sufficient condition, as the full matrix optimization yields no higher risk and thus will also improve upon both endpoints.

Assumption 5.2. We assume that the indirect and direct learners are trained via cross-fitting (e.g., on independent folds), so that $\Sigma_{\text{ind,dir}} = 0$.

Theorem 5.3 *(MSE expansion and optimal weight).* Under Assumptions 5.2, the MSE of the H-learner estimator is

$$\text{MSE}_{\text{H}} = \mathbb{E}\big[\|\hat{\theta}_H - \theta^\star\|_2^2\big] = (1-\omega)^2\,\text{MSE}_{\text{ind}} + \omega^2\,\text{MSE}_{\text{dir}} + 2\omega(1-\omega)\,D,$$

where $D = b_{\text{ind}}^\top b_{\text{dir}}$.

The optimal convex weight that minimizes MSE_{H} is

$$\omega^* = \min\left\{1,\ \max\left\{0,\ \frac{\text{MSE}_{\text{ind}} - D}{\text{MSE}_{\text{ind}} + \text{MSE}_{\text{dir}} - 2D}\right\}\right\}.$$

Theorem 5.3 decomposes the H-learner's MSE into a weighted sum of the direct and indirect errors together, and characterizes the optimal weight ω^*. Consequently, the H-learner strictly outperforms both learners whenever $0 < \omega^* < 1$. We next present two corollaries that provide sufficient conditions for this to hold.

Corollary 5.4. If $b_{\text{ind}}^\top b_{\text{dir}} < 0$, then $\text{MSE}_{\text{H}} < \min\{\text{MSE}_{\text{ind}},\ \text{MSE}_{\text{dir}}\}$.

Corollary 5.5. If $\|b_{\text{ind}}\|_2 \geqslant \|b_{\text{dir}}\|_2$ and $\text{tr}(\Sigma_{\text{dir}}) > b_{\text{dir}}^\top(b_{\text{ind}} - b_{\text{dir}})$, then $\text{MSE}_{\text{H}} < \min\{\text{MSE}_{\text{ind}},\ \text{MSE}_{\text{dir}}\}$.

Corollary 5.4 shows that when the biases point in opposite directions, the H-learner benefits from their cancellation. When they point in the same direction, Corollary 5.5 makes explicit the bias–variance trade-off between indirect and direct learners. It implies that whenever the indirect learner has a larger bias and the direct learner's variance is sufficiently large relative to this bias gap, then the H-learner strictly outperforms both individual estimators.

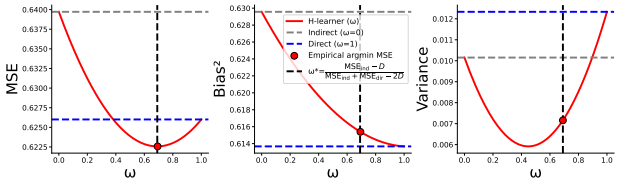

Figure 3: MSE and bias-variance decomposition of H-learner across convex weight ω. Empirical optimum $\omega^\star$ aligns with the theoretical optimal point. H-learner with $\omega^\star$ achieves lower MSE than either baseline.

We further illustrate this with a 1D example in Figure 3. The plot shows the MSE of the H-learner as a function of the convex weight ω, together with its decomposition into bias and variance. As expected, the indirect learner exhibits higher bias, while the direct learner suffers from higher variance. H-learner achieves strictly lower MSE than either baseline by reducing variance below both while maintaining a small level of bias.

6 EXPERIMENTS

In this section, we evaluate the performance of H-learner through semi-synthetic experiments and established benchmarks. Section 6.1 outline the datasets and experimental setup. Section 6.2 highlight the inconsistent performance between indirect and direct learners using custom-designed semi-synthetic setups and demonstrate the robustness of H-learner. In Section 6.3, we present results on widely used benchmark datasets, further highlighting the improved performance of H-learner over existing baselines.

6.1 Setup

Datasets. We evaluate performance using two widely adopted benchmark datasets: IHDP and ACIC 2016, both commonly used in the literature on heterogeneous treatment effect estimation (Shalit et al., 2017; Alaa and Van Der Schaar, 2017; Shi et al., 2019; Lee et al., 2020; Curth and Van der Schaar, 2021). Following prior work, we use a 63/27/10 train/validation/test split for both datasets.

IHDP. The IHDP dataset is based on covariates from the Infant Health and Development Program, a randomized study assessing the effect of specialist home visits on cognitive outcomes for premature infants (Hill, 2011). Confounding and treatment imbalance are introduced by removing a subset of treated units. The resulting dataset includes 747 observations with 25 covariates. Following prior work, we evaluate performance by averaging over 1,000 realizations from setting "A" implemented in the NPCI package (Dorie, 2016).

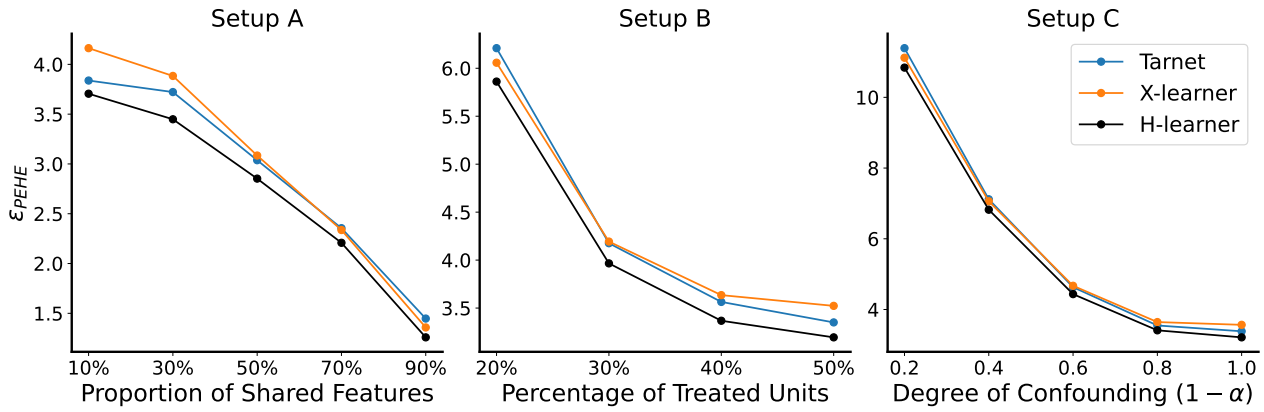

Figure 4: PEHE results for three semi-synthetic scenarios. In all setups, we observe performance trade-offs between TARNet (indirect learner) and X-learner (direct learner), while the proposed H-learner consistently outperforms both counterparts.

ACIC 2016. The ACIC 2016 datasets are derived from the Collaborative Perinatal Project and were introduced as part of the Atlantic Causal Inference Competition (Dorie et al., 2019). This benchmark provides a comprehensive evaluation that varies in functional complexity, confounding, overlap, and treatment effect heterogeneity. It includes 77 distinct DGPs, with 4,802 observations and 79 covariates. Following (Lee et al., 2020), we remove rows where the POs below the 1st and above the 99th percentile to eliminate outliers and evaluate performance by averaging over 5 random realizations per setting.

Evaluations. We evaluate performance using *Precision in Estimation of Heterogeneous Effect* (PEHE) (Hill, 2011). The PEHE loss is defined as $\epsilon_{\text{PEHE}} = \frac{1}{n}\sum_{i=1}^{n}(\hat{\tau}(x_i) - \tau(x_i))^2$, which corresponds to the mean squared error in estimating the CATE. We report performance on both in-sample (training and validation) and out-of-sample (test) data. Note that in-sample evaluation is non-trivial, as only the factual outcomes are observed during training, while the true CATE $\tau(x)$ is never observed.

6.2 Semi-Synthetic Experiments

Real-world datasets often lead to inconsistent performance between direct and indirect learners due to a mixture of underlying factors. To disentangle these effects, we design semi-synthetic DGPs that isolate and examine three key sources of variation: (A) Relative complexity of POs and CATE, (B) Imbalance between treated and control units, and (C) Strength of confounding.

Experiment setup. In all setups, we use covariates from the IHDP dataset ($n = 747,\ d = 25$) but design custom response surfaces as follows.

Let S_0 and S_1 denote sets of randomly selected features used to construct $\mu_0(x)$ and $\mu_1(x)$, respectively, and let $S = S_0 \cup S_1$. We then simulate the response surfaces and the treatment assignment as follows:

$$Y_i = 2\sum_{j\in S_{T_i}} X_j + 2\sum_{j\in S_{T_i}} X_j^2 + \sum_{j,k\in S_{T_i}} X_j X_k + \varepsilon_i,$$
$$\mathbb{P}(T_i = 1 \mid X_i) = \sigma\Big(\alpha \cdot \sum_{j\in S} \beta_j X_j\Big),$$

where $\varepsilon_i, \beta_j \sim \mathcal{N}(0,1)$, $T_i \sim \text{Bernoulli}(\mathbb{P}(T_i = 1 \mid X_i))$, and $\sigma(\cdot)$ denotes the sigmoid function.

In Setup A, we vary the relative complexity of the CATE and PO functions by changing the proportion of shared features between S_0 and S_1 in $\{10\%, 30\%, 50\%, 70\%, 90\%\}$, where a greater number of shared features implies a simpler CATE function. In Setup B, we vary group imbalance by adjusting the proportion of treated units in the dataset to $\{20\%, 30\%, 40\%, 50\%\}$, where a higher percentage indicates more balanced treatment and control groups. In Setup C, we introduce varying levels of confounding in the treatment assignment function by adjusting $\alpha \in \{0, 0.2, 0.4, 0.6, 0.8\}$, where larger values of α correspond to stronger confounding. All simulations are averaged over 80 runs, with features in S_0 and S_1 randomly selected in each run.

Results. Figure 4 presents results from the three simulated scenarios. We show the performance of the H-learner using X-learner pseudo-outcomes in its first stage and compare it against its constituent components: the X-learner (direct learner) and TARNet (indirect learner). Results using alternative learners are provided in Appendix C.

Setup A shows the performance trade-off through the *relative complexity of the POs and the CATE*. As

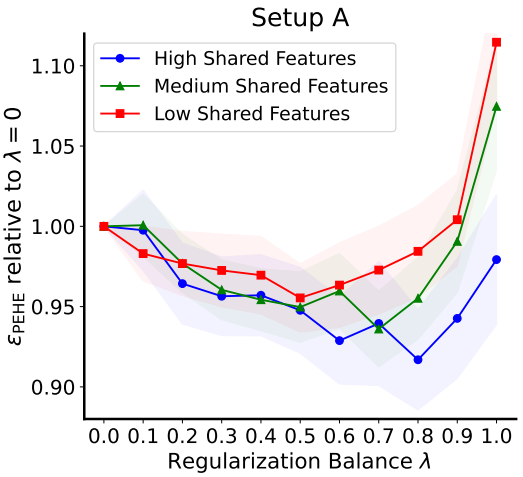

Figure 5: Performance of H-learner across different λ values for Setup A. Optimal performance is consistently achieved at intermediate λ values.

the proportion of shared features increases, the PO functions become more similar, which leads to a simpler CATE surface. Correspondingly, we observe that the X-learner only outperforms TARNet when the proportion of shared features is high, as its constructed pseudo-outcomes target a simpler function, but underperforms when the shared feature proportion is low.

Setup B demonstrates the performance trade-off driven by the *treatment–control imbalance*. As the proportion of treated units decreases, the imbalance in sample sizes between the treatment and control groups becomes more severe. This imbalance disproportionately affects indirect learners, as their separate outcome models are trained on increasingly unbalanced samples and are more susceptible to "regularization-induced confounding". Empirically, TARNet performs better under balanced treatment but underperforms when only 20% of units receive treatment.

Setup C demonstrates the performance trade-off driven by the *strength of confounding*. As the degree of confounding increases, the feature distributions differ more significantly across treatment groups. This poses greater challenges for indirect learners, as their separate outcome models are trained on increasingly divergent data distributions and suffer from reduced sample efficiency. As expected, TARNet performs well only under weaker confounding, but fails to match the X-learner when confounding is stronger.

In contrast, across all varying setups, the H-learner consistently outperforms both its direct and indirect counterparts by adjusting its inductive bias to better align with the underlying data, demonstrating robustness by unifying the strengths of both learners.

Table 2: Within-sample and out-of-sample PEHE results on benchmark datasets IHDP and ACIC 2016. Entries marked with * indicate statistically significant improvements over all baselines based on one-sided paired t-tests.

	IHDP		ACIC 2016	
	In-sample $\sqrt{\epsilon_{\text{PEHE}}}$	Out-of-sample $\sqrt{\epsilon_{\text{PEHE}}}$	In-sample $\sqrt{\epsilon_{\text{PEHE}}}$	Out-of-sample $\sqrt{\epsilon_{\text{PEHE}}}$
T-learner	$0.97 \pm .02$	$1.07 \pm .03$	$2.08 \pm .04$	$2.19 \pm .04$
TARNet	$0.81 \pm .02$	$0.93 \pm .03$	$1.66 \pm .03$	$1.77 \pm .04$
IPW-learner	$4.93 \pm .19$	$4.95 \pm .20$	$2.69 \pm .07$	$2.68 \pm .07$
DR-learner	$0.98 \pm .02$	$1.08 \pm .02$	$1.63 \pm .04$	$1.75 \pm .04$
X-learner	$0.79 \pm .01$	$0.91 \pm .02$	$1.58 \pm .03$	$1.70 \pm .04$
TARNet-WR	$1.19 \pm .03$	$1.33 \pm .04$	$1.89 \pm .04$	$1.97 \pm .04$
OffsetNet	$2.11 \pm .08$	$2.18 \pm .09$	$2.10 \pm .05$	$2.15 \pm .05$
FlexTENet	$1.18 \pm .03$	$1.30 \pm .04$	$1.81 \pm .04$	$1.89 \pm .04$
H-learner (DR)	$0.81 \pm .01$	$0.91 \pm .02$	$1.57 \pm .03$	$1.68 \pm .04$
H-learner (X)	$\mathbf{0.78 \pm .01}^\star$	$\mathbf{0.88 \pm .02}^\star$	$\mathbf{1.56 \pm .03}^\star$	$\mathbf{1.67 \pm .04}^\star$

Effect of regularization parameter λ. We further investigate the performance of H-learner across different λ values. Figure 5 shows how the test PEHE varies with λ in Setup A, across different levels of feature sharing: low (10%), medium (50%), and high (90%).

Results. Figure 5 shows that the optimal value of λ consistently lies between the two extremes—$\lambda = 0$ (indirect learner) and $\lambda = 1$ (direct learner)—further highlighting that combining both regularizations yields better performance than relying on either alone. Across different levels of feature sharing, the H-learner with the best-performing λ consistently improves upon the weaker learner by at least 10% and the stronger learner by at least 5%. Moreover, the trend in the optimal λ reflects the inductive bias inherent in the data: as more features are shared, the underlying CATE function becomes simpler and tends to favor direct regularization. We observe that the H-learner adapts to this inductive bias by yielding a larger optimal λ as the number of shared features increases. Similar performance results in Setups B and C are provided in Appendix C.

6.3 Benchmark Evaluation

Here we present results on the IHDP and ACIC 2016 benchmark datasets, comparing the H-learner with baseline meta-learners and regularization strategies from the literature.

Baselines. We consider T-learner and TARNet as baselines for indirect learners, and X-learner, DR-learner, and IPW-learner as baselines for direct learners. Additionally, we include three state-of-the-art regularization strategies: (1) *Weight Regularization*: TARNet-WR, a variant of TARNet with an explicit regularization that penalizes differences between the weights of the two output heads (Hahn et al., 2018; Curth and van der Schaar, 2021). (2) *Reparametrization:* Off-

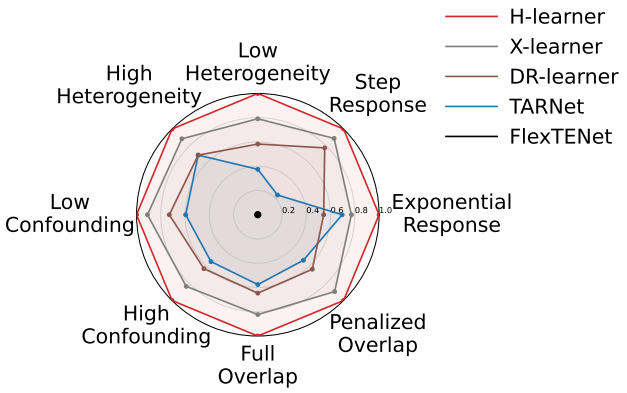

Figure 6: Radar chart of ACIC performance by DGP characteristics. PEHE is min–max normalized so that higher values indicate better performance, with 1 denoting the best. Only the top method per class is shown. H-learner performs best across all settings.

setNet, a neural network–based indirect learner that achieves implicit regularization through the reparameterization $\mu_1(x) = \mu_0(x) + \tau(x)$, enabling direct estimation of $\tau(x)$ as an offset from $\mu_0(x)$ (Imai and Ratkovic, 2013; Curth and van der Schaar, 2021). (3) *Structural Regularization:* FlexTENet, a neural network architecture that imposes regularization through flexible feature sharing across layers (Curth and van der Schaar, 2021).

Results. As shown in Table 2, H-learner outperforms all other baselines in terms of PEHE on both datasets. The best results are achieved when H-learner constructs X-learner pseudo-outcomes in its first stage.

Comparison to direct and indirect learners: Among indirect learners, TARNet outperforms the T-learner due to its shared representation and improved sample efficiency. Among direct learners, the X-learner outperforms both the IPW-learner and DR-learner, which suffer from higher variance. Notably, H-learner consistently outperforms both its indirect and direct counterparts on both datasets when constructing pseudo-outcomes from either the DR-learner or X-learner in the first stage.

Comparison to other regularization strategies: Note that all other existing regularization strategies fail to perform well on these benchmarks. As discussed in Section 4.2, these strategies rely on the assumption that reparameterization or sharing of weights simplifies the learning problem—an assumption that only holds when the treatment effect is small. Since the assumption fails to hold in practical scenarios, using these strategies can actually harm performance. For example, TARNet with weight regularization performs worse than the standard TARNet model. This further highlights the robustness of H-learner in handling diverse data conditions without relying on any structural assumptions.

Performance breakdown by DGP characteristics: We further analyze performance on ACIC 2016 by decomposing results along specific DGP characteristics, as summarized in Figure 6. The H-learner consistently achieves the best performance across a wide range of settings, including variations in response models, treatment effect heterogeneity, confounding strength, and covariate overlap. Complete results are provided in Appendix D.

7 DISCUSSION

Meta-learning approaches remain a prominent strategy for estimating CATE due to their intuitive design and strong empirical performance. However, achieving robustness across diverse DGPs requires a deeper understanding of how different meta-learners encode inductive biases through their regularization strategies. Indirect learners regularize models for POs, while direct learners directly regularize the CATE function by constructing pseudo-outcome estimators. These differing inductive biases make certain strategies more effective under specific DGPs while failing on others.

H-learner introduces a generalized regularization strategy that interpolates between indirect and direct regularizations while adaptively tuning the appropriate strength of each. This adaptivity enables H-learner to adjust its inductive bias to better match the underlying structure of the data without requiring prior knowledge of the DGPs. Experiments show that H-learner effectively overcomes the trade-off between indirect and direct learners, and achieving robust performance.

H-learner also offers several promising directions for future extensions. Potential directions include integrating H-learner with more complex causal models, such as those involving hidden confounding or multi-level treatments. Further, our experiments are currently limited to neural network implementations, extending H-learner to other ML models is left for future work.

Acknowledgements

This work was supported by a Hellman Fellowship award from the Society of Hellman Fellows and the Evergreen Foundation.

References

Naoufal Acharki, Ramiro Lugo, Antoine Bertoncello, and Josselin Garnier. Comparison of meta-learners for estimating multi-valued treatment heterogeneous effects. In *International conference on machine learning*, pages 91–132. PMLR, 2023.

Ahmed Alaa and Mihaela Van Der Schaar. Validating causal inference models via influence functions. In *International Conference on Machine Learning*, pages 191–201. PMLR, 2019.

Ahmed M Alaa and Mihaela Van Der Schaar. Bayesian inference of individualized treatment effects using multi-task gaussian processes. *Advances in neural information processing systems*, 30, 2017.

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

Karla V Ballman. Biomarker: predictive or prognostic? *Journal of clinical oncology: official journal of the American Society of Clinical Oncology*, 33(33):3968–3971, 2015.

Richard K Crump, V Joseph Hotz, Guido W Imbens, and Oscar A Mitnik. Nonparametric tests for treatment effect heterogeneity. *The Review of Economics and Statistics*, 90(3):389–405, 2008.

Alicia Curth and Mihaela van der Schaar. On inductive biases for heterogeneous treatment effect estimation. *Advances in Neural Information Processing Systems*, 34:15883–15894, 2021.

Alicia Curth and Mihaela Van der Schaar. Nonparametric estimation of heterogeneous treatment effects: From theory to learning algorithms. In *International Conference on Artificial Intelligence and Statistics*, pages 1810–1818. PMLR, 2021.

Vincent Dorie. Npci: Non-parametrics for causal inference. *URL: https://github. com/vdorie/npci*, 11:23, 2016.

Vincent Dorie, Jennifer Hill, Uri Shalit, Marc Scott, and Dan Cervone. Automated versus do-it-yourself methods for causal inference: Lessons learned from a data analysis competition. *Statistical Science*, 2019.

P Richard Hahn, Carlos M Carvalho, David Puelz, and Jingyu He. Regularization and confounding in linear regression for treatment effect estimation. 2018.

P Richard Hahn, Jared S Murray, and Carlos M Carvalho. Bayesian regression tree models for causal inference: Regularization, confounding, and heterogeneous effects (with discussion). *Bayesian Analysis*, 15(3):965–1056, 2020.

Jennifer L Hill. Bayesian nonparametric modeling for causal inference. *Journal of Computational and Graphical Statistics*, 20(1):217–240, 2011.

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

Kosuke Imai and Marc Ratkovic. Estimating treatment effect heterogeneity in randomized program evaluation. *The Annals of Applied Statistics*, pages 443–470, 2013.

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

Edward H Kennedy. Towards optimal doubly robust estimation of heterogeneous causal effects. *arXiv preprint arXiv:2004.14497*, 2020.

Sören R Künzel, Jasjeet S Sekhon, Peter J Bickel, and Bin Yu. Metalearners for estimating heterogeneous treatment effects using machine learning. *Proceedings of the national academy of sciences*, 116(10):4156–4165, 2019.

Hyun-Suk Lee, Yao Zhang, William Zame, Cong Shen, Jang-Won Lee, and Mihaela van der Schaar. Robust recursive partitioning for heterogeneous treatment effects with uncertainty quantification, 2020. URL https://arxiv.org/abs/2006.07917.

Divyat Mahajan, Ioannis Mitliagkas, Brady Neal, and Vasilis Syrgkanis. Empirical analysis of model selection for heterogeneous causal effect estimation. *arXiv preprint arXiv:2211.01939*, 2022.

Jerzy S Neyman. On the application of probability theory to agricultural experiments. essay on principles. section 9.(tlanslated and edited by dm dabrowska and tp speed, statistical science (1990), 5, 465-480). *Annals of Agricultural Sciences*, 10:1–51, 1923.

Donald B Rubin. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of educational Psychology*, 66(5):688, 1974.

Uri Shalit, Fredrik D Johansson, and David Sontag. Estimating individual treatment effect: generalization bounds and algorithms. In *International conference on machine learning*, pages 3076–3085. PMLR, 2017.

Claudia Shi, David M. Blei, and Victor Veitch. Adapting neural networks for the estimation of treatment effects, 2019. URL https://arxiv.org/abs/1906.02120.

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.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes] (see Sections 3 and 5)
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes] (see Appendix A)
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes] (see Supplementary Material)

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes] (see Section 5)
   (b) Complete proofs of all theoretical results. [Yes] (see Appendix B)
   (c) Clear explanations of any assumptions. [Yes] (see Section 5)

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes] (see Appendix A and Supplementary Material)
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes] (see Appendix A)
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes] (see Section 6)
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes] (see Appendix A)

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Not Applicable]
   (b) The license information of the assets, if applicable. [Not Applicable]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]
   (d) Information about consent from data providers/curators. [Not Applicable]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

A Implementations Details

A.1 Pseudo-algorithm of H-learner

Below, we present the pseudo-code for the H-learner.

Algorithm 1: H-learner

Input : $\mathcal{D} = \{(X_i, T_i, Y_i)\}_{i=1}^n$
Output : $\hat{\theta}(x) = \hat{f}_1(x) - \hat{f}_0(x)$

Split $\mathcal{D}$ *into* $\mathcal{D}_1$ *and* $\mathcal{D}_2$;
Stage 1: Pseudo-outcome construction ;
Estimate $\hat{\varphi} = (\hat{\pi}, \hat{\mu}_0, \hat{\mu}_1)$ using $\mathcal{D}_1$;
Construct Y_φ via the options in Table 1 ;
Stage 2: Training the H-learner ;
Train $\hat{f}_0, \hat{f}_1$ on $\mathcal{D}_2$ using the loss in (7) ;
return $\hat{\theta}(x) = \hat{f}_1(x) - \hat{f}_0(x)$

A.2 Model Architectures and Training Details

We implement all models following the architecture used in (Shalit et al., 2017). Specifically, the representation network of H-learner consists of three fully connected layers, each with 200 units and exponential linear unit (ELU) activations. Each output head comprises two additional dense layers with 100 units, followed by a final prediction layer. We implement similar architectures for the other meta-learners to ensure a fair comparison. All models are trained for 1000 epochs using the AdamW optimizer with mini-batches of size 100 and a cosine annealing learning rate schedule. We perform a grid search over learning rates {0.0001, 0.0005, 0.001}, and checkpoint achieving the lowest validation loss is used for final evaluation. We detail the validation criteria in the following section.

While we propose the H-learner with sample splitting using $\mathcal{D}_1$ and $\mathcal{D}_2$ for first and second stages as is commonly done in the literature, we found that using the full dataset for both stages often yields better empirical performance, particularly in small-sample settings. Similar observations have been made in Curth and Van der Schaar (2021), where sample splitting is then omitted. We therefore do not apply sample splitting and instead train both stages using the full dataset $\mathcal{D}$. Experiments were conducted on machine with an AMD EPYC 7543 32-Core Processor.

A.3 Hyperparameter Tuning

Hyperparameter tuning in CATE estimation is inherently challenging, as the true CATE is never observed, making standard supervised validation infeasible. To address this, we impute the unobserved counterfactual outcome using first-stage estimates $\hat{\mu}_0(x)$ and $\hat{\mu}_1(x)$, and use the following proxy loss on the validation set to select the best model checkpoint and hyperparameter:

$$\widehat{\text{PEHE}}(f_0, f_1) = \frac{1}{n} \sum_{i=1}^{n} \left[(f_1(x_i) - f_0(x_i)) - (t_i(y_i - \hat{\mu}_0(x_i)) + (1 - t_i)(\hat{\mu}_1(x_i) - y_i)) \right]^2 .$$

However, the above mechanism cannot be used to select the regularization strength λ, since imputing the unobserved counterfactual outcomes using first-stage estimates produces X-learner pseudo-outcomes $Y_{\varphi,i} = T_i(Y_i - \hat{\mu}_0(X_i)) + (1 - T_i)(\hat{\mu}_1(X_i) - Y_i)$. As a result, when H-learner constructs X-learner pseudo-outcomes in its first stage, this validation loss will be biased toward favoring no indirect regularization (i.e., $\lambda = 1$).

To address this issue, we instead estimate the POs by training outcome models on the **validation set**, yielding $\check{\mu}_0(X_i)$ and $\check{\mu}_1(X_i)$. We then select the value of λ that minimizes the following validation loss: $\frac{1}{n} \sum_{i=1}^{n} \left[(f_1(x_i) - f_0(x_i)) - (t_i(y_i - \check{\mu}_0(x_i)) + (1 - t_i)(\check{\mu}_1(x_i) - y_i)) \right]^2$. Similar strategies have also been employed in prior works (Alaa and Van Der Schaar, 2019; Mahajan et al., 2022). We compare the λ chosen by our validation-based strategy with the ground-truth optimal λ^* (minimizing test PEHE) in Table 3, showing that the validation-selected λ closely tracks the optimal value. Note that all validations are performed using the same validation set as other meta-learners, ensuring H-learner uses the same amount of information for fair comparisons.

Feature Overlap Ratio	λ^* (Test PEHE)	λ (Validation X-score)
0.1	0.49	0.52
0.3	0.54	0.57
0.5	0.57	0.58
0.7	0.57	0.58
0.9	0.67	0.62

Table 3: Comparison between the optimal λ^* minimizing test PEHE and the λ selected using the X-score on the validation set in the first synthetic setup of Section 6.2.

B Proof and Additional Theoretical Analysis

B.1 Complete Proof for the H-Learner Solution in Linear Models

Setup. Let $X \in \mathbb{R}^{n\times d}$ denote the covariate matrix, $T \in \{0,1\}^n$ denote the treatment vector, and $Y \in \mathbb{R}^n$ denote the observed outcomes. Let $X_1 \in \mathbb{R}^{n_1\times d}$ and $X_0 \in \mathbb{R}^{n_0\times d}$ denote the submatrices of X corresponding to treated ($T=1$) and control ($T=0$) units, and similarly $Y_1 \in \mathbb{R}^{n_1}$ and $Y_0 \in \mathbb{R}^{n_0}$. Let $Y_\phi \in \mathbb{R}^n$ be the pseudo-outcomes. For ease of notation, let $G_1 = X_1^\top X_1,\ G_0 = X_0^\top X_0,\ G = X^\top X,\ S_1 = X_1^\top Y_1,\ S_0 = X_0^\top Y_0,\ S_\phi = X^\top Y_\phi$.

Indirect and direct estimators. The indirect learner fits separate linear models to both groups:

$$\hat\theta^{(1)}_{\text{ind}} = (X_1^\top X_1)^{-1}X_1^\top Y_1 = G_1^{-1}S_1, \qquad \hat\theta^{(0)}_{\text{ind}} = (X_0^\top X_0)^{-1}X_0^\top Y_0 = G_0^{-1}S_0, \qquad \hat\theta_{\text{ind}} = \hat\theta^{(1)}_{\text{ind}} - \hat\theta^{(0)}_{\text{ind}}$$

The direct estimator from the pseudo-outcome regression is

$$\hat\theta_{\text{dir}} = (X^\top X)^{-1}X^\top Y_\phi = G^{-1}S_\phi$$

H-learner. We study the H-learner setup by modeling f_1 and f_0 as linear models with parameters $\theta_1, \theta_0 \in \mathbb{R}^d$:

$$f_1(x) = \theta_1^\top x, \qquad f_0(x) = \theta_0^\top x, \qquad f_1(x) - f_0(x) = (\theta_1 - \theta_0)^\top x = \hat\theta_H^\top x$$

We minimize the H-learner objective

$$L(\theta_1,\theta_0) = (1-\lambda)\big(\|Y_1 - X_1\theta_1\|_2^2 + \|Y_0 - X_0\theta_0\|_2^2\big) + \lambda\,\|X(\theta_1-\theta_0) - Y_\phi\|_2^2$$

Taking gradients and setting to zero yields

$$\frac{\partial L}{\partial \theta_1} = -2(1-\lambda)X_1^\top(Y_1 - X_1\theta_1) + 2\lambda X^\top(X(\theta_1-\theta_0) - Y_\phi) = 0$$
$$\frac{\partial L}{\partial \theta_0} = -2(1-\lambda)X_0^\top(Y_0 - X_0\theta_0) - 2\lambda X^\top(X(\theta_1-\theta_0) - Y_\phi) = 0$$

Rearrange:

$$(1-\lambda)G_1\theta_1 + \lambda G(\theta_1-\theta_0) = (1-\lambda)S_1 + \lambda S_\phi \tag{1}$$
$$(1-\lambda)G_0\theta_0 - \lambda G(\theta_1-\theta_0) = (1-\lambda)S_0 - \lambda S_\phi \tag{2}$$

Multiply (1) by G_1^{-1} and (2) by G_0^{-1}:

$$(1-\lambda)\theta_1 + \lambda G_1^{-1}G(\theta_1-\theta_0) = (1-\lambda)G_1^{-1}S_1 + \lambda G_1^{-1}S_\phi,$$
$$(1-\lambda)\theta_0 - \lambda G_0^{-1}G(\theta_1-\theta_0) = (1-\lambda)G_0^{-1}S_0 - \lambda G_0^{-1}S_\phi$$

Subtract the second from the first:

$$\big[(1-\lambda)I + \lambda(G_1^{-1}+G_0^{-1})G\big](\theta_1-\theta_0) = (1-\lambda)(G_1^{-1}S_1 - G_0^{-1}S_0) + \lambda(G_1^{-1}+G_0^{-1})S_\phi$$

Then

$$\big[(1-\lambda)I + \lambda A\big]\hat{\theta}_H = (1-\lambda)\hat{\theta}_{\text{ind}} + \lambda A\hat{\theta}_{\text{dir}}, \quad \text{where } A = (G_1^{-1} + G_0^{-1})G$$

Hence,

$$\begin{aligned}\hat{\theta}_H &= \big[(1-\lambda)I + \lambda A\big]^{-1}\big[(1-\lambda)\hat{\theta}_{\text{ind}} + \lambda A\hat{\theta}_{\text{dir}}\big] \\ &= (1-\lambda)\big[(1-\lambda)I + \lambda A\big]^{-1}\hat{\theta}_{\text{ind}} + \lambda A\big[(1-\lambda)I + \lambda A\big]^{-1}\hat{\theta}_{\text{dir}}\end{aligned}$$

Therefore,

$$\hat{\theta}_H = (I - W)\hat{\theta}_{\text{ind}} + W\hat{\theta}_{\text{dir}}, \quad W = \lambda A\big[(1-\lambda)I + \lambda A\big]^{-1}$$

B.2 Bias–Variance Analysis of the H-Learner Estimator

B.2.1 Equivalence Between Parameter-Space MSE and CATE Prediction Error

Assumption (Isotropy). Throughout this analysis, we assume the covariates are isotropic, i.e.,

$$\Sigma_X := \mathbb{E}[XX^\top] = cI_d, \text{ for some constant } c > 0.$$

Case 1: When the true CATE function is linear $\tau(x) = x^\top\theta^\star$, then for any estimator $\hat{\theta}$, the CATE prediction error is

$$\begin{aligned}\mathbb{E}\big[(\hat{\tau}(X) - \tau(X))^2\big] &= \mathbb{E}\Big[(X^\top\hat{\theta} - X^\top\theta^\star)^2\Big] = \mathbb{E}\Big[(X^\top(\hat{\theta} - \theta^\star))^2\Big] \\ &= (\hat{\theta} - \theta^\star)^\top\,\mathbb{E}[XX^\top]\,(\hat{\theta} - \theta^\star) = (\hat{\theta} - \theta^\star)^\top\,\Sigma_X\,(\hat{\theta} - \theta^\star) \qquad (3)\end{aligned}$$

Under isotropy, $\Sigma_X = cI_d$, so (3) becomes

$$\mathbb{E}\big[(\hat{\tau}(X) - \tau(X))^2\big] = c\,\|\hat{\theta} - \theta^\star\|_2^2.$$

Therefore, comparing CATE error is equivalent (up to the constant) to comparing their parameter-space errors:

$$\mathbb{E}\big[(\hat{\tau}(X) - \tau(X))^2\big] \;\propto\; \mathbb{E}\Big[\|\hat{\theta} - \theta^\star\|_2^2\Big]$$

Case 2: When the true CATE function $\tau(x)$ is not linear, let

$$\theta^\star := \arg\min_\theta\; \mathbb{E}\big[(x^\top\theta - \tau(x))^2\big]$$

be its best linear projection. Then for any estimator $\hat{\theta}$,

$$\begin{aligned}\mathbb{E}\big[(x^\top\hat{\theta} - \tau(x))^2\big] &= \mathbb{E}\big[(x^\top\theta^\star - \tau(x))^2\big] + \mathbb{E}\big[(x^\top\hat{\theta} - x^\top\theta^\star)^2\big] \\ &= \underbrace{\mathbb{E}\big[(x^\top\theta^\star - \tau(x))^2\big]}_{\text{approximation error (independent of } \hat{\theta})} + c\,\mathbb{E}\big[\|\hat{\theta} - \theta^\star\|_2^2\big].\end{aligned}$$

Therefore, even in the nonlinear case, all estimator comparisons can be carried out entirely in parameter space via $\mathbb{E}\|\hat{\theta} - \theta^\star\|_2^2$. Hence, in the following sections, we compare estimators through their parameter-space MSE.

B.2.2 Mean Squared Error Decomposition for the H-Learner

Let $e_{\text{ind}} = \hat{\theta}_{\text{ind}} - \theta^\star, e_{\text{dir}} = \hat{\theta}_{\text{dir}} - \theta^\star$ denote the estimation errors of the indirect and direct estimators, denote the estimation errors of the indirect and direct estimators, where $\theta^\star \in \mathbb{R}^d$ is the true treatment-effect parameter in the linear CATE case, or the best linear projection as defined above in the nonlinear case.

Define their biases and covariances as

$$b_{\text{ind}} = \mathbb{E}[e_{\text{ind}}], \qquad b_{\text{dir}} = \mathbb{E}[e_{\text{dir}}], \qquad \Sigma_{\text{ind}} = \text{Var}(e_{\text{ind}}), \qquad \Sigma_{\text{dir}} = \text{Var}(e_{\text{dir}})$$

Then their mean squared errors can be written as

$$\mathrm{MSE}_{\mathrm{ind}} = \mathbb{E}\big[\|\hat{\theta}_{\mathrm{ind}} - \theta^\star\|_2^2\big] = \|b_{\mathrm{ind}}\|^2 + \mathrm{tr}(\Sigma_{\mathrm{ind}}), \qquad \mathrm{MSE}_{\mathrm{dir}} = \mathbb{E}\big[\|\hat{\theta}_{\mathrm{dir}} - \theta^\star\|_2^2\big] = \|b_{\mathrm{dir}}\|^2 + \mathrm{tr}(\Sigma_{\mathrm{dir}})$$

H-learner. We begin by analyzing the MSE along the scalar path $W = \omega I$, under which $\hat{\theta}_H$ forms a convex combination of the indirect and direct OLS estimators: $\hat{\theta}_H = (1-\omega)\,\hat{\theta}_{\mathrm{ind}} + \omega\,\hat{\theta}_{\mathrm{dir}}$.

We then compute the bias of the H-learner.

$$\begin{aligned} b_H &= \mathbb{E}[\hat{\theta}_H - \theta^\star] \\ &= \mathbb{E}\left[(1-\omega)\,\hat{\theta}_{\mathrm{ind}} + \omega\,\hat{\theta}_{\mathrm{dir}} - \theta^\star\right] \\ &= (1-\omega)\,b_{\mathrm{ind}} + \omega\,b_{\mathrm{dir}}. \end{aligned}$$

Variance of the H-learner:

$$\begin{aligned} \mathrm{Var}(\hat{\theta}_H) &= \mathrm{Var}\left((1-\omega)\,\hat{\theta}_{\mathrm{ind}} + \omega\,\hat{\theta}_{\mathrm{dir}}\right) \\ &= (1-\omega)^2\,\mathrm{Var}(\hat{\theta}_{\mathrm{ind}}) + \omega^2\,\mathrm{Var}(\hat{\theta}_{\mathrm{dir}}) + 2\omega(1-\omega)\,\mathrm{Cov}(\hat{\theta}_{\mathrm{ind}}, \hat{\theta}_{\mathrm{dir}}). \end{aligned}$$

Therefore, the MSE of the H-learner is:

$$\begin{aligned} \mathrm{MSE}_H &= \mathbb{E}\Big[\big\|\hat{\theta}_H - \theta^\star\big\|_2^2\Big] \\ &= \|b_H\|_2^2 + \mathrm{tr}\big(\mathrm{Var}(\hat{\theta}_H)\big) \\ &= \big\|(1-\omega)\,b_{\mathrm{ind}} + \omega\,b_{\mathrm{dir}}\big\|_2^2 + \mathrm{tr}\Big((1-\omega)^2\Sigma_{\mathrm{ind}} + \omega^2\Sigma_{\mathrm{dir}} + 2\omega(1-\omega)\Sigma_{\mathrm{ind,dir}}\Big) \\ &= (1-\omega)^2\big(\|b_{\mathrm{ind}}\|_2^2 + \mathrm{tr}(\Sigma_{\mathrm{ind}})\big) + \omega^2\big(\|b_{\mathrm{dir}}\|_2^2 + \mathrm{tr}(\Sigma_{\mathrm{dir}})\big) + 2\omega(1-\omega)\big(b_{\mathrm{ind}}^\top b_{\mathrm{dir}} + \mathrm{tr}(\Sigma_{\mathrm{ind,dir}})\big) \\ &= (1-\omega)^2\,\mathrm{MSE}_{\mathrm{ind}} + \omega^2\,\mathrm{MSE}_{\mathrm{dir}} + 2\omega(1-\omega)\,D, \end{aligned}$$

where $D = b_{\mathrm{ind}}^\top b_{\mathrm{dir}} + \mathrm{tr}\left(\Sigma_{\mathrm{ind,dir}}\right)$, $\Sigma_{\mathrm{ind,dir}} = \mathrm{Cov}(\hat{\theta}_{\mathrm{ind}}, \hat{\theta}_{\mathrm{dir}})$

B.2.3 When Does the H-Learner Strictly Improve MSE?

We now analyze conditions that the optimal convex combination $\hat{\theta}_H = (1-\omega)\,\hat{\theta}_{\mathrm{ind}} + \omega\,\hat{\theta}_{\mathrm{dir}}$ achieve *strictly smaller* MSE than either endpoint ($\omega = 0$ or $\omega = 1$).

From the quadratic form of the H-learner MSE, the optimal weight is

$$\omega^\star = \frac{\mathrm{MSE}_{\mathrm{ind}} - D}{\mathrm{MSE}_{\mathrm{ind}} + \mathrm{MSE}_{\mathrm{dir}} - 2D},$$

The H-learner strictly improves on both endpoints if and only if $\omega^\star \in (0,1)$, which holds when

$$D < \min\{\mathrm{MSE}_{\mathrm{ind}},\, \mathrm{MSE}_{\mathrm{dir}}\} \tag{4}$$

Now we prove Corollary 5.4 and 5.5:

Under the assumption of cross-fitting, $\Sigma_{\mathrm{ind,dir}} = \mathrm{Cov}(\hat{\theta}_{\mathrm{ind}}, \hat{\theta}_{\mathrm{dir}}) = 0, D = b_{\mathrm{ind}}^\top b_{\mathrm{dir}}$.

- **Case 1**: If $b_{\mathrm{ind}}^\top b_{\mathrm{dir}} < 0$, then $D < 0 \leqslant \min\{\mathrm{MSE}_{\mathrm{ind}}, \mathrm{MSE}_{\mathrm{dir}}\}$, so condition (4) holds.
- **Case 2**: If $\|b_{\mathrm{ind}}\|_2 \geqslant \|b_{\mathrm{dir}}\|_2$ and $\mathrm{Var}(\hat{\theta}_{\mathrm{ind}}) = \mathrm{tr}(\Sigma_{\mathrm{dir}}) > b_{\mathrm{dir}}^\top(b_{\mathrm{ind}} - b_{\mathrm{dir}})$. Then

$$D = b_{\mathrm{ind}}^\top b_{\mathrm{dir}} \leqslant \|b_{\mathrm{ind}}\|_2\,\|b_{\mathrm{dir}}\|_2 \leqslant \|b_{\mathrm{ind}}\|_2^2 < \|b_{\mathrm{ind}}\|_2^2 + \mathrm{tr}(\Sigma_{\mathrm{ind}}) = \mathrm{MSE}_{\mathrm{ind}}$$

$$D = b_{\mathrm{ind}}^\top b_{\mathrm{dir}} < b_{\mathrm{dir}}^\top b_{\mathrm{dir}} + \mathrm{tr}(\Sigma_{\mathrm{dir}}) = \mathrm{MSE}_{\mathrm{dir}}$$

  Therefore $D < \min\{\mathrm{MSE}_{\mathrm{ind}}, \mathrm{MSE}_{\mathrm{dir}}\}$, so condition (4) holds.

Since $\min_W \mathrm{MSE}_H(W) \leqslant \min_{\omega\in[0,1]} \mathrm{MSE}_H(\omega I)$, the conditions derived above along this scalar path provide a sufficient guarantee: whenever the scalar path admits an interior minimizer, the full matrix optimization achieves no higher risk and therefore strictly improves upon both endpoints.

C Additional Semi-Synthetic Experiment Results

In the main paper, we present results from the semi-synthetic experiment evaluating the H-learner (X). Here, we provide additional results for the H-learner (DR), comparing it against its counterparts: the DR-learner (direct learner) and TARNet (indirect learner). We follow the same experimental setup detailed in Section 6.2 and fix the proportion of shared features at 90% in Setups B and C to better observe the performance trade-offs. The results are summarized in Figure 7.

Results. Setup A demonstrates the performance trade-off through the *relative complexity of the POs and the CATE*. As the proportion of shared features increases, the CATE becomes simpler. Correspondingly, we observe that the DR-learner only outperforms TARNet when the proportion of shared features is high but underperform when the shared feature proportion is low. Setup B demonstrates the performance trade-off driven by the *data imbalance between treated and control populations*. As the proportion of treated units decreases, the imbalance in sample sizes between the treatment and control groups becomes more severe. This affects the DR-learner, as its pseudo-outcomes depend on inverse propensity scores that exhibit high variance. Empirically, DR-learner performs better under balanced treatment assignment but underperforms when only 10% of units receive treatment. Setup C demonstrates the trade-off associated with the *strength of confounding*. As confounding increases and feature distributions diverge more significantly, the DR-learner performs worse due to exacerbated variance in its inverse propensity pseudo-outcomes. In contrast, the H-learner is more robust across these settings and performs better.

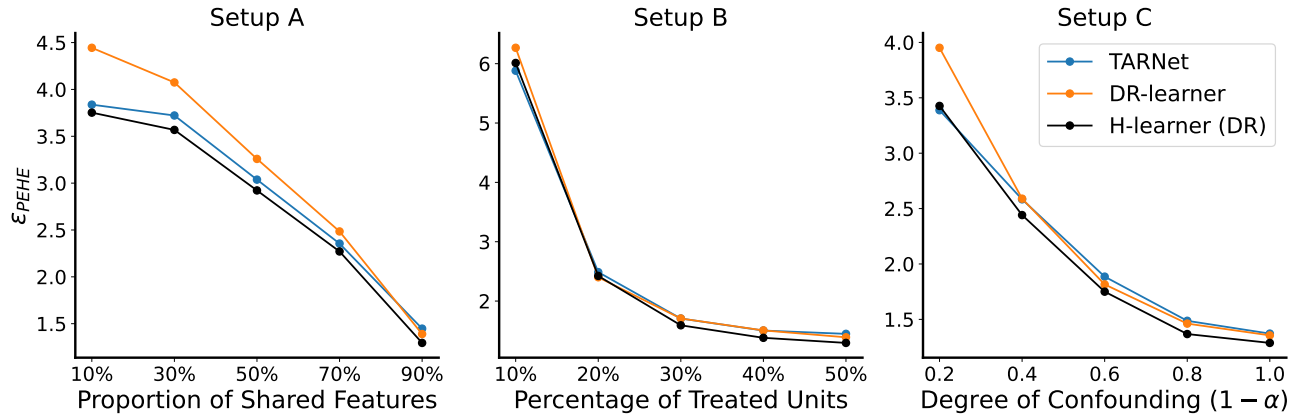

Figure 7: PEHE results for three semi-synthetic scenarios. The H-learner is more robust across these settings and outperforms both learners in most cases.

Effect of regularization parameter λ. Lastly, we add additional results by expanding Figure 5 to include all three experimental setups. Figure 8 presents the H-learner's performance across different values of the regularization strength λ for each setup. Note that $\lambda = 0$ corresponds to the indirect learner, and $\lambda = 1$ to the direct learner. Depending on the levels of feature sharing, treatment proportion, and confounding, either $\lambda = 0$ or $\lambda = 1$ may perform better. However, the optimal value of λ consistently falls between the two extremes, demonstrating the strength of the H-learner in leveraging both types of regularization. Across different setups, the H-learner consistently outperforms both the direct and indirect learners.

D Additional Results on ACIC 2016

The ACIC 2016 benchmark datasets were introduced as part of the Atlantic Causal Inference Competition (Dorie et al., 2019) and offer a comprehensive testbed for evaluating causal inference methods. The benchmark includes 77 distinct DGPs that vary in the complexity of the response surface, the degree of confounding, covariate overlap, and treatment effect heterogeneity. While the main paper reports results averaged over all DGPs, here we provide aggregated results for specific settings.

Results. We present the results in Tables 4, 5, 6, and 7. The H-learner consistently achieves state-of-the-art performance across various response models, levels of heterogeneity, confounding, and overlap. The best performance is obtained when constructing the X-learner pseudo-outcome in the first stage. We also observe that

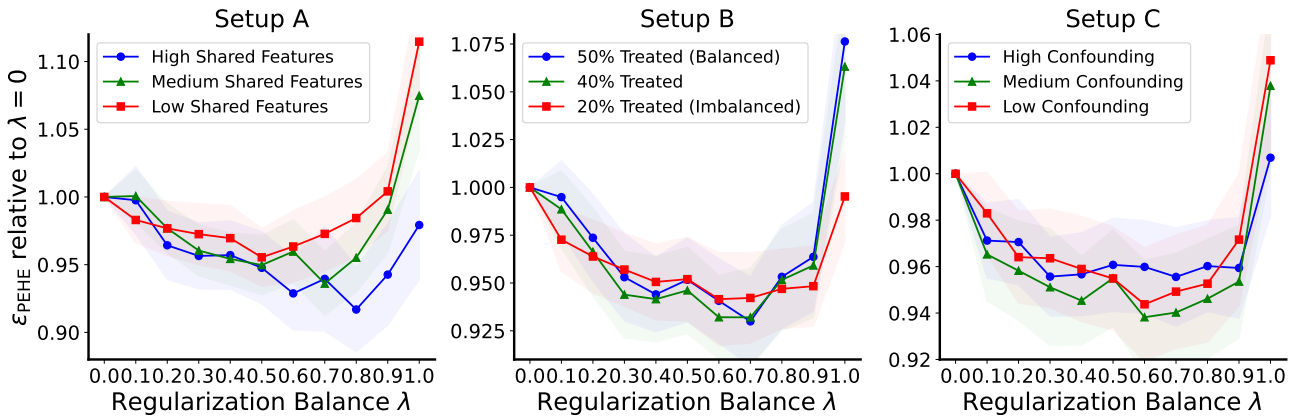

Figure 8: Performance of the H-learner across different λ values for all three setups. Note that $\lambda = 0$ corresponds to the indirect learner (TARNet), and $\lambda = 1$ to the direct learner (X-learner). The H-learner consistently outperforms both learners across all three setups.

using the DR-learner pseudo-outcome yields competitive performance under settings with low confounding or low heterogeneity. This aligns with expectations, as the DR pseudo-outcome estimator tends to suffer less from high variance in such scenarios.

Table 4: Performance on the ACIC 2016 benchmark dataset across different **response models**.

Response Model	**Linear**		**Exponential**		**Step**	
	In-sample $\sqrt{\varepsilon_{\text{PEHE}}}$	Out-of-sample $\sqrt{\varepsilon_{\text{PEHE}}}$	In-sample $\sqrt{\varepsilon_{\text{PEHE}}}$	Out-of-sample $\sqrt{\varepsilon_{\text{PEHE}}}$	In-sample $\sqrt{\varepsilon_{\text{PEHE}}}$	Out-of-sample $\sqrt{\varepsilon_{\text{PEHE}}}$
T-learner	1.51 ± .09	1.56 ± .11	2.05 ± .05	2.19 ± .05	2.23 ± .06	2.30 ± .06
TARNet	1.01 ± .06	1.03 ± .07	1.73 ± .05	1.86 ± .05	1.70 ± .05	1.79 ± .06
IPW-learner	1.83 ± .17	1.84 ± .18	3.13 ± .10	3.13 ± .10	2.31 ± .09	2.30 ± .08
DR-learner	1.00 ± .07	1.03 ± .09	1.75 ± .05	1.90 ± .06	1.60 ± .06	1.69 ± .06
X-learner	**0.96 ± .07**	**0.96 ± .07**	1.69 ± .05	1.84 ± .05	1.58 ± .05	1.67 ± .06
TARNet-WR	1.32 ± .09	1.35 ± .11	2.02 ± .06	2.12 ± .06	1.82 ± .06	1.89 ± .06
OffsetNet	1.37 ± .12	1.38 ± .12	2.30 ± .07	2.37 ± .07	1.99 ± .07	2.03 ± .08
FlexTENet	1.19 ± .10	1.23 ± .12	1.94 ± .06	2.04 ± .06	1.76 ± .06	1.83 ± .07
H-learner (DR)	**0.96 ± .07**	1.00 ± .09	1.67 ± .05	1.79 ± .05	1.57 ± .06	1.66 ± .06
H-learner (X)	0.98 ± .07	1.00 ± .08	**1.65 ± .04**	**1.78 ± .05**	**1.56 ± .06**	**1.65 ± .06**

Table 5: Performance on the ACIC 2016 benchmark dataset across different **heterogeneity levels**.

Heterogeneity Level	**Low**		**High**	
	In-sample $\sqrt{\varepsilon_{\text{PEHE}}}$	Out-of-sample $\sqrt{\varepsilon_{\text{PEHE}}}$	In-sample $\sqrt{\varepsilon_{\text{PEHE}}}$	Out-of-sample $\sqrt{\varepsilon_{\text{PEHE}}}$
T-learner	1.99 ± .05	2.06 ± .06	2.16 ± .05	2.30 ± .06
TARNet	1.56 ± .05	1.65 ± .05	1.77 ± .05	1.90 ± .05
IPW-learner	2.47 ± .11	2.45 ± .10	2.95 ± .08	2.97 ± .09
DR-learner	1.51 ± .05	1.61 ± .06	1.77 ± .05	1.90 ± .06
X-learner	1.48 ± .05	1.57 ± .05	1.71 ± .05	1.85 ± .05
TARNet-WR	1.75 ± .06	1.82 ± .06	2.04 ± .06	2.13 ± .06
OffsetNet	1.91 ± .07	1.96 ± .07	2.31 ± .06	2.37 ± .07
FlexTENet	1.65 ± .06	1.72 ± .06	1.97 ± .06	2.08 ± .06
H-learner (DR)	**1.45 ± .05**	**1.53 ± .05**	1.71 ± .05	1.83 ± .05
H-learner (X)	**1.45 ± .05**	**1.53 ± .05**	**1.69 ± .05**	**1.82 ± .05**

Table 6: Performance on the ACIC 2016 benchmark dataset across different **confounding levels**.

Confounding Level	**Low**		**High**	
	In-sample $\sqrt{\varepsilon_{\text{PEHE}}}$	Out-of-sample $\sqrt{\varepsilon_{\text{PEHE}}}$	In-sample $\sqrt{\varepsilon_{\text{PEHE}}}$	Out-of-sample $\sqrt{\varepsilon_{\text{PEHE}}}$
T-learner	$2.09 \pm .05$	$2.19 \pm .06$	$2.08 \pm .05$	$2.20 \pm .06$
TARNet	$1.64 \pm .05$	$1.75 \pm .06$	$1.69 \pm .05$	$1.79 \pm .05$
IPW-learner	$2.62 \pm .10$	$2.59 \pm .10$	$2.75 \pm .10$	$2.76 \pm .10$
DR-learner	$1.61 \pm .05$	$1.72 \pm .06$	$1.66 \pm .05$	$1.77 \pm .06$
X-learner	$1.57 \pm .05$	$1.68 \pm .06$	$1.61 \pm .05$	$1.72 \pm .06$
TARNet-WR	$1.88 \pm .06$	$1.96 \pm .06$	$1.91 \pm .06$	$1.99 \pm .06$
OffsetNet	$2.05 \pm .07$	$2.11 \pm .07$	$2.15 \pm .07$	$2.20 \pm .08$
FlexTENet	$1.79 \pm .06$	$1.88 \pm .07$	$1.83 \pm .06$	$1.92 \pm .06$
H-learner (DR)	$\mathbf{1.55 \pm .05}$	$\mathbf{1.65 \pm .06}$	$1.60 \pm .05$	$1.70 \pm .06$
H-learner (X)	$\mathbf{1.55 \pm .05}$	$1.66 \pm .06$	$\mathbf{1.58 \pm .05}$	$\mathbf{1.68 \pm .05}$

Table 7: Performance on the ACIC 2016 benchmark dataset across different **overlap levels**.

Overlap Level	**Full**		**Penalize**	
	In-sample $\sqrt{\varepsilon_{\text{PEHE}}}$	Out-of-sample $\sqrt{\varepsilon_{\text{PEHE}}}$	In-sample $\sqrt{\varepsilon_{\text{PEHE}}}$	Out-of-sample $\sqrt{\varepsilon_{\text{PEHE}}}$
T-learner	$2.09 \pm .05$	$2.21 \pm .06$	$2.07 \pm .05$	$2.17 \pm .05$
TARNet	$1.66 \pm .05$	$1.78 \pm .06$	$1.66 \pm .05$	$1.76 \pm .05$
IPW-learner	$2.73 \pm .10$	$2.76 \pm .10$	$2.65 \pm .09$	$2.63 \pm .09$
DR-learner	$1.63 \pm .05$	$1.76 \pm .06$	$1.64 \pm .05$	$1.74 \pm .06$
X-learner	$1.58 \pm .05$	$1.71 \pm .06$	$1.59 \pm .05$	$1.69 \pm .05$
TARNet-WR	$1.92 \pm .06$	$2.01 \pm .07$	$1.86 \pm .05$	$1.94 \pm .06$
OffsetNet	$2.17 \pm .07$	$2.23 \pm .08$	$2.05 \pm .06$	$2.09 \pm .07$
FlexTENet	$1.85 \pm .06$	$1.94 \pm .07$	$1.78 \pm .05$	$1.86 \pm .06$
H-learner (DR)	$1.57 \pm .05$	$1.68 \pm .06$	$1.58 \pm .05$	$\mathbf{1.67 \pm .05}$
H-learner (X)	$\mathbf{1.55 \pm .05}$	$\mathbf{1.66 \pm .06}$	$\mathbf{1.57 \pm .05}$	$\mathbf{1.67 \pm .06}$