Distilling Heterogeneity: From Explanations of Heterogeneous Treatment Effect Models to Interpretable Policies

Han Wu*  
HANWU71@STANFORD.EDU

Sarah Tan*  
SARAHTAN@FB.COM

Weiwei Li  
WEIWEILI90@FB.COM

Mia Garrard  
MGARRARD@FB.COM

Adam Obeng  
ADAMOBENG@FB.COM

Drew Dimmery  
DREW.DIMMERY@UNIVIE.AC.AT

Shaun Singh  
SHAUNDSINGH@GMAIL.COM

Hanson Wang  
HANSONW@FB.COM

Daniel Jiang  
DRJIANG@FB.COM

Eytan Bakshy  
EBAKSHY@FB.COM

Abstract

Internet companies are increasingly using machine learning models to create personalized policies which assign, for each individual, the best predicted treatment for that individual. They are frequently derived from black-box heterogeneous treatment effect (HTE) models that predict individual-level treatment effects. In this paper, we focus on (1) learning explanations for HTE models; (2) learning interpretable policies that prescribe treatment assignments. We also propose guidance trees, an approach to ensemble multiple interpretable policies without the loss of interpretability. These rule-based interpretable policies are easy to deploy and avoid the need to maintain a HTE model in a production environment.

Keywords: Interpretability, policy learning, heterogeneous treatment effect models

* Equal contribution
1. Introduction

Unlike conventional A/B testing where individuals are randomly assigned to a treatment condition, internet companies are increasingly using heterogeneous treatment effects models (HTE) and personalized policies to decide, for each individual, the best predicted treatment (Garcin et al., 2013; Toğuç and Kuzu, 2020). However, to flexibly model heterogeneity in treatment and outcome surfaces, state-of-the-art HTE models used for personalization (Wager and Athey, 2018; Shi et al., 2019; Kennedy, 2020) frequently leverage black-box, uninterpretable base learners such as gradient boosted trees and neural networks. To generate a HTE prediction, some of them combine outputs from multiple models via stacking, weighting, etc. (Künzel et al., 2019; Nie and Wager, 2021; Montoya et al., 2021). Moreover, their treatment effect predictions are sometimes used as inputs to additional black-box policy learning models (Imai and Strauss, 2011; McFowland et al., 2020).

There are several challenges with these approaches. Firstly, when there are multiple (not just binary) outcomes, the creation of a policy is no longer as trivial as assigning the treatment group that maximizes a single treatment effect. Secondly, as dataset size and the number of features used for HTE modeling or policy learning increases, the cost of maintaining such models in deployment increases (Sculley et al., 2015a). Finally, the black-box nature of HTE models and resulting policies makes them difficult to interpret, which can deter their uptake for critical applications.

This paper describes methods for: (1) learning explanations of HTE models, to better understand such models; (2) generating interpretable policies that can perform on par with or even better than black-box policies. Our explanations and interpretable policies are constructed from if-else rules, using individual features as inputs (Figure 1). We demonstrate a variety of different ways to learn interpretable policies, from methods that act on HTE model predictions to methods that work on conditional average treatment effects. We also propose ways to ensemble multiple interpretable policies while still remaining interpretable.

Crucially, all of our proposed methods work in the general setting of more than two treatment groups and multiple outcomes, which is needed in production environments at internet companies. For example, a new website feature may increase user engagement at the expense of revenue. We do so by leveraging techniques such as model distillation (Bucilua et al., 2006; Hinton et al., 2014) with multitask (Caruana, 1997) and multiclass student models, and linear scalarization of multiple outcomes from the multi-objective optimization literature (Gunantara (2018)).

![Figure 1: An interpretable policy using two features – age and gender – to prescribe which of three treatments to assign to an individual.](image-url)
2. Related Work

We do not attempt to comprehensively survey the state of HTE estimation since our contribution is agnostic to the choice of HTE model. Most modern HTE estimation methods are compatible with our proposed methods.

Subgroup finding. Our work on interpreting HTE models by surfacing segments with heterogeneity produces a representation analogous to subgroup finding methods. These methods are divided into: (1) statistical tests to confirm or deny a set of pre-defined hypotheses about subgroups (Assmann et al., 2000; Song and Chi, 2007); (2) methods that aim to discover subgroups directly from data (Imai and Ratkovic, 2013; Chen et al., 2015; Wang and Rudin, 2021; Nagpal et al., 2020); (3) methods that act on inferred individual-level treatment effects to discover subgroups Foster et al. (2011); Lee et al. (2020). However, the comparison of subgroup finding algorithms is still an open question. Loh et al. empirically compared 13 different subgroup-finding algorithms (Loh et al., 2019), finding that no one algorithm satisfied all the desired properties of such algorithms.

Policy learning. The problem of policy learning from observational/experimental data has received significant attention due to its wide applicability (Beygelzimer and Langford, 2009; Qian and Murphy, 2011; Zhao et al., 2012; Swaminathan and Joachims, 2015; Kitagawa and Tetenov, 2018; Zhou et al., 2018; Kallus and Zhou, 2018), but their interpretability and scalability has received less attention. Several existing works (Amram et al., 2020; Athey and Wager, 2021; Jo et al., 2021) do construct interpretable tree policies based on counterfactual estimation. However, they focus on constructing exactly optimal trees, which is prohibitively slow in a large scale and unsuitable for a production environment.

Interpretability via distillation and imitation learning. Black-box classification and regression models have been distilled into interpretable models such as trees, rule lists, etc. (Craven and Shavlik, 1995; Frost and Hinton, 2017; Tan et al., 2018b,a). Craven and Shavlik (1995) distilled a neural net to a decision tree and evaluated the explanation in terms of fidelity, accuracy, and complexity. Namkoong et al. (2020) distilled a Thompson Sampling policy to policies represented using MLPs. Interpreting reinforcement learning methods has also garnered interest; see Alharin et al. (2020) for a survey. These methods can be roughly divided into those that leverage the interpretable representation provided by trees to approximate states and actions (Bastani et al., 2018; Vasic et al., 2019), saliency and attention methods (Wang et al., 2016; Greydanus et al., 2018; Qureshi et al., 2017), and distillation (Czarnecki et al., 2019). We also use distillation techniques, but in a non-sequential setting.

3. Methods

3.1. Problem Setting and Notation

Suppose we have a dataset $\mathcal{D}$ of $N$ points and $P$ features with $K$ treatment groups, 1 control group, and $J$ outcomes. Let $W_i \in \{0, 1, \ldots, K\}$ denote the treatment group indicator for the $i$-th individual ($W_i = 0$ means the individual is in the control group). Let $\mathcal{G}_k$ denote the set of individuals in the $k$-th treatment group, $k = 0, 1, \ldots, K$. Following the potential outcome framework (Neyman, 1923; Rubin, 1974), we assume for each individual $i$ and outcome $j$ there are $K + 1$ potential outcomes $Y_{ij}^{(0)}, \ldots, Y_{ij}^{(K)}$. For each individual $i$, we observe features $X_i \in \mathbb{R}^P$ and outcomes $(Y_{i1}, \ldots, Y_{iJ})$ where $Y_{ij} = Y_{ij}^{(W_i)}$, the observed potential outcome (out of $K + 1$ potential outcomes). We suppose
we have access to predicted outcomes \( (Y_{i1}^{(0)}, ..., Y_{i1}^{(K)}, ..., Y_{ij}^{(0)}, ..., Y_{ij}^{(K)}) \), where \( \hat{Y}_{ij}^{(W)} \) is the predicted \( j \)-th outcome for individual \( i \) assuming treatment group \( W \). These could be obtained directly from the HTE model or together with estimates of the baseline surface.

Throughout this work, we combine multiple outcomes into a single outcome via linear scalarizations, parameterized by a set of weights \( (c_1, ..., c_J) \). Thus, for each individual \( i \), the multiple observed outcomes becomes a single outcome:

\[
Y_i = Y_i^{(W_i)} = c_1 Y_{i1}^{(W_i)} + \cdots + c_J Y_{ij}^{(W_i)} = c_1 Y_{i1} + \cdots + c_J Y_{ij}
\]

We also combine predicted outcomes in the same way: \( \hat{Y}_i^{(W)} = c_1 \hat{Y}_{i1}^{(W)} + \cdots + c_J \hat{Y}_{ij}^{(W)} \). Our proposed methods take \( (c_1, ..., c_J) \) as inputs and assume a value 1 (equal weighting on outcomes) if not provided. In practice, it is common to either tune the weights \( (c_1, ..., c_M) \) (Letham and Bakshy, 2019) or handpick weights to manage particular preferences or tradeoffs.

### 3.3. Explaining Heterogeneous Treatment Effect Models

**Multi-task decision trees** (MTDT) extend single-task decision trees by combining the prediction loss on each outcome, across multiple outcomes, into a single scalar loss function. This simple representation is effective and suitable for locating segments where individuals present heterogeneity across multiple outcomes. In this tree, each task (label) is the predicted treatment effect for outcome \( j \), and each node is a segment identified to have elevated or depressed treatment effects. We learn these trees using distillation techniques. The resulting method is called Distilling HTE model predictions (Distill-HTE). Figure 2 provides a concrete example.

**Distillation.** To learn an MTDT model in a HTE model-agnostic fashion, we leverage model distillation techniques, taking the HTE model as the teacher and the MTDT model as a student. After obtaining predicted outcomes \( \hat{Y}_{ij}^{(W)} \) from trained HTE models, where \( \hat{Y}_{ij}^{(W)} \) is the predicted \( j \)-th outcome for individual \( i \) assuming treatment group \( W \), we construct pairwise predicted treatment effects \( \hat{T}_{ij} \), i.e. individual \( i \)'s predicted treatment effect for outcome \( j \) for a particular treatment contrast (analogous definitions will hold for all treatment contrasts). Let \( \hat{F} \) be the prediction function of an MTDT model using the following distillation training objective that minimizes the mean square error between the HTE model predicted treatment effects and \( \hat{F} \):

\[
L = \frac{1}{N} \sum_{i=1}^{N} \left\| \sum_{j=1}^{J} c_j \left[ \hat{T}_{ij} - \hat{F}_j(X_{ij}) \right] \right\|^2_2
\]

\( \hat{F}_j \) is the MTDT model’s prediction for just outcome \( j \), accompanied by \( c_j \), a weight that encodes how much outcome \( j \) contributes to the overall loss, as reviewed in Section 3.2. When \( c_j = 1 \), as we will assume by default, each outcome contributes equally to the overall distillation loss.
Figure 2: Distill-HTE: Multitask treatment effect decision tree learned by distilling a black-box HTE model. After the tree is learned, we “unroll” the tree such that each terminal node forms a segment visualized in a barplot.

**Counterfactual reasoning.** To improve the robustness of the subgroups, we introduce three elements based on counterfactual reasoning ideas: (1) Terminal nodes without sufficient overlap (i.e., enough treatment AND control points) are post-pruned from the tree, and the prediction regenerated; (2) Confidence intervals are provided for treatment effects within each node; (3) Honesty (Athey and Imbens, 2016): splits and predictions are determined on different data splits.

**From treatment effects to policies.** With $K$ treatment groups, 1 control group, and $M$ outcomes, the proposed method learns $K$ MTDT models, where each model predicts treatment effects of $M$ outcomes for each treatment group compared to control. One MTDT model can then be used to generate a policy that indicates, for each segment, which of the pair of treatment groups should be assigned to maximize treatment effects. However, generating a single policy still requires combining multiple MTDT, and it is not immediately obvious the best way to do so. This motivates our proposal of directly learning a single policy that applies to multiple treatment groups and outcomes.

### 3.4. Interpretable Policy Learning

To generate a single interpretable policy $\Pi$ that applies to multiple treatment groups and outcomes, we now describe different approaches, all using if-else rules representations.

#### 3.4.1. Approaches Based on HTE Model Predictions

An already trained HTE model from which predicted outcomes can be obtained is a prerequisite for the approaches described in this subsection. We describe two ways of leveraging these predictions: GreedyTreeSearch-HTE and Distill-Policy.
Greedy tree search from HTE model predictions (GreedyTreeSearch-HTE). In this approach, we directly solve the following optimization problem in the space of trees with pre-determined maximum tree depth, assuming without loss of generality that higher outcomes are better:

$$\Pi^* = \arg\max_{\Pi} \frac{1}{N} \sum_{i=1}^{N} \hat{Y}_i(\Pi(X_i)).$$  \hspace{1cm} (2)

This can be seen as a cost-sensitive classification problem, where the cost of assigning a point to a group is the negative value of the predicted potential outcome (Elkan, 2001). To achieve a scalable implementation, we solve the optimization problem greedily instead of resorting to exact tree search over the whole space of possible trees. Define $$\hat{M}_j(S) = \sum_{u_i \in S} \hat{Y}_i(j)$$ and $$\hat{M}(S) = \max_j \hat{M}_j(S)$$.

The detailed implementation can be found in Algorithm 1.

Algorithm 1 Greedy Tree Search from HTE Model Predictions (GreedyTreeSearch-HTE)

Input : Maximal tree depth $$d$$, dataset $$D$$, predictions from HTE model $$\{(\hat{Y}_i^{(0)}, ..., \hat{Y}_i^{(K)})\}_{i=1}^{N}$$
Output: A list of segments: segments.

1. Set depth := 0 and segments := $$\{D\}$$.
2. Add segments to segments using greedy search:
   
   while depth $$\leq m$$ do
     
     Set nodes := $$\{\}$$
     
     for $$S$$ in segments do
       
       Split $$S$$ into $$S_l^*$$ and $$S_r^*$$ such that:
       
       $$(S_l^*, S_r^*) = \arg\max_{\hat{M}(S_l) + \hat{M}(S_r) > \hat{M}(S)} (\hat{M}(S_l) + \hat{M}(S_r)).$$
       
       If such $$S_l^*$$ and $$S_r^*$$ exist, we add them to nodes; otherwise we add $$S$$ to nodes.
     
     end
     
     Let segments := nodes and depth := depth + 1.
   
end

This approach is similar to Zhou et al. (2018) and Amram et al. (2020) which respectively solve an exact tree search problem over predicted outcomes and find the optimal tree using coordinate ascent. Due to scalability concerns and the need for faster test-time inference, we take a greedy approach. On a sample of 100k data points with 12 features and three treatment groups, the Zhou et al. (2018) method took 2.5 hours to run. With a quadratic runtime in the number of data points, and the amount of data at large internet companies sometimes exceeding 100 million data points, it is impossible to run it at a scale suitable for internet companies.

Distilling policy derived from HTE model predictions (Distill-Policy). In this approach, we start from the naive policy $$\Pi$$ implied by the outcome predictions $$\hat{Y}_i^{(0)}, ..., \hat{Y}_i^{(K)}$$, i.e. $$\Pi_{HTE}(X_i) = \arg\max_W \hat{Y}_i^{(W)}$$. Then, we train a decision tree classifier on the dataset $$(X_i, \Pi_{HTE}(X_i))_{i=1}^{N}$$, and output the segments from the decision tree classifier. The resulting policy assigns all individuals in the same segment to the majority treatment group of individuals in that segment.

3.4.2. Approaches Not Based on HTE Model Predictions

The non-HTE-model based methods proposed in this section are useful when we are not able to train an accurate HTE model. Here we aim to generate an interpretable policy without training a
HTE model, by leveraging average treatment effects. Concretely, we define the segment average treatment effect for segment $S$ and treatment $j$ as:

$$A_j(S) = \frac{1}{|G_j \cap S|} \sum_{u_i \in G_j \cap S} Y_i - \frac{1}{|G_0 \cap S|} \sum_{u_i \in G_0 \cap S} Y_i,$$

where $G_j$ denote the set of individuals in the $j$-th treatment group. We then define a splitting metric $A(S) = \max_{j=0,1,...,K} A_j(S)$, which considers only the most responsive treatment in that segment. We build a tree and consider binary splits that improve this splitting metric.

Algorithm 3 and Algorithm 4 describe two different implementations for this. In the greedy implementation, a split is only considered if both the left and right child segment have an improvement over the parent segment. In the iterative implementation, we run several iterations of splitting. For each iteration we keep the best segment, excluding it in the next run. A split is considered as long as one of the child segments improves the outcome compared to the parent node. One pitfall with this iterative implementation is that segments are not necessarily disjoint in feature space, so one individual could appear in several segments. We resolve this by always assigning the individual to the first found segment.

For both methods, if no segments are found because no eligible split exists, the policy defaults to assigning all individuals to the treatment group with highest average treatment effect. The resulting policy is:

$$\Pi^* = \arg\max_j A_j(S) \text{ if } S \in \text{segments and individual } i \in S \text{ and if } i \not\in S \text{ for } \forall S \in \text{segments},$$

We introduce two ways of doing so. For ease of notation, we assume the trained policies $\Pi_1, \Pi_2, ..., \Pi_Q$ were obtained from another split of the dataset and we can safely use $D$ as the validation set on which we learn the ensemble policy.

**Ensemble based on Uniform Exploration** (GUIDE-UniformExplore): This method is inspired by the explore-exploit paradigm in the contextual bandits literature (see Slivkins (2021) for a review). To perform this offline, we use HTE outcome predictions when the observed outcome is not revealed in the dataset. Algorithm 2 provides the implementation. The ensemble policy $\hat{\pi}$ is generated using Algorithm 1 but with the HTE predictions $(\hat{Y}_i^{(0)}, ..., \hat{Y}_i^{(K)})$ replaced by $(O_i^{(0)}, ..., O_i^{(K)})$.

**Ensemble using Offline Evaluation** (GUIDE-OPE): We aim to find one feature split such that the left and right children uses a different candidate policy. We do so using offline policy evaluation (OPE; see Algorithm 5). Suppose we split dataset $D$ into $D_l$ and $D_r$, let $\hat{\Pi}$ be the ensemble policy that applies $\Pi_k$ to $D_l$ and $\Pi_q$ to $D_r$. To create $\hat{\Pi}$, we use exhaustive search to find the optimal feature
Algorithm 2: Ensemble based on Uniform Exploration (GUIDE-UniformExplore)

**Input**: Maximal tree depth $d$, dataset $D$, predictions $\{(\hat{Y}_i^{(0)}, \ldots, \hat{Y}_i^{(K)})\}_{i=1}^N$, $Q$ trained interpretable policies $\Pi_1, \ldots, \Pi_Q$.

**Output**: A list of segments: $\text{segments}$.

1. Generate a dataset with randomly selected policies on individuals

   for $i = 1$ to $N$ do
     Randomly select a policy from $\Pi_1, \ldots, \Pi_Q$. Let $A_i$ be the index of this policy, $A_i \in 1, \ldots, Q$.
     Apply this policy to individual $i$, assume the policy assigns treatment group $k \in \{0, 1, \ldots, K\}$.
     Initialize $O_i = [0, \ldots, 0] \in \mathbb{R}^{K+1}$. Let $W_i$ be the treatment received by individual $i$ in the dataset.
     If $k == W_i$, set $O_i^{(k)} := Y_i$. Otherwise, set $O_i^{(k)} := \hat{Y}_i^{(k)}$.
   end

   We now have a new dataset of the form $(X_i, A_i \in \{1, 2, \ldots, Q\}, O_i)$.

2. Apply Algorithm 1 to the new dataset, using $(O_i^{(0)}, \ldots, O_i^{(K)})$ in place of $(\hat{Y}_i^{(0)}, \ldots, \hat{Y}_i^{(K)})$.

split and candidate policies $(D^*_1, D^*_r, \Pi^*_l, \Pi^*_q)$ that solves $\max_{D_1, D_r} \max_{1 \leq k \neq q \leq Q} \text{OPE}(D, \tilde{\Pi})$. We assign to individual $i$ policy $\Pi^*_k$ if individual $i \in D^*_1$ and $\Pi^*_q$ otherwise.

Both methods return trees that we call guidance trees; see Figure 3 for an example. While there exists many other ways to ensemble policies, such as SuperLearner (Montoya et al., 2021), we do not consider these methods as they result in non-interpretable policies.

4. Results

4.1. Comparing Explanations of HTE Models

We compare the Distill-HTE method proposed in Section 3.3 against several other subgroup finding methods that take the predictions of HTE models as input: (1) Virtual Twins (VT) (Foster et al., 2011); (2) R2P (Lee et al., 2020). We deliberately restrict to such post-hoc methods, due to shared motivation of explaining already-trained HTE models. We also compare to a black-box model: a T-Learner that does not find subgroups but rather provides one prediction per individual.

**Setup**: The first dataset, Synthetic COVID, was proposed by Lee et al. (2020) and uses patient features as in an initial clinical trial for the Remdesivir drug, but generates synthetic outcomes where the drug reduces the time to improvement for patients with a shorter period of time between symptom onset to starting the trial. The second dataset, Synthetic A, was proposed in Athey and
Table 1: Test-set performance of subgroup finding methods on synthetic datasets. For PEHE and variance, lower is better. Best method for each column in bold.

For PEHE and variance, lower is better. Best method for each column in bold.

### Synthetic COVID

| Data       | Method          | PEHE    | Within-subgroup var | Between-subgroup var |
|------------|-----------------|---------|---------------------|----------------------|
|            | Virtual Twins RF| 3.054 ± 0.960 | 2.009 ± 0.731 | 0.518 ± 0.715 |
|            | Virtual Twins GBDT | 1.903 ± 0.660 | 0.164 ± 0.127 | 2.362 ± 0.216 |
|            | Distill-HTE     | 0.648 ± 0.080 | 0.337 ± 0.108 | 1.986 ± 0.282 |
|            | R2P             | 3.265 ± 0.671 | 2.355 ± 0.120 | 0.180 ± 0.186 |
|            | T-Learner GBDT  | 1.902 ± 0.654 | –                | –                   |
|            | T-Learner DT    | 4.777 ± 1.081 | –                | –                   |

### Synthetic A

| Data       | Method          | PEHE    | Within-subgroup var | Between-subgroup var |
|------------|-----------------|---------|---------------------|----------------------|
|            | Virtual Twins RF| 0.433 ± 0.050 | 0.051 ± 0.008 | 0.174 ± 0.023 |
|            | Virtual Twins GBDT | 0.151 ± 0.016 | 0.010 ± 0.001 | 0.255 ± 0.027 |
|            | Distill-HTE     | 0.189 ± 0.033 | 0.037 ± 0.009 | 0.241 ± 0.042 |
|            | R2P             | 0.719 ± 0.200 | 0.211 ± 0.077 | 0.036 ± 0.071 |
|            | T-Learner GBDT  | 0.151 ± 0.016 | –                | –                   |
|            | T-Learner DT    | 0.616 ± 0.064 | –                | –                   |

For each dataset we generate ten train-test splits, on which we compute the mean and standard deviation of estimates. For R2P, we use the implementation of R2P provided by the authors; For VT we used our own implementation.

**Evaluation:** On synthetic data where ground truth treatment effects are available, we report the Precision in Estimation of Heterogeneous Effect (PEHE) (Hill, 2011), defined as

\[
PEHE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (\hat{\mu}_1 - \hat{\mu}_0 - [Y_1 - Y_0])^2}
\]

Unlike bias and RMSE metrics computed relative to ground truth treatment effects, PEHE requires accurate estimation of both counterfactual and factual outcomes (Johansson et al., 2016). We also compute between- and within-subgroup variance.

**Results:** Table 1 presents the results. We make a few observations: (1) **Black-box vs. white-box:** As expected, GBDT T-Learners perform well as they do not have interpretability constraints, unlike all the other methods (VT, Distill-HTE, R2P), all of which modify standard decision trees while still remaining visualize-able as a tree. Yet, Distill-HTE tends to be far more accurate than other decision tree approaches, in terms of PEHE. (2) **Optimization criterion:** R2P, the only method of those presented here, that considers not only homogeneity within subgroups but also heterogeneity between subgroups, has the lowest between-subgroup variance. Other methods that do not try to reduce heterogeneity between subgroups do not fare so well on this metric. On the other hand, they fare better than R2P at minimizing within subgroup variance, because they do not consider the tradeoffs between minimizing within and between subgroup variance, unlike R2P. However, R2P does this at the expense of PEHE. (3) **HTE model class:** The choice of HTE model matters, with Virtual Twins not performing as well when using an RF HTE model compared to GBDT HTE model. Similarly, GBDT T-Learners perform better than DT T-Learners. (4) **The impact of distillation:** While the T-Learner DT model did not perform well, being worst in terms of PEHE on all datasets, Virtual Twins RF, Virtual Twins GBDT and Distill-HTE that train modified decision trees have a marked improvement over T-Learner DT, suggesting that distilling a complex GBDT or RF teacher rather than learning a tree directly is beneficial, which agrees with the existing distillation literature.
| Data                | Method                                           | Test-Set Regret     |
|---------------------|--------------------------------------------------|---------------------|
|                     | Assign all to treatment 0                        | 111.131 ± 10.279    |
| Semi-synthetic IHDP | Assign all to treatment 1                        | 21.537 ± 3.672      |
| Hill (2011)         | BlackBox                                         | 59.079 ± 4.997      |
|                     | GreedyTreeSearch-HTE                            | 45.763 ± 8.981      |
|                     | Distill-Policy                                  | 59.102 ± 11.262     |
|                     | No-HTE (Greedy and Iterative)*                  | 21.537 ± 3.672      |
|                     | GUIDE-UniformExplore (guide depth=1)            | 62.306 ± 11.039     |
| Synthetic A         | GUIDE-UniformExplore (guide depth=2)            | 56.534 ± 3.116      |
| Athey and Imbens (2016) | GUIDE-OPE (guide depth=1)                        | 65.555 ± 9.676      |
|                     | Assign all to treatment 0                        | 62.162 ± 2.232      |
|                     | Assign all to treatment 1                        | 59.021 ± 2.178      |
|                     | BlackBox                                         | 1.259 ± 0.397       |
|                     | GreedyTreeSearch-HTE                            | 0.032 ± 0.029       |
|                     | Distill-Policy                                  | 0.064 ± 0.056       |
|                     | No-HTE (Greedy and Iterative)*                  | 59.021 ± 2.178      |
|                     | GUIDE-UniformExplore (guide depth=1)            | 0.016 ± 0.016       |
|                     | GUIDE-UniformExplore (guide depth=2)            | 0.021 ± 0.014       |
|                     | GUIDE-OPE (guide depth=1)                       | 0.015 ± 0.011       |
| Email Marketing     | Assign all to treatment 0                        | 48840.305 ± 1887.596|
| Hillstrom (2008)    | Assign all to treatment 1                        | 41093.311 ± 3377.773|
|                     | Assign all to treatment 2                        | 33976.605 ± 705.053 |
|                     | BlackBox                                         | 32575.921 ± 767.482|
|                     | GreedyTreeSearch-HTE                            | 33976.605 ± 705.053 |
|                     | Distill-Policy                                  | 33976.605 ± 705.053 |
|                     | No-HTE-Greedy                                   | 34324.125 ± 835.736|
|                     | No-HTE-Iterative                                | 42217.430 ± 1651.130|
|                     | GUIDE (UniformExplore and OPE)                  | 33976.605 ± 705.053 |

* denotes no segments were found, and the resulting policy assigned all individuals to one treatment group.

Table 2: Regret (lower is better) of policy-generation methods on synthetic and semi-synthetic datasets. Methods colored in red do not personalize. Methods colored in black are black-box policies. Methods colored in blue are interpretable policies. Methods colored in green are ensembles that still remain interpretable. Best method’s regret in bold. If the best policy is no personalization (assigning all individuals to one treatment group), the next best policy is also bolded. If the best policy is an ensemble policy, the constituent policies are also bolded.

4.2. Comparing Policy Learning Methods

We compare the methods proposed in Section 3.4 to (1) a black-box policy: training a T-Learner HTE model, then assigning each individual to the treatment group with best predicted treatment effects; (2) a policy which chooses the treatment for each unit uniformly at random.

Setup: Besides the synthetic datasets described in 4.1, we use other publicly-available datasets. The IHDP dataset (Hill, 2011) studied the impact of specialized home visits on infant cognition using mother and child features. A multiple-outcome dataset, Email Marketing (Hillstrom, 2008) has 64k points, and visits, conversions, and money spent outcomes (details on how we constructed potential outcomes in Appendix). We combine multiple outcomes as explained in Section 3.2. The
ensemble policies (GUIDE-UniformExplore, GUIDE-OPE) are based on GreedyTreeSearch-HTE and Distill-Policy – selected because of their individual performance.

**Evaluation:** To evaluate the different policies $\Pi$, we define a notion of regret:

$$R(\Pi) = \sum_{i=1}^{n} \max_{k} Y_{i}^{(k)} - Y_{i}^{(\Pi(X_i))} \quad k = 0, \ldots, K$$  \hspace{1cm} (4)$$

where $\Pi(X_i)$ is the treatment group prescribed by that particular policy for individual $i$ and $Y_{i}^{(k)}$ is individual $i$’s potential outcome under treatment $k$.

**Results:** Table 2 presents the results. We make a few observations: (1) In many cases, at least one interpretable policy (GreedyTreeSearch-HTE, Distill-Policy, No-HTE, ensembled policies) improved over or is not far behind the black-box policy. (2) **Personalization:** In datasets like IHDP where the benefit from personalization was limited, as the majority of points have positive treatment effects, No-HTE-Greedy and No-HTE-Iterative were able to pick up on this best policy and assign all points to one treatment. (3) **The impact of ensembling:** In general, ensembled policies GUIDE-UniformExplore and GUIDE-OPE improved over their constituent policies GreedyTreeSearch-HTE and Distill-Policy. However, ensemble methods with deeper guidance trees are not necessarily better. Hence, the tradeoff between one more level of guidance, and reduced interpretability, should be considered.

### 4.3. Ensembling Interpretable Policies

We now display several of the interpretable policies, and show how they can be ensembled while remaining interpretable. On Synthetic A, a simple dataset with only two features, the following policies were obtained:

**GreedyTreeSearch-HTE:** If feature0 < -0.019, assign control. Otherwise, assign treatment.

**Distill-Policy:** If feature0 $\leq$ -0.02, assign control. Otherwise, assign treatment.

We applied the GUIDE-UniformExplore and GUIDE-OPE approaches to learn guidance trees: trees that ensemble these two policies while still remaining interpretable. While the GUIDE-OPE guidance tree suggested simply following Distill-Policy, the GUIDE-UniformExplore guidance tree suggested:

**GUIDE-UniformExplore:** If feature1 < -0.426, follow Distill-Policy. Otherwise, follow GreedyTreeSearch-HTE.

By leveraging feature1 which had so far not been used in the individual policies, GUIDE-UniformExplore returned the lowest regret of all methods, including a regret lower than that of the individual policies it ensembled, GreedyTreeSearch-HTE and Distill-Policy (Table 2). Interestingly, even if the individual policies GreedyTreeSearch-HTE and Distill-Policy were grown to greater depth, feature1 was not still not selected. While we do not always see gains from ensembling, in some datasets guidance trees can correct for the greediness of tree learning. While typical ways of ensembling trees (e.g. random forest) reduces interpretability, depth-constrained guidance trees, added to the top of an interpretable policy tree, merely makes the policy tree a bit deeper.

### 4.4. Bridging Explanations and Policies

We now present the explanations and resulting interpretable policies on the Synthetic COVID data.
Figure 4: Segments found by Distill-HTE method on synthetic COVID data.

Interpreting HTE model: Figure 4.4 presents segments found by Distill-HTE on the synthetic COVID data. The segment with the most negative predicted treatment effect (first segment; red color), at -0.0304 ± 0.0010, covers individuals who started taking the drug between 4.5 and 10.5 days after the onset of symptoms and had Aspartate Aminotransferase and Lactate Dehydrogenase levels within normal ranges (MedicineNet, 2021), suggesting that they were not extremely sick. It is therefore not surprising that they were not predicted to benefit as much from the drug.

On the other hand, the segment with the most positive predicted treatment effect (last segment; green color) covers individuals who started taking the drug soon (< 4.5 days) after exhibiting COVID symptoms. These individuals were predicted to benefit the most from the drug, with treatment effect 0.0187 ± 0.0006. This agrees with the finding that the Remdesivir drug results in a faster time to clinical improvement for the patients with shorter time from symptom onset to starting trial (Wang et al., 2020).

Policy Generation: A simple policy can be derived from the Distill-HTE segments, with all red segments assigned to not receiving the drug (as they are predicted to not benefit from it) and all green segments assigned to receiving the drug. This policy is different from the interpretable policies learned directly (Table 2). For example, the GreedyTreeSearch-HTE policy is extremely simple: if days from symptoms onset to starting trial < 11, assign to receive the drug. Otherwise, assign to not receive the drug. Another example is No-HTE-Greedy and No-HTE-Iterative whose rather stringent splitting criteria (Equation 3) did not find enough heterogeneity worth assigning different treatment groups to, and assigned all individuals to receive the drug.

5. Discussion and Conclusion

The motivation for this work was three-fold. (1) HTE models sometimes overfit on extremely large datasets, due to having a large number of data points, extremely noisy outcomes, noisy features, etc. (2) Interpretable policies can avoid much of the tech-debt ML-based policies are known to incur
Distilling Heterogeneity (Sculley et al., 2015b). (3) Scalability constraints excluded many optimal tree-based policy learning algorithms from deployment in production environments, as illustrated in Section 3.4.

The interpretable policy methods proposed in this paper are not theoretically optimal, a shortcoming we acknowledge. Nonetheless, we believe that the comparison to sensible baselines, such as policies derived from T-Learner HTE models, and the ability of some policies to achieve low regret on simple synthetic datasets, demonstrates their merit.

In real, large-scale datasets where the potential benefit from personalization was higher, No-HTE-Greedy and No-HTE-Iterative suggested more personalized policies, especially when feature selection was performed beforehand. This is because the splitting criteria included all features. If feature selection is not performed, this, together with using sample average treatment effects as our splitting criteria makes it easier to find a good split on the training dataset by chance but not on the test set. Further work is needed to realize the full potential of these methods.

In practice, we have seen that the choice between deriving policies from black-box HTE models and directly learning ground-up interpretable policies depends on the ability of the individuals utilizing such policies to maintain HTE models in production, and the need to explain how exactly personalization is happening, especially for applications with where personalization based on sensitive features incurs disparate treatment unfairness (Lal et al., 2020). While HTE models are resource and maintenance intensive, the ability to continuously retrain the model allows for adjustment to a dynamic user population. Conversely, interpretable policies are easy to implement and maintain, but may not perform best over the long-term without policy regeneration as the features of the user population shift.
Appendix A. Data Generation Details

The Email Marketing dataset (Hillstrom, 2008) is a real dataset that came from a randomized experiment, where customers were randomized into receiving one of three treatments. To generate potential outcomes, for each individual $i$ in treatment group $k$, we searched for the 5-nearest-neighbors in treatment group $k'$, averaging their outcomes to get the potential outcome for individual $i$ for treatment group $k'$. To compute distance between individuals, we used Euclidean distance in feature space.

Appendix B. Training Details

Unless otherwise mentioned, the HTE models we train are T-learners consisting of GBDT base learners. In general, we use 40% of the data as the test set on which we report results, and 30% of the remaining 60% as the validation set. We learn individual policies on the training set. When learning ensemble policies, we learn the ensemble on the validation set.

Appendix C. Algorithmic details

Algorithm 3 No-HTE-model policy generation: greedy implementation (No-HTE-Greedy)

**Input**: Maximal tree depth $d$, dataset $\mathcal{D}$.

**Output**: segments

1. Set depth $= 0$, segments $= \{\mathcal{D}\}$.
2. Split the segments:
   
   while depth $\leq d$ do
   
   Set nodes $:= \{\}$
   
   for $S \in$ segments do
   
   Consider splits that satisfy $\min(A(S_l), A(S_r)) > A(S)$, find the optimal split defined as follows:
   
   $$(S^*_l, S^*_r) = \arg\max_{S_l, S_r} \min(A(S_l), A(S_r)).$$
   
   If such $S^*_l$ and $S^*_r$ exist, add them to nodes; otherwise add $S$ to nodes.
   
   end
   
   Let segments $:= \text{nodes}$ and depth $:= \text{depth} + 1$.
   
   end

Algorithm 5 Off-Policy Evaluation (OPE) with Inverse Propensity Scoring

**Input**: Policy $\Pi$, dataset $\mathcal{D}$, propensity score $\{p_i\}_{i=1}^N$ for all individuals ($p_i = \frac{1}{K+1}$ for randomized data with $K + 1$ treatment groups).

**Output**: The estimated policy value.

1. Generate the index set where policy decision matches the observed treatment decision in $\mathcal{D}$:
   
   $\mathcal{I} = \{i | W_i == \Pi(X_i)\}$.
   
2. Output $\frac{1}{N} \sum_{i \in \mathcal{I}} Y_i / p_i(W_i)$. 

14
Algorithm 4 No-HTE-model policy generation: iterative implementation (No-HTE-Iterative)

**Input**: Maximal tree depth $d$, number of iterations $t$, dataset $D$.

**Output**: segments

1. Set iterations := 0 and segments = {}.
2. Add to segments iteratively
   
   **while** iterations $\leq t$ **do**
   
   Step 1 to 2 in Algorithm 3, but change all $\text{min}$ to $\text{max}$. Denote the resulting list of segments as $C$.
   
   Add to segments the $S$ in $C$ that maximizes $A(S)$ and remove the data points in $S$ from $D$.
   
   Set iterations := iterations + 1.
   
   **end**

References

Alnour Alharin, Thanh-Nam Doan, and Mina Sartipi. Reinforcement learning interpretation methods: A survey. *IEEE Access*, 8:171058–171077, 2020.

Maxime Amram, Jack Dunn, and Ying Daisy Zhuo. Optimal policy trees. *arXiv preprint arXiv:2012.02279*, 2020.

Susan F Assmann, Stuart J Pocock, Laura E Enos, and Linda E Kasten. Subgroup analysis and other (mis) uses of baseline data in clinical trials. *The Lancet*, 355(9209):1064–1069, 2000.

Susan Athey and Guido Imbens. Recursive partitioning for heterogeneous causal effects. *PNAS*, 113(27):7353–7360, 2016.

Susan Athey and Stefan Wager. Policy learning with observational data. *Econometrica*, 89(1):133–161, 2021.

Osbert Bastani, Yewen Pu, and Armando Solar-Lezama. Verifiable reinforcement learning via policy extraction. In *NeurIPS*, 2018.

Alina Beygelzimer and John Langford. The offset tree for learning with partial labels. In *KDD*, pages 129–138, 2009.

Cristian Bucilua, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *KDD*, 2006.

Rich Caruana. Multitask learning. *Machine learning*, 28(1):41–75, 1997.

Gong Chen, Hua Zhong, Anton Belousov, and Viswanath Devanarayan. A prim approach to predictive-signature development for patient stratification. *Statistics in medicine*, 34(2):317–342, 2015.

Mark W. Craven and Jude W. Shavlik. Extracting tree-structured representations of trained networks. In *NeurIPS*, 1995.

Wojciech M Czarnecki, Razvan Pascanu, Simon Osindero, Siddhant Jayakumar, Grzegorz Swirszcz, and Max Jaderberg. Distilling policy distillation. In *AISTATS*, pages 1331–1340, 2019.
Raaz Dwivedi, Yan Shuo Tan, Briton Park, Mian Wei, Kevin Horgan, David Madigan, and Bin Yu. Stable discovery of interpretable subgroups via calibration in causal studies. *International Statistical Review*, 88:S135–S178, 2020.

Charles Elkan. The foundations of cost-sensitive learning. In *IJCAI*, IJCAI’01, page 973–978, 2001.

Jared C Foster, Jeremy MG Taylor, and Stephen J Ruberg. Subgroup identification from randomized clinical trial data. *Statistics in medicine*, 30(24):2867–2880, 2011.

Nicholas Frosst and Geoffrey Hinton. Distilling a neural network into a soft decision tree. *arXiv preprint arXiv:1711.09784*, 2017.

Florent Garcin, Christos Dimitrakakis, and Boi Faltings. Personalized news recommendation with context trees. In *RecSys*, page 105–112, 2013.

Samuel Greydanus, Anurag Koul, Jonathan Dodge, and Alan Fern. Visualizing and understanding Atari agents. In *ICML*, 2018.

Nyoman Gunantara. A review of multi-objective optimization: Methods and its applications. *Cogent Engineering*, 5(1):1502242, 2018.

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

Kevin Hillstrom. Minethatdata e-mail analytics and data mining challenge. [https://www.uplift-modeling.com/en/v0.3.1/api/datasets/fetch_hillstrom.html](https://www.uplift-modeling.com/en/v0.3.1/api/datasets/fetch_hillstrom.html), 2008. Accessed October 19, 2021.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. In *NeurIPS Deep Learning and Representation Learning Workshop*, 2014.

Kosuke Imai and Marc Ratkovic. Estimating treatment effect heterogeneity in randomized program evaluation. *The Annals of Applied Statistics*, 7(1):443–470, 2013.

Kosuke Imai and Aaron Strauss. Estimation of heterogeneous treatment effects from randomized experiments, with application to the optimal planning of the get-out-the-vote campaign. *Political Analysis*, 19(1):1–19, 2011.

Nathanael Jo, Sina Aghaei, Andrés Gómez, and Phebe Vayanos. Learning optimal prescriptive trees from observational data. *arXiv preprint arXiv:2108.13628*, 2021.

Fredrik Johansson, Uri Shalit, and David Sontag. Learning representations for counterfactual inference. In *ICML*, pages 3020–3029, 2016.

Nathan Kallus and Angela Zhou. Policy evaluation and optimization with continuous treatments. In *AISTATS*, pages 1243–1251, 2018.

Edward H. Kennedy. Optimal doubly robust estimation of heterogeneous causal effects, 2020.
Distilling Heterogeneity

Toru Kitagawa and Aleksey Tetenov. Who should be treated? empirical welfare maximization methods for treatment choice. *Econometrica*, 86(2):591–616, 2018.

Sören R Künnel, 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.

G Roshan Lal, Sahin Cem Geyik, and Krishnaram Kenthapadi. Fairness-aware online personalization. *arXiv preprint arXiv:2007.15270*, 2020.

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. In *NeurIPS*, 2020. [https://github.com/vanderschaarlab/mlforhealthlabpub/tree/main/alg/r2p-hte](https://github.com/vanderschaarlab/mlforhealthlabpub/tree/main/alg/r2p-hte).

Benjamin Letham and Eytan Bakshy. Bayesian optimization for policy search via online-offline experimentation. *J. Mach. Learn. Res.*, 20:145–1, 2019.

Wei-Yin Loh, Luxi Cao, and Peigen Zhou. Subgroup identification for precision medicine: A comparative review of 13 methods. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 9(5):e1326, 2019.

Edward McFowland, Sandeep Gangarapu, Ravi Bapna, and Tianshu Sun. A prescriptive analytics framework for optimal policy deployment using heterogeneous treatment effects. *Forthcoming at MIS Quarterly*, 2020.

MedicineNet. Liver function tests (normal, low, and high ranges & results). [https://www.medicinenet.com/liver_blood_tests/article.htm], 2021. Accessed October 19, 2021.

Lina Montoya, Mark van der Laan, Alexander Luedtke, Jennifer Skeem, Jeremy Coyle, and Maya Petersen. The optimal dynamic treatment rule superlearner: Considerations, performance, and application. *arXiv preprint arXiv:2101.12326*, 2021.

Chirag Nagpal, Dennis Wei, Bhanukiran Vinzamuri, Monica Shekhar, Sara E Berger, Subhro Das, and Kush R Varshney. Interpretable subgroup discovery in treatment effect estimation with application to opioid prescribing guidelines. In *Proceedings of the ACM Conference on Health, Inference, and Learning*, pages 19–29, 2020.

Hongseok Namkoong, Samuel Daulton, and Eytan Bakshy. Distilled thompson sampling: Practical and efficient thompson sampling via imitation learning. *arXiv preprint arXiv:2011.14266*, 2020.

Jersey Neyman. Sur les applications de la théorie des probabilités aux experiences agricoles: Essai des principes. *Roczniki Nauk Rolniczych*, 10:1–51, 1923.

Xinkun Nie and Stefan Wager. Quasi-oracle estimation of heterogeneous treatment effects. *Biometrika*, 108(2):299–319, 2021.

Min Qian and Susan A Murphy. Performance guarantees for individualized treatment rules. *Annals of Statistics*, 39(2):1180, 2011.
A. H. Qureshi, Yutaka Nakamura, Y. Yoshikawa, and H. Ishiguro. Show, attend and interact: Perceivable human-robot social interaction through neural attention q-network. *ICRA*, 2017.

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

D. Sculley, Gary Holt, Daniel Golovin, Eugene Davydov, Todd Phillips, Dietmar Ebner, Vinay Chaudhary, Michael Young, Jean-François Crespo, and Dan Dennison. Hidden technical debt in machine learning systems. In *NeurIPS*, volume 28, 2015a.

David Sculley, Gary Holt, Daniel Golovin, Eugene Davydov, Todd Phillips, Dietmar Ebner, Vinay Chaudhary, Michael Young, Jean-François Crespo, and Dan Dennison. Hidden technical debt in machine learning systems. *Advances in neural information processing systems*, 28:2503–2511, 2015b.

Claudia Shi, David M Blei, and Victor Veitch. Adapting neural networks for the estimation of treatment effects. In *NeurIPS*, 2019.

Aleksandrs Slivkins. Introduction to multi-armed bandits, 2021.

Yang Song and George YH Chi. A method for testing a prespecified subgroup in clinical trials. *Statistics in medicine*, 26(19):3535–3549, 2007.

Adith Swaminathan and Thorsten Joachims. Counterfactual risk minimization: Learning from logged bandit feedback. In *ICML*, pages 814–823, 2015.

Sarah Tan, Rich Caruana, Giles Hooker, Paul Koch, and Albert Gordo. Considerations when learning additive explanations for black-box models. *arXiv preprint arXiv:1801.08640*, 2018a.

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

H. Toğuc and R. S. Kuzu. Hybrid models of factorization machines with neural networks and their ensembles for click-through rate prediction. In *2020 5th International Conf. on Computer Science and Engineering (UBMK)*, pages 31–36, Hangzhou, China, 2020. ICEECS. doi: 10.1109/UBMK50275.2020.9219371.

Marko Vasic, Andrija Petrovic, Kaiyuan Wang, Mladen Nikolic, Rishabh Singh, and Sarfraz Khurshid. Moêt: Interpretable and verifiable reinforcement learning via mixture of expert trees. *arXiv preprint arXiv:1906.06717*, 2019.

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

Tong Wang and Cynthia Rudin. Causal rule sets for identifying subgroups with enhanced treatment effect. *INFORMS Journal on Computing*, 2021.
Yeming Wang, Dingyu Zhang, Guanhua Du, Ronghui Du, Jianping Zhao, Yang Jin, Shouzhi Fu, Ling Gao, Zhenshun Cheng, Qiaofa Lu, et al. Remdesivir in adults with severe covid-19: a randomised, double-blind, placebo-controlled, multicentre trial. The Lancet, 395(10236):1569–1578, 2020.

Ziyu Wang, Tom Schaul, Matteo Hessel, Hado Van Hasselt, Marc Lanctot, and Nando De Freitas. Dueling network architectures for deep reinforcement learning. In ICML, 2016.

Yingqi Zhao, Donglin Zeng, A John Rush, and Michael R Kosorok. Estimating individualized treatment rules using outcome weighted learning. Journal of the American Statistical Association, 107(499):1106–1118, 2012.

Zhengyuan Zhou, Susan Athey, and Stefan Wager. Offline multi-action policy learning: Generalization and optimization. arXiv preprint arXiv:1810.04778, 2018.