

The Holdout Randomization Test for Feature Selection in Black Box Models

Wesley Tansey^{*1}, Victor Veitch³, Haoran Zhang⁴, Raul Rabadan², and David M. Blei^{1,3,5}

¹Department of Epidemiology & Biostatistics, Memorial Sloan Kettering Cancer Center, New York, NY, USA

²Department of Systems Biology, Columbia University Medical Center, New York, NY, USA

³Department of Statistics, Columbia University, New York, NY, USA

⁴Department of Computer Science, University of Texas at Austin, Austin, TX, USA

⁵Department of Computer Science, Columbia University, New York, NY, USA

⁶Data Science Institute, Columbia University, New York, NY, USA

Abstract

We propose the holdout randomization test (HRT), an approach to feature selection using black box predictive models. The HRT is a specialized version of the conditional randomization test (CRT) [Candes et al., 2018] that uses data splitting for feasible computation. The HRT works with any predictive model and produces a valid p -value for each feature. To make the HRT more practical, we propose a set of extensions to maximize power and speed up computation. In simulations, these extensions lead to greater power than a competing knockoffs-based approach, without sacrificing control of the error rate. We apply the HRT to two case studies from the scientific literature where heuristics were originally used to select important features for predictive models. The results illustrate how such heuristics can be misleading relative to principled methods like the HRT. Code is available at <https://github.com/tansey/hrt>.

1 Introduction

A key scientific problem is sifting through many candidate features to find explanatory signals in data. Often, predictive models are used for this task: the model is fit, error on heldout data is measured, and strong performing models are assumed to have discovered some fundamental properties of the system under study. A heuristic method (e.g. tree membership counts in random forests or coefficient magnitudes in lasso models) is then used to rank important features, with top features reported as discoveries [cf. Barretina et al., 2012, Keller et al., 2017]. However, such heuristics provide no statistical guarantees and can result in many false positives when features are correlated. This paper presents a principled alternative to feature selection heuristics by casting the task as a multiple hypothesis testing problem.

Formally, we consider the task of selecting the subset of features (X) that are relevant to an outcome (Y). The inferential goal is to conduct a conditional independence test for each feature X_j , with the null hypothesis:

$$H_0: X_j \perp\!\!\!\perp Y \mid X_{-j}, \quad (1)$$

^{*}tanseyw@mskcc.org (corresponding author)

where X_{-j} is every feature in X except X_j . Under H_0 , the feature contains no information about Y that is not contained in the other features. Features for which the null hypothesis is rejected are labeled as discoveries (potential drivers of the observed response).

To test the hypothesis in Eq. (1), Candes et al. [2018] proposed the conditional randomization test (CRT). The CRT works by repeatedly sampling and evaluating null samples \tilde{X}_j that are conditionally independent of Y . The null samples are drawn from the *complete conditional* $P(X_j | X_{-j})$, the distribution of the j^{th} feature given all other features. The complete conditional serves as a valid null model for Eq. (1) by preserving the joint dependency structure of $P(X)$ and removing any dependency between X_j and Y . To evaluate each sample, the CRT compares a test statistic $T(\tilde{X}_j, X_{-j}, Y)$ on the null samples to the statistic on the observed sample $T(X_j, X_{-j}, Y)$. We recall the CRT in Algorithm 1.

Algorithm 1 Conditional Randomization Test

```

1: procedure CRT(features  $X$ , response  $Y$ , target  $j$ , test statistic  $T$ , null draws  $K$ )
2:   Compute the test statistic,  $t \leftarrow T(X_j, X_{-j}, Y)$ .
3:   for  $k \leftarrow 1, \dots, K$  do
4:     Sample  $\tilde{X}_j \sim P(X_j | X_{-j})$ .
5:     Compute the null statistic,  $\tilde{t}^{(k)} \leftarrow T(\tilde{X}_j, X_{-j}, Y)$ .
6:   return A (one-sided)  $p$ -value,
```

$$\hat{p}_j = \frac{1}{K+1} \left(1 + \sum_{k=1}^K \mathbb{I}[t \geq \tilde{t}^{(k)}] \right)$$

The vanilla CRT is statistically flexible but computationally expensive. The algorithm makes no assumptions on the form of the relationship between Y and X . It also works with any choice of test statistic T . Yet for many choices of T , the CRT will involve substantial computation. For example, Candes et al. [2018] propose running the lasso and evaluating the magnitude of β_j , the coefficient for X_j . This requires re-running the lasso model for every null sample \tilde{X}_j . In large datasets, fitting even a single model can be expensive. Refitting an expensive model thousands of times to run a CRT is prohibitive in many data analysis scenarios.

This paper proposes an alternative: the holdout randomization test (HRT). The HRT is a specialized CRT designed to work in large datasets where it may be expensive to fit a predictive model. At a high level, the HRT works by splitting the data into train (X, Y) and test (X', Y') sets. A predictive model π is fit on the training set and repeatedly evaluated on the test set. The prediction quality of $\pi(X')$ is compared to $\pi(\tilde{X}')$, where \tilde{X}' is a copy of X' that replaces the feature X'_j to be tested with a null sample \tilde{X}'_j . Intuitively, if X'_j is predictive for Y then replacing it with a null sample \tilde{X}'_j is likely to lead to worse predictions. A p -value for the hypothesis test is then approximated by repeatedly resampling \tilde{X}_j and comparing predictive performance under the null with performance using the original data. The HRT procedure is presented in Algorithm 2.

Empirical risk on held out data is the HRT test statistic. Splitting the data into train and test sets decouples model fitting from the hypothesis testing procedure. From the CRT perspective, the model is fixed and only evaluated on the test data. By decoupling model fitting and testing, the HRT makes the CRT computationally feasible for a larger class of

Algorithm 2 Holdout Randomization Test

- 1: **procedure** HRT(training data (X, Y) , test data (X', Y') , model π , empirical risk function \mathcal{G} , null draws K)
 - 2: Fit the model π on the training data.
 - 3: Compute the empirical risk on held out data, $t \leftarrow \mathcal{G}(X', Y', \pi(X'))$.
 - 4: **for** $k \leftarrow 1, \dots, K$ **do**
 - 5: Sample $\tilde{X}'_j \sim P(X_j | X_{-j})$.
 - 6: Compute the empirical risk, $\tilde{t}^{(k)} \leftarrow \mathcal{G}(\tilde{X}', Y', \pi(\tilde{X}'))$.
 - 7: **return** A (one-sided) p -value,

$$\hat{p}_j = \frac{1}{K+1} \left(1 + \sum_{k=1}^K \mathbb{I}[t \geq \tilde{t}^{(k)}] \right)$$

predictive models and data analysis tasks.

The empirical risk test statistic makes the HRT applicable to any predictive model. There is no need to provide a model-specific feature importance heuristic like coefficient magnitudes or decision tree inclusion statistics. The HRT only needs to query the model for predictions in order to perform feature selection. This decoupling means the scientist does not need knowledge of the internals of the predictive model. It suffices to be able to fit a model and use it to make predictions. Being model agnostic facilitates adoption of the HRT into scientific analysis pipelines that use black box machine learning models.

The remainder of the paper addresses several pragmatic questions regarding the HRT:

- Is it possible to use the entire dataset, rather than splitting it into train and test, so as to maximize power? (Sections 3.1 and 3.2)
- Does the complete conditional need to be known or can we know it only proportionally? (Section 3.3)
- Can we speed up the algorithm further? (Section 3.4)
- How does model predictive performance relate to feature selection power? (Section 4.3)
- If we estimate the conditional $P(X_j | X_{-j})$ from data, will the HRT be robust to estimation errors? If not, can we somehow err on side of caution and produce conservative p -values? (Appendix A)

2 Connections to existing work

Most classical work on feature selection relies on strong parametric assumptions on the response function. For instance, the train-test splitting procedure of the HRT resembles the data splitting procedure of Wasserman and Roeder [2009] for high-dimensional feature selection in linear models, with the cross-validation HRT extension (Section 3.1) then mirroring the multiple re-splitting extension of Meinshausen et al. [2009]. However, the HRT is focused on the case where no parametric model is assumed. A flurry of recent techniques have been proposed to extract information from such black box predictive models. We relate the HRT to these methods.

Many approaches are model-specific. Examples include iterative random forests [Basu et al., 2018], Bayesian kernel regression [Crawford et al., 2018], and Bayesian neural networks [Liang et al., 2018]. These approaches constrain the scientist to a specific modeling choice, rather than allowing the scientist to choose the best model or build a custom one specific for their problem. The HRT allows the scientist to build the best predictive model they can, with reassurance that strong-performing models are likely to have high power.

Other methods focus on interpretation of the model, rather than testing for conditional independence. These include LIME [Ribeiro et al., 2016], DeepLIFT [Shrikumar et al., 2017], SHAP [Lundberg and Lee, 2017], and L2X [Chen et al., 2018]. SHAP and L2X are related to the HRT in that they measure the conditional mutual information between each feature and the response. This can be seen as an optimization variant of the HRT where the statistic is change in prediction rather than prediction accuracy; however, for computational purposes both methods make simplifying independence assumptions between covariates that may lead to false positives. Even with a correct model for measuring mutual information, interpretability of the model is a related but different question. A poor model may use some true-null feature to make its predictions, in which case a correct interpretation would be to flag that variable as important to the model’s prediction, even though it is conditionally independent of the true label. An incorrect model is not a problem for the HRT; it will simply have low power to detect the true signals. See Fisher et al. [2018] for an in-depth discussion on different notions of variable importance.

The leave-one-covariate-out (LOCO) inference method [Lei et al., 2018] is a nonparametric method that assesses feature importance. LOCO inference works in the conformal inference setting [Vovk, 2012]. Conformal inference provides coverage guarantees marginally over X , whereas the HRT is valid conditioned on $X = x$. Further, the LOCO test is focused on assessing the impact of a feature X_j on the mean of the response Y . By contrast, the HRT admits any test statistic. For instance, a variable may be predictive of higher or lower variance in a response. This could be detected by the HRT but LOCO would have zero power to reject the null for this variable.

A handful of methods exist for feature selection in arbitrary black box models.

Conditional randomization tests (CRTs) Candes et al. [2018] proposed CRTs as a generic approach for performing the conditional independence test in Eq. (1) using any test statistic and any predictive model. A CRT repeatedly samples from the conditional null for the feature being tested and compares the test statistic under the null with the test statistic of the original data. For instance, the example test statistic in Candes et al. [2018] was the feature coefficient magnitude in a lasso regression. But this choice comes at a computational cost: refitting the model. In order to calculate the lasso coefficient statistic under a null sample, the regression must be re-run using the null data. Repeating this computation many times to approximate the feature p -value is prohibitively expensive. The basic HRT is a special case of the CRT, where the test statistic is carefully chosen to avoid the need to refit after every null sample.

Model-X knockoffs Although Candes et al. [2018] originally proposed the CRT, they discarded it in favor of the *model-X knockoffs* approach, which avoids the computational issues outlined above. In the model-X knockoffs framework, a “knockoff” feature is generated for every feature in the dataset and the model is fit using both the original and knockoff features. Generating the knockoffs requires knowing (or estimating) the joint distribution

over the features; selection is performed by a model-specific variable importance function for the original and knockoff features. The knockoffs procedure has been extended to different feature distributions than the original multivariate normal, such as hidden Markov models [Sesia et al., 2017] and Gaussian mixture models [Gimenez et al., 2018]. Machine learning approaches have also been proposed that use deep learning models to fit flexible knockoff distributions [Romano et al., 2019, Jordon et al., 2019]. Others have suggested altering certain black box models to support knockoffs, such as using a special input layer in neural networks [Lu et al., 2018]. While knockoffs have the appeal of computational efficiency over a generic CRT, Candes et al. [2018] showed in experiments that CRTs have higher power. The HRT aims to bridge the gap between the two procedures. It makes CRTs computationally tractable without sacrificing power.

Mimic and Classify Sen et al. [2018] propose “Mimic and Classify,” an approach inspired by generative adversarial networks [Goodfellow et al., 2014]. Mimic and Classify fits a conditional model of the feature, similarly to a CRT, but then also fits a “discriminator” model to distinguish between samples from the true dataset and samples from the null model, with the difference in error magnitude as the test statistic. This approach is conceptually similar to the HRT but differs in key ways: i) it does not leverage a pre-built predictive model, ii) it does not provide direct p -value calculation, iii) it requires fitting a discriminator per hypothesis test, each of which may be as difficult to build as a predictive model for the response, and iv) it implicitly ties the magnitude of the dependency between X and Y to the test statistic, whereas the HRT is able to detect small but consistent performance differences between the original data and the null samples. Since Mimic and Classify does not provide a way to control the error rate at a target level, we do not consider it in our comparisons.

Conditional permutation tests (CPTs) Berrett et al. [2018] propose the CPT as a robust choice of CRT. In addition to conditioning on the other features, the CPT also conditions on the order statistics when drawing randomizations. Thus, rather than randomly shuffling the order statistics like in a classical permutation test, each feature is shuffled in a non-uniform way. CPTs are an interesting approach and are complementary to the HRT. In particular, they are more robust than vanilla CRTs and may increase the robustness of the HRT even further. In Section 3.3, we merge the HRT and the CPT to create the holdout permutation test (HPT). We leverage the finite support of the empirical permutation distribution to develop a caching scheme that speeds up the MCMC algorithm from Berrett et al. [2018]. In Section 3.4, we further extend this idea by considering combinations of discrete points—rather than permutations—to develop an HRT method that is faster than the original HRT or the HPT, while still maintaining finite-sample validity.

Finally, Shah and Peters [2018] show that there is no free lunch for conditional independence testing. It is impossible to develop a test with nontrivial power without assuming something about the joint distribution of (Y, X) . We emphasize that neither the HRT nor the above methods avoid this result. The assumption made in the HRT is that a reasonable approximation to the complete conditional distribution of X_j given X_{-j} can be obtained. Berrett et al. [2018] provide a theoretical investigation under this assumption for CPTs.

Algorithm 3 Cross-Validation Holdout Randomization Test

```
1: procedure CV-HRT(data split into  $M$  folds:  $\{(X^{(m)}, Y^{(m)})\}_{m=1}^M$ , model  $\pi$ , empirical  
risk function  $\mathcal{G}$ , null draws  $K$ )  
2:   Initialize  $t \leftarrow 0$   
3:   for  $m \leftarrow 1, \dots, M$  do  
4:     Run the fold-specific HRT,  $p_j^{(m)} \leftarrow \text{HRT}((X^{(-m)}, Y^{(-m)}), (X^{(m)}, Y^{(m)}), \pi, \mathcal{G}, K)$ .  
5:   return A multiplicity-corrected  $p$ -value,
```

$$\hat{p}_j = M \times \min(p_j^{(1)}, \dots, p_j^{(M)}). \quad (2)$$

3 Extensions to boost power and reduce computation

The HRT provides valid p -values under the assumption that we have access to the true or well-estimated complete conditional distributions. Yet even with access to the complete conditionals, practical limitations to using the HRT arise. For instance, the availability of test data and the time to calculate the test statistics are finite. Thus, if the predictive model is computationally expensive to run and the testing dataset is small, the basic HRT may be slow and have low power to detect nonnull features. This section presents extensions to the HRT for improving the statistical and computational performance of the basic algorithm in light of these practical limitations. The empirical performance of these improvements is evaluated in Section 4.

3.1 The cross-validation HRT

Modeling pipelines typically allocate the bulk of the total data to the training set, leaving only 10–20% for testing. Since the HRT only uses the test set and the trained model, the majority of the data will not be used for testing. In scientific settings where sample sizes are small-to-moderate, discarding this much data may result in the HRT having low power. Here we show how the train-test paradigm of the basic HRT can be extended to a cross-validation paradigm that uses the entire dataset.

The cross-validation holdout randomization test (CV-HRT) is presented in Algorithm 3. The CV-HRT splits the data into M folds, rather than the train-test split of the basic HRT. The algorithm fits M models, with the m^{th} model trained using the m^{th} fold as the holdout set. For each model, a separate HRT p -value is run using the corresponding cross-validation fold as a test set; this yields M fold-specific p -values.

Since the data are re-used in the cross-validation procedure, it is possible that there will be dependency between the fold-specific p -values. To correct for this possible dependency, the CV-HRT applies a Bonferroni correction to the p -values in Eq. (2) and returns the smallest corrected value. The Bonferroni correction ensures that the resulting p -value from the CV-HRT retain finite-sample control of the target error rate (i.e. it will stochastically dominate a $U(0, 1)$ random variable).

3.2 Higher power via an approximate cross-validation HRT

The Bonferroni correction in the CV-HRT is stringent. However, it is necessary to maintain strict control over the finite-sample error rate due to the potential for dependence between

Algorithm 4 Approximate Cross-Validation Holdout Randomization Test

```

1: procedure CV-HRT(data split into  $M$  folds:  $\{(X^{(m)}, Y^{(m)})\}_{m=1}^M$ , model  $\pi$ , empirical
   risk function  $\mathcal{G}$ , null draws  $K$ )
2:   Initialize  $t \leftarrow 0$ 
3:   for  $m \leftarrow 1, \dots, M$  do
4:     Fit  $\pi^{(m)}$  using  $(X^{(-m)}, Y^{(-m)})$ .
5:     Add the fold empirical risk,  $t \leftarrow t + \mathcal{G}(X^{(m)}, Y^{(m)}, \pi^{(m)}(X^{(m)}))$ .
6:   for  $k \leftarrow 1, \dots, K$  do
7:     Sample  $\tilde{X}_j \sim P(X_j | X_{-j})$ .
8:     Initialize  $\tilde{t}^{(k)} \leftarrow 0$ 
9:     for  $m \leftarrow 1, \dots, M$  do
10:      Add the fold empirical risk on the randomized data,

$$\tilde{t}^{(k)} \leftarrow \tilde{t}^{(k)} + \mathcal{G}(\tilde{X}^{(m)}, Y^{(m)}, \pi^{(m)}(\tilde{X}^{(m)})) .$$

11:   return A (one-sided)  $p$ -value,
```

$$\hat{p}_j = \frac{1}{K+1} \left(1 + \sum_{k=1}^K \mathbb{I} [t \geq \tilde{t}^{(k)}] \right)$$

the cross-validation p -values. In practice, we find that the dependence between the p -values is minimal for even small sample sizes. To boost power, we therefore propose an approximate cross-validation extension to the HRT that merges test statistics without correction.

The approximate cross-validation holdout randomization test is presented in Algorithm 4. As in the valid CV-HRT, the approximate CV-HRT splits the data into M folds, fits M models, and uses the m^{th} model to make predictions for samples in the m^{th} fold. Instead of calculating fold-specific p -values and merging them, the approximate CV-HRT calculates the average empirical risk across all folds as the test statistic. This leads to a higher-power, lower-variance estimate of the usefulness of the j^{th} feature in predicting Y .

The theoretical validity of the approximate CV-HRT in Algorithm 4 is less clear than the provably-valid CV-HRT in Algorithm 3. Both methods take full advantage of the dataset by building multiple models, with one valid test model per sample. The combination of these models in the case of the approximate CV-HRT, however, produces dependency between each model's predictions under the null. In simulation and in the benchmarks in Section 4, Algorithm 4 produces valid or even conservative p -values. Further, because the approximate CV-HRT does not require the Bonferroni correction, the power of the procedure is increased.

Stronger predictive models in any HRT procedure are likely to capture more of the dependency structure between X and Y . Consequently, more folds leads to more training data per model, which in turn should lead to better predictive performance per model. For the valid CV-HRT, the increase in folds comes with the tradeoff of a larger multiplicity correction term. For the approximate CV-HRT, leave-one-out cross-validation will maximize power, but this comes at the computational expense of fitting n^* models, which may be prohibitive. In practice, we observe $M = 5$ folds is sufficient to achieve high power in both algorithms.

Algorithm 5 Holdout Permutation Test

```

1: procedure HPT(training data  $(X, Y)$ , test data  $(X', Y') = \{(X'_i, Y'_i)\}_{i=1}^{n'}$ , model  $\pi$ ,  

   empirical risk function  $\mathcal{G}$ , null draws  $K$ , MCMC steps  $C$ )
2:   Fit the model  $\pi$  on the training data.
3:   Initialize  $n' \times n'$  matrices,  $\mathbf{P}$  and  $\mathbf{T}$ .
4:   Compute the empirical risk on held out data,  $t \leftarrow \mathcal{G}(X', Y', \pi(X'))$ .
5:   for  $i \leftarrow 1, \dots, n'$  do
6:     for  $s \leftarrow 1, \dots, n'$  do
7:       Let  $\tilde{X}'_i$  be the sample  $X'_i$  with  $X'_{ij}$  replaced with  $X'_{sj}$ .
8:       Compute the sample probability,  $\mathbf{P}_{is} \leftarrow P(X_j = \tilde{X}'_{ij} | X_{-j} = \tilde{X}'_{i,-j})$ .
9:       Compute empirical risk,  $\mathbf{T}_{is} \leftarrow \mathcal{G}(\tilde{X}'_i, Y'_i, \pi(\tilde{X}'_i))$ .
10:      for  $k \leftarrow 1, \dots, K$  do
11:        Sample permutation  $Z$  via  $C$  MCMC steps [Berrett et al., 2018, Algorithm 1]  

         using cached  $\mathbf{P}$ .
12:        Compute the empirical risk,  $\tilde{t}^{(k)} \leftarrow \frac{1}{n'} \sum_{i=1}^{n'} \mathbf{T}_{i,\text{rank}(Z_i)}$ .
13:      return A (one-sided)  $p$ -value,

```

$$\hat{p}_j = \frac{1}{K+1} \left(1 + \sum_{k=1}^K \mathbb{I} [t \geq \tilde{t}^{(k)}] \right)$$

3.3 The holdout permutation test

The conditional permutation test (CPT) of Berrett et al. [2018] is an alternative to the conditional randomization test. The two tests are similar, but the CPT considers the permutation distribution of the feature X_j , rather than the total support of the independent complete conditionals. The CPT offers some benefits over the CRT, namely in terms of robustness to misspecification of the complete conditionals. However, the CPT requires a costly Markov chain Monte Carlo (MCMC) algorithm to sample over the conditional permutation distribution [Berrett et al., 2018, Algorithm 1]. The CPT algorithm may therefore be computationally prohibitive when the test set size n' is large, the complete conditionals are expensive to evaluate, or the number of null samples required is large. Here we show how to adapt the HRT to the conditional permutation test to improve its scalability for these cases while still retaining the finite-sample validity of the CPT.

Algorithm 5 presents the holdout permutation test (HPT). The algorithm works similarly to the basic HRT, but leverages the finite support of the permutation distribution to improve scalability. For each sample in the test set, the null distribution only contains at most n' unique values.¹ The HPT caches the probability of each unique value and the empirical risk of the predictive model under that value (Lines 4–9). This requires $\mathcal{O}(n'^2)$ operations. Once cached, the sampling of new values and computing null test statistics can be done using only the cached values. Thus, when the number of null samples K and MCMC samples M are large, such that $KC > (n')^2$, the caching scheme in the HPT will be faster than the naive algorithm. Candes et al. [2018] show that K must be on the order of $1/\tau$, where $\tau = qR/p$ is

¹We assume the complete conditional distributions are continuous. If they are not, additional computational speedups can be gained. Validity of the procedure for discrete distributions still holds under Theorem 1 of Berrett et al. [2018].

Algorithm 6 Holdout Grid Test

```

1: procedure HGT(training data  $(X, Y)$ , test data  $(X', Y') = \{(X'_i, Y'_i)\}_{i=1}^{n'}$ , model  $\pi$ , empirical risk function  $\mathcal{G}$ , null draws  $K$ , grid size  $S$ )
2:   Fit the model  $\pi$  on the training data.
3:   Compute the empirical risk on held out data,  $t \leftarrow \mathcal{G}(X', Y', \pi(X'))$ .
4:   Initialize  $n' \times (S + 1)$  matrices,  $\mathbf{P}$  and  $\mathbf{T}$ .
5:   for  $i \leftarrow 1, \dots, n'$  do
6:     Set the first column using the observed data,

$$\mathbf{P}_{i1} \leftarrow P(X_j = X'_{ij} | X_{-j} = X'_{i,-j}), \quad \mathbf{T}_{i1} \leftarrow \mathcal{G}(X'_i, y'_i, \pi(X'_i)).$$

7:     Construct an  $S$  grid,  $Z = (Z_1, \dots, Z_S)$ , to approximate  $P(X_j | X_{-j} = X'_{i,-j})$ .
8:     for  $s \leftarrow 1, \dots, S$  do
9:       Let  $\tilde{X}'_i$  be the sample  $X'_i$  with  $X'_{ij}$  replaced with grid point  $Z_s$ .
10:      Compute sample probability and empirical risk,

$$\mathbf{P}_{i,s+1} \leftarrow P(X_j = Z_s | X_{-j} = X'_{i,-j}), \quad \mathbf{T}_{i,s+1} \leftarrow g(\tilde{X}'_i, Y'_i, \pi(\tilde{X}'_i)).$$

11:    for  $k \leftarrow 1, \dots, K$  do
12:      Sample column indices  $\{V_i\}_{i=1}^{n'}$  proportional to their weight in  $\mathbf{P}_i$ .
13:      Compute the empirical risk,  $\tilde{t}^{(k)} \leftarrow \frac{1}{n'} \sum_{i=1}^{n'} \mathbf{T}_{i,V_i}$ .
14:    return A (one-sided)  $p$ -value,
```

$$\hat{p}_j = \frac{1}{K+1} \left(1 + \sum_{k=1}^K \mathbb{I} [t \geq \tilde{t}^{(k)}] \right)$$

the BHq cutoff for R rejected p -values. In high-dimensional feature selection problems, this number often exceeds $(n')^2$, suggesting the caching of the HPT will be useful when applying the CPT large feature sets.

3.4 Faster computation via the holdout grid test

The HRT relies on random sampling to approximate the expectation of the indicator function in Algorithm 2. Unlike other randomization tests, however, the HRT assumes a fixed model and a test statistic that is a summation of independent components. These two properties can be leveraged to substantially speed up the computation of the p -value approximation, in a manner similar to the caching strategy of the HPT.

Algorithm 6 presents the holdout grid test (HGT). The HGT first creates a finite grid approximation \mathbf{P} to the complete conditional for each sample in the test dataset. It then queries the model at each point to create a matrix \mathbf{T} of cached risk scores. In the test statistic computation loop (Lines 11–13), the cached scores are sampled proportional to their likelihood in the complete conditional. The HGT then averages these cached scores rather than querying the predictive model. This lowers the computational complexity from $\mathcal{O}(Kn')$ to $\mathcal{O}(Sn')$. In the case of discrete distributions, the discretization step is unnecessary; the HGT would still benefit computationally from the caching strategy, however. As with the HPT, when K is large (e.g. in the case of many candidate features being tested) the HGT

is 1–2 orders of magnitude faster than the basic HRT.

The HGT is a valid extension of the HRT. The following theorem states this formally,

Theorem 1. *The p -values generated by Algorithm 6 are super-uniform if $H_0: Y \perp\!\!\!\perp X_j | X_{-j}$ is true.*

We can view the grid caching scheme as a coarsening of X'_i for each sample i . Define a new random variable W_i whose support is over the discrete set of values in the i^{th} grid. The local null hypothesis then becomes

$$\tilde{H}_0^{(i)}: Y \perp\!\!\!\perp W_i | X_{-j} = X'_{i,-j},$$

and Algorithm 6 generates a p -value for the intersection null hypothesis,

$$\tilde{H}_0: \bigcap_{i=1}^{n'} \tilde{H}^{(i)}.$$

Testing \tilde{H}_0 is done by a CRT on $W = (W_1, \dots, W_{n'})$. The complete conditional distribution $P(W | X_{-j} = X'_{-j})$ is simply the product of the individual W_i complete conditionals. Since the variable W is constructed without ever looking at Y , \tilde{H}_0 is true by construction if the original null H_0 is true. The CRT in Algorithm 6 is then the HRT applied to W . \square

Note that while \tilde{H}_0 is false only if H_0 is false, the converse is not true. That is, it is possible that H_0 is false but \tilde{H}_0 is true. This will be the case if Y depends on X_j only for some subset of values not contained in the grids. Thus in general Algorithm 6 will be less powerful than the basic HRT. Simulations in Section 4.1 investigate the difference in power between the HGT and HRT.

4 Benchmarks

We investigate the basic HRT and the proposed HRT extensions in two simulation settings. First, we consider the case where the complete conditionals are known. In this setting, we show that the cross-validation HRT methods boost power and the holdout grid test speeds up computation. We then consider the case where the conditionals are unknown and must be estimated from data. Here, we show the HRT—in combination with the calibration heuristic described in Appendix A—is robust to misspecification and finite-sample estimation error. Finally, we investigate the choice of predictive model and show that the power of the HRT is directly related to the generalization error of the predictive model. Further, with the same predictive model choice (the lasso), the HRT outperforms the model-X knockoffs approach of Candes et al. [2018].

Algorithm	Splits	$n = 20$	$n = 50$	$n = 100$	$n = 200$	$n = 500$	$n = 1000$
HRT (Algorithm 2)	1	0.05	0.18	0.38	0.64	0.7	0.7
	2	0.08	0.2	0.5	0.67	0.71	0.7
	3	0.06	0.19	0.42	0.63	0.7	0.68
CV-HRT (Algorithm 3)	4	0.06	0.15	0.38	0.63	0.71	0.69
	5	0.03	0.13	0.35	0.58	0.7	0.71
	10	0	0.07	0.19	0.46	0.67	0.69
	20	0	0.02	0.13	0.26	0.61	0.69
CV-HRT (Algorithm 4)	2	0.1	0.29	0.51	0.68	0.7	0.7
	3	0.09	0.27	0.56	0.7	0.69	0.7
	4	0.14	0.29	0.58	0.7	0.72	0.69
	5	0.1	0.31	0.54	0.69	0.71	0.71
	10	0.12	0.34	0.56	0.68	0.71	0.71
	20	0.08	0.29	0.56	0.7	0.69	0.7

Table 1: True positive rate for the random forest predictive model in Section 4.1. Both cross-validation methods improve on the basic HRT in terms of sample efficiency. The approximate CV-HRT method in Algorithm 4 has the highest power as it does not use the multiple testing correction of Algorithm 3.

4.1 The HRT and extensions when conditionals are known

We benchmark the different holdout testing algorithms on a synthetic dataset sampled from a nonlinear factor model with a nonlinear response,

$$\begin{aligned}
 z_{ik} &\sim \text{Gamma}(1, 1) \\
 w_{jk} &\sim \mathcal{N}\left(0, \frac{1}{K}\right) \\
 x_{ij} &\sim \mathcal{N}\left(\mathbf{z}_i^\top \mathbf{w}_j, \sigma^2\right) \\
 y_i &= \beta_1 \tanh(x_{i1}) + 5 \tanh(\beta_2 x_{i2} + \beta_3 x_{i3}) + \epsilon_i .
 \end{aligned} \tag{3}$$

We set $\sigma = 1$, use $K = 5$ latent factors, and $\epsilon_i \sim \mathcal{N}(0, 1)$. For each method, we provide the ground truth latent factors \mathbf{Z} and \mathbf{W} ; each complete conditional is independent of the other covariates conditioned on the latent factors.

We run 100 independent trials with 3 nonnull features and 3 null features. The number of samples n^* is varied from 20 to 1000. For each trial, we run the basic HRT, valid and approximate cross-validation HRT, HPT, and HGT. The HPT uses $M = 50$ MCMC steps per null sample and the HGT uses a grid size of $S = 50$. Each method samples $K = 10000$ null samples to estimate the feature p -value. After p -value calculation, we select features using Benjamini-Hochberg (BH) [Benjamini and Hochberg, 1995] with nominal FDR level of 10%.

We analyze four predictive model choices: ordinary least squares, lasso with regularization parameter chosen via cross-validation, Bayesian ridge regression with automatic relevance determination, and random forests (RFs) with 20 trees. Results for all four models are similar. We focus here on RFs; all results for all predictive models are presented in Appendix D.

	Algorithm	$n = 20$	$n = 50$	$n = 100$	$n = 200$	$n = 500$	$n = 1000$
TPR	HRT (Algorithm 2)	0.05	0.18	0.38	0.64	0.7	0.7
	HPT (Algorithm 5)	0.07	0.19	0.39	0.63	0.71	0.71
	HGT (Algorithm 6)	0.04	0.18	0.34	0.61	0.7	0.69
Time	HRT (Algorithm 2)	5.88	5.99	6.29	6.94	9.55	14.02
	HPT (Algorithm 5)	13.16	13.72	14.04	15.50	19.75	26.39
	HGT (Algorithm 6)	0.02	0.03	0.04	0.06	0.13	0.25

Table 2: Power and timing (in seconds) results for the HRT, HPT, and HGT using random forests as the predictive model.

Cross-validation extensions. Table 1 presents the results for the two cross-validation HRT methods. As the sample size approaches $n = 1000$, the basic HRT performs as well as both cross-validation variants. When the sample size is smaller, both CV-HRT methods can outperform the basic method. For the CV-HRT in Algorithm 3, a smaller number of splits leads to higher power since the multiple testing correction term is smaller. The approximate CV-HRT in Algorithm 4, in contrast, does not account for dependence and consequently generally benefits from more splits but sees diminishing returns beyond 4 folds. Despite being only an approximate method, Algorithm 4 produces empirically valid or even conservative null test statistics and controls the FDR across all simulation settings; see Appendix D for detailed results.

Permutation and grid extensions. Table 2 presents the results for the two alternative holdout testing methods. The HGT runs orders of magnitude faster than both the HRT and HPT. This comes at no cost in power relative to the HRT; the HPT has power benefits with small sample sizes, but comes at a higher computational cost. As with the CV-HRT, all three methods control the FDR across all simulations.

4.2 Performance when conditionals are unknown

To evaluate a more practical scenario, we next study the HRT on a variant of a benchmark from a recent paper on feature selection in Bayesian neural networks [Liang et al., 2018]. In this scenario, we do not assume the complete conditionals are known and instead estimate them from the observed data. We generate 100 independent datasets of 500 samples each with a nonlinear ground truth regression model,

$$y = \sum_{j=0}^9 [w_{4j}x_{4j} + w_{4j+1}x_{4j+1} + \tanh(w_{4j+2}x_{4j+2} + w_{4j+3}x_{4j+3})] + \sigma\epsilon, \quad (4)$$

where $\sigma = 0.5$ and $\epsilon \sim \mathcal{N}(0, 1)$. The 500 features are generated to have 0.5 correlation coefficient with each other,

$$x_j = (\rho + z_j)/2, \quad j = 1, \dots, 500, \quad (5)$$

where ρ , z_j , and w_j are independently generated from $\mathcal{N}(0, 1)$.

The ground truth in (4) uses the first 40 features, representing true signals; each sample also contains 460 null features. We note that Liang et al. [2018] only had 4 signal features

in their simulation. In preliminary simulations, the HRT had nearly 100% power and 0% FDP in this setting; we extend the experiment to 40 features to make the benchmark more challenging and model comparisons more informative.

We choose this setup for several reasons: (i) it is a nonlinear ground truth, (ii) most features are not involved in the response, (iii) all features are highly correlated, and (iv) it is compatible with the available implementation of model-X knockoffs [Candes et al., 2018] for the lasso; we compare to this implementation in Section 4.3.

For all experiments, we use a mixture density network [Bishop, 1994] with 5 components as the conditional density estimator in the HRT. Note that this flexible model is effectively misspecified and, at low sample sizes such as this experiment, is highly variable in its predictions across random restarts. We use this as it represents a realistic scenario where conditional densities are likely to be poorly estimated.

For each feature, we fit a bag of $b = 100$ bootstrap estimators and use the data-adaptive calibration technique in Appendix B to choose the approximate lower and upper bounds. We use mean-squared error (MSE) as the empirical risk function. We select significant features with a target FDR of 10%, using the Benjamini-Hochberg procedure (BH) [Benjamini and Hochberg, 1995] for multiple testing correction. For BH correction to be theoretically valid, the p -values must satisfy the *positive regression dependence on a subset* (PRDS) criterion [Benjamini and Yekutieli, 2001]. The HRT provides no guarantees about the dependence between the p -values. To fully control FDR under dependence, Benjamini and Yekutieli [2001] provide a more conservative procedure (typically called BY correction). We present results using BH selection since it is the most common procedure and empirically robust; results using BY correction are provided in Appendix F.

We first compare several variants of the HRT procedure to investigate the effect of each component. For each dataset, the predictive model is a neural network with 2 hidden layers of 200 nodes each and ReLU activation functions; we fit the model using RMSprop with fixed learning rate 3×10^{-5} . We use an 80%/20% train/test split for the basic HRT and 5-fold cross-validation for the CV-HRT in Algorithm 4. Fig. 1 shows the power and FDP results for the different variants. We discuss the individual variants and results below.²

Marginal Permutation. A common approach to testing for feature importance in many applications is the classical permutation test, where the values of a given feature are shuffled. Combining this with empirical risk on held out data was suggested by Breiman [2001] as a feature selection technique. To be valid, permutation testing requires that features are independent, which is explicitly not the case in the benchmark nor in many—if not most—real world datasets. The two left-most results in Fig. 1 show performance using a permutation approach, rather than the complete conditional, in both the basic and cross-validation HRT. The misspecified independence assumption causes an extreme inflation in the empirical false discovery rate to nearly 80% instead of the target 10%.

Calibration. The middle two results in Fig. 1 show performance using a single, uncalibrated estimation of $P(X_j | X_{-j})$ in the basic and cross-validation HRT. The estimated conditional is a much better approximation of the true feature distribution than the marginal permutation above, but is still insufficient to control the empirical FDR at the target level. By contrast, the two right-most results show that the calibrated model better conserves

²Since the FDP results have high variance, the box plots show the median rather than the mean. The mean false discovery proportions for the uncalibrated and calibrated CV-HRT are 18.46% and 14.48%, respectively.

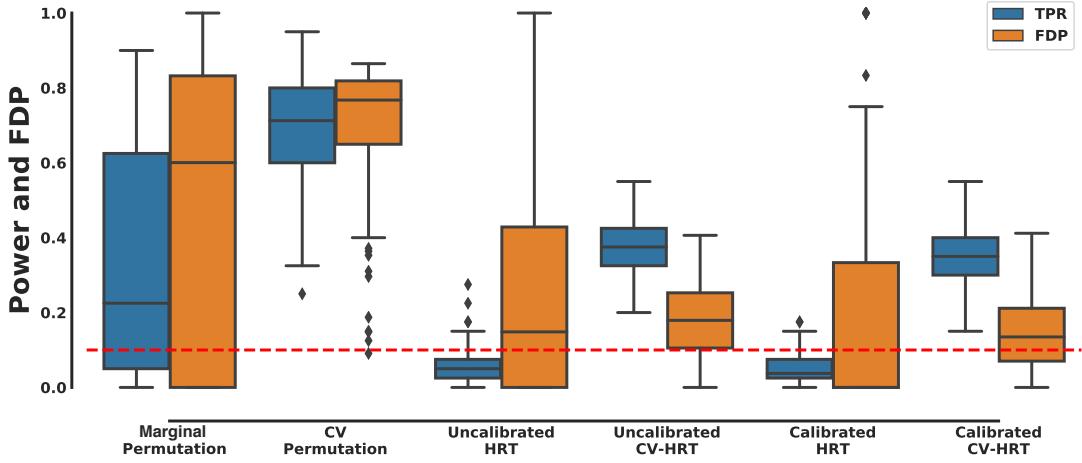


Figure 1: Power and FDP results for each HRT variant on the benchmark simulation. The calibration technique from Appendix A adjusts the sample weights to achieve tighter control over the FDR at the specified 10% level (dashed red line). Using cross-validation for predictive modeling increases power and reduces variance by using the entire dataset rather than just a held out subset.

FDR, with minimal impact on power compared to the uncalibrated model. Additional investigation of the effects of calibration are available in Appendix E.

Basic HRT vs. CV-HRT. The right-most results in Fig. 1 demonstrate the effect of using a basic train-test split HRT versus a 5-fold cross-validation HRT. Cross-validation has the effect of boosting power and reducing variance compared to the basic HRT.

4.3 Model selection and knockoffs comparison

We next compare several choices of predictive model on the same benchmark task as above. Specifically, we ran the CV-HRT procedure with the neural network model from the previous section and the following alternative predictive models:

- Ordinary least squares (OLS)
- Partial least squares (PLS) with 10 components.
- Lasso with penalty parameter chosen via 5-fold cross-validation on each training set.
- Elastic net with both penalty parameters chosen via 5-fold cross-validation on each training set.
- Bayesian ridge with normal-inverse-gamma priors for coefficients and weakly informative scale hyperpriors.
- Kernel ridge with cubic polynomial kernel with a fixed penalty weight of 1.
- Support vector with radial basis function (RBF) kernel.
- Random forest with 20 trees.

All model hyperparameters not specified were set to their default in the `scikit-learn` Python package. We also compare to a lasso knockoff model using coefficient difference statistic; we use the implementation provided in the `knockoffs` R package.

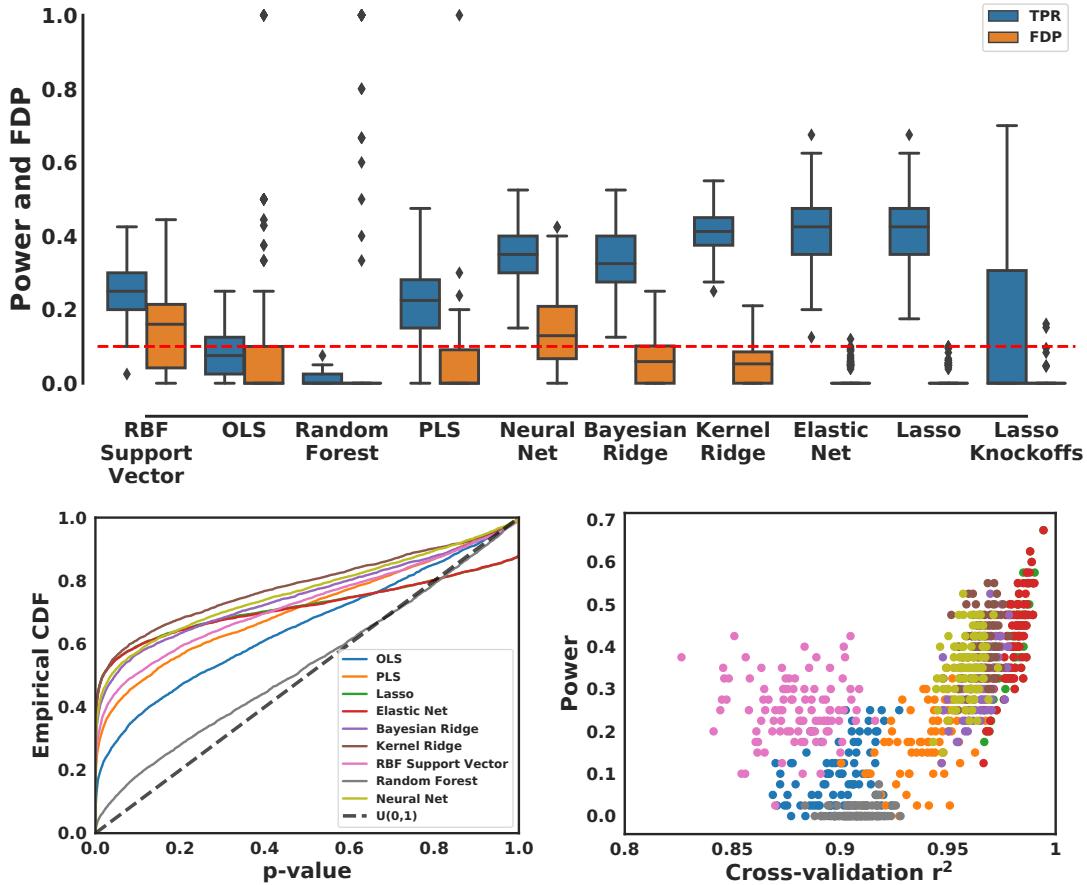


Figure 2: Top: Power and FDP for different choices of predictive model used in the CV-HRT, ordered by empirical risk of the predictive model. The right-most result is for a lasso model using model-X knockoffs for feature selection. Bottom left: distribution of p-values for signal features using each predictive model. Bottom right: predictive model performance versus feature selection power for each model and independent trial. Colors in the bottom right plot correspond to those in the bottom left.

Fig. 2 (top) shows the power and FDP results for each model. The predictive models are ordered by average cross-validation r^2 across all trials. CV-HRT power roughly increases with the quality of the predictive model, with the lasso model producing the highest power results. Fig. 2 (bottom right) demonstrates this with more granularity. It shows the power for each individual model on each trial as a function of the individual model r^2 . Even within the same model class, the models that generalize better tend to have higher power in the CV-HRT.

A notable exception is the random forest model, which has low power across all trials. This may be due to the nature of the model. Random forests average predictions from an ensemble of decision trees. Each tree predicts by recursively dividing into a series of half-planes, with each half-plane based on a single feature. Most null samples in the HRT

are near, but slightly different than, the original data. It is likely that a small change to the original data will not change the random forest prediction, since it may not cause the feature to cross any half-plane. Since the comparison of the test statistic under the original model with each null sample statistic is not strict in the HRT, the insensitivity of random forests to small data changes leads to lower power. One solution to this is to use more trees in the forest, with data subsampling or bootstrapping to produce many different half-planes for each feature.

Fig. 2 also presents the results of a lasso knockoffs model using the coefficient magnitude as a test statistic, as proposed by Candes et al. [2018]. The power of the CV-HRT lasso model in this simulation is approximately 3x more than knockoffs (45% versus 15% on average).

5 Case studies from the scientific literature

We demonstrate the usefulness of the HRT on two datasets from real experiments. The first dataset measures the genomic profile of hundreds of cancer cell lines and their response to an anti-cancer drug. The second measures the molecular structure of hundreds of chemicals and their perceived olfactory properties. In both experiments, the original scientific analyses followed the same template: (i) a large number of features were gathered about the target, (ii) a predictive model was fit, (iii) a model-specific feature importance heuristic was used to rank features, and (iv) top-ranked features were reported as discoveries.

In both case studies, we followed the same procedures for steps (i) and (ii), but then replaced (iii) and (iv) with the calibrated CV-HRT (Algorithm 4). We use the same conditional estimator as in Section 4.1, but for numerical stability take the first 100 principal components as inputs rather than raw features. We briefly describe each experiment and the original analysis approach, then present a comparison of the heuristics employed with the discoveries made by the HRT.

Drug response experiment The first dataset is a study of anti-cancer drug response in cancer cell lines [Barretina et al., 2012]. Multiple drugs were tested against hundreds of cell lines; phenotypic response was measured as the area under the dose-response curve (AUC). Each cell line was analyzed to obtain gene mutation and expression features. The scientific goal is to discover the genomic features associated with drug response. Genomic features were first screened to filter out features with less than 0.1 magnitude Pearson correlation to the AUC. An elastic net model was fit for each drug, with hyperparameters chosen via 10-fold cross-validation. Features were then ranked by average coefficient magnitude. We choose a single drug, PLX4720, as an illustrative example. The results presented are similar to the original publication results for PLX4720, though not identical. We followed the analysis to the best of our ability, but the publicly available data is newer than the dataset used in the original publication. Our results are therefore different than those published, but not meaningfully so.

Olfactory perception experiment The second dataset is a study of the perceived fragrant properties of various molecules [Keller et al., 2017]. Several hundred human subjects smelled and rated each molecule across 20 different categories. For each molecule, several thousand descriptive features about its chemical structure were measured. A random forest model was fit for each category to predict the average human rating given the features for a

(a) Elastic net for cancer drug response

Genomic Feature	Imp. score	Est. p-value
BRAF V600E Mut	0.0975	$\leq 10^{-5}*$
RP11-208G20.3	0.0715	0.0065*
RP6-149D17.1	0.0665	0.2108
RNU6-104P	0.0652	0.9970
RNA5SP184	0.0634	0.3359
VPS13B Mut	0.0557	0.0042*
RP11-567M16.3	0.0535	0.0079*
MTMR11 Mut	0.0525	0.0821
ZNF549	0.0524	0.1915
HIP1 Mut	0.0519	$\leq 10^{-5}*$

(b) Random forests for olfactory perception

Molecular Feature	Imp. score	Est. p-value
Isovanillin	0.2629	0.0012*
Vanillin isobutyrate	0.0528	0.2553
Ethyl vanillin	0.0481	0.0430
Ethyl vanillin acetate	0.0258	0.6066
Protocatechualdehyde	0.0236	0.1515
Vanillin acetate	0.0232	0.2845
2-Formyylimidazole	0.0186	0.8132
R7e+	0.0179	0.5806
Ethyl Isovalerate	0.0123	0.7837
SM05 AEA(ri)	0.0099	0.2751

Table 3: Two examples of predictive modeling with heuristic post-hoc feature importance ranking in scientific studies. a) Genomic features were used to predict cell line response to treatment with the drug PLX-4720 [Barretina et al., 2012]. b) Molecular features were used to predict perceived fragrant properties of molecules [Keller et al., 2017]. Asterisks indicate features selected by the HRT at a 20% FDR threshold.

molecule. Different types of molecular descriptors were tried, with two sets (Dragon and Morgan descriptors) being selected based on performance on a held out test set. Features were then ranked by the random forest feature importance heuristic in `scikit-learn`. This heuristic estimates the expected number of samples in which a feature is used, based on how often and how deep it appears in the constituent trees. We again choose a single illustrative example, the *Bakery* category.

To increase power, we use the heuristics as a filter on which features to test in both case studies. Features with less than 10^{-3} heuristic importance (coefficient magnitude in elastic net and expected usage in random forests) are ignored. This leaves 873 features for the drug response dataset and 93 features for the olfactory perception dataset. This filtering step does not affect the statistical guarantees of the HRT since it is not based on any of the held out test data. It only avoids testing features that are completely ignored by the model and which the HRT will have no power to detect, if they are signals.

Table 3 shows the top 10 ranked features in both datasets, following their respective heuristics. Beside each feature, we report its heuristic importance score and the p -value assigned by the HRT. The heuristic rankings correlate poorly with the (theoretically grounded) p -value assigned by the HRT. Features with an asterisk denote those selected by the HRT with Benjamini-Hochberg correction at a 20% FDR threshold. It is impossible to know whether these features are all true positives, but the HRT demonstrates that the statistical evidence from the model does not match with the ranking heuristics.

Additional discoveries by the HRT. In the drug response experiment, the HRT selected additional features. Table 4 shows all discoveries by the HRT at a 20% FDR threshold that were not ranked in the top 10 by the heuristic. Alongside each feature we show its p -value estimate, heuristic score, and heuristic ranking. Several features were found to have significant predictive power despite their estimated model effect size being relatively low. This suggests there may be new potential targets of therapy worthy of follow-up

Genomic Feature	Heuristic ranking	Coefficient magnitude	Est. <i>p</i> -value
CNTN1 Mut	11	0.0512	0.0059
RNU6-448P	12	0.0486	0.0000
MIR4482-1	14	0.0474	0.0004
MTND4P25	19	0.0457	0.0010
RP11-585F1.8	20	0.0451	0.0058
RN7SL528P	21	0.0433	0.0034
KRTAP23-1	25	0.0384	0.0006
GS1-24F4.3	29	0.0372	0.0000
FLT3 Mut	31	0.0363	0.0000
RNU6-1287P	32	0.0360	0.0000
RP11-575H3.1	44	0.0294	0.0035
SNAPC3	51	0.0268	0.0012
RP11-541E12.1	55	0.0251	0.0000
RP11-488I4.2	70	0.0225	0.0012
RNA5SP234	72	0.0224	0.0000
CDC42BPA Mut	76	0.0213	0.0000
NEURL	79	0.0205	0.0000
RP11-481A12.2	82	0.0203	0.0002
HCP5	86	0.0200	0.0011
CTC-539A10.7	91	0.0190	0.0062
PIK3R4 Mut	98	0.0183	0.0025
SLC35G6	115	0.0165	0.0008
PIP5K1A Mut	122	0.0162	0.0002
CCND2 Mut	171	0.0123	0.0026
CDC37L1	178	0.0121	0.0000
CSPG4 Mut	211	0.0094	0.0003
RP5-1195D24.1	213	0.0094	0.0000
DIP2C Mut	256	0.0081	0.0003
RIOK3 Mut	358	0.0055	0.0049
POLR2J4	401	0.0047	0.0031
CACNA2D2	409	0.0046	0.0060
RNA5SP280	420	0.0044	0.0020
NTSR1 Mut	672	0.0018	0.0002

Table 4: Additional genomic features selected by the HRT at a 20% false discovery rate threshold in the drug response study, but not in the top 10 ranking of coefficient magnitude.

investigation.

6 Discussion

When Leo Breiman wrote about the two cultures in statistics [Breiman, 2001], he divided statisticians into the *data modelers* and the *algorithmic modelers*. Data modelers relied on inflexible, idealized parametric models that make unrealistic assumptions about the true data generating distribution in order to gain the mathematical convenience of verification tools like goodness-of-fit tests. Algorithmic modelers chose to treat the latent function mapping features to response as a black box. In this latter paradigm, predictive performance on held out data was the only way to truly measure the strength of a model. Breiman estimated at the time that 98% of the mainstream analyst world fell into the data modeling crowd, while a mere 2% were algorithmic modelers.

Nearly 20 years later, the tables have turned. A confluence of factors (better numerical computing tools, the rise of Computer Science as one of the most popular majors in university, the dawn of the “Big Data” era, high throughput screening techniques in science, and the practical successes of machine learning) has led us to a world where algorithmic modeling is the norm and test error is the gold standard for model evaluation. The gravity of algorithmic modeling has grown so strong that it has even pulled in many scientific fields, leading to prediction challenges like the DREAM series [Stolovitzky et al., 2007]. These challenges publicly release scientific data and teams compete to build the best model based solely on performance on held out test data. One would be hard pressed to argue that the algorithmic modelers are in the minority today.

Yet as we have shown, this shift has come at the cost of reliably extracting understanding from the data, what Breiman called the *information* goal. Articles in premier scientific journals, such as the two case studies in Section 5, now commonly use machine learning methods to build predictive models then derive scientific conclusions from ad-hoc model interpretation. These heuristic approaches rest on unstable ground and may lead scientists astray by convincing them that a strong signal exists where there is only noise. Breiman himself nearly solved this by effectively proposing to use what we call the marginal permutation HRT (Section 4.1) as a means for interpreting random forests, but he did not consider the need to account for feature dependency structure in his test. It is fitting then that rigorously extracting information from black box models, as the HRT does, requires using *more* black box modeling (i.e. conditional density estimators) to disentangle the features and perform a reliable conditional independence test.

A key question left open by the HRT is how to estimate complete conditionals reliably. The heuristic proposal in Appendix A works in our experiments, but it is ad hoc and requires a heavy computation to generate bootstrapped confidence intervals. The same question is raised for other methods such as model-X knockoffs, which in practice all estimate the knockoff distribution from data without guarantees on the results. Future theoretical work is needed to understand how to estimate complete conditionals with minimal assumptions and statistical guarantees on the resulting CRT. Similarly, it may be for certain models and datasets to derive an optimal splitting scheme that maximizes power in a robust class of response functions. The theoretical foundations of how to allocate training and testing splits is also left for future work.

There are several additional directions to explore for extending the HRT to other analysis settings. In some models, such as empirical Bayes models that use predictive models for

prior estimation [Tansey et al., 2018, 2020a,b], inference is still too expensive to run the HRT. Efficiently searching over the space of potential features to test, using an approach like Bayesian optimization [Shahriari et al., 2016] or multi-armed bandit learning [Gandy and Hahn, 2017], would facilitate applying the HRT to compute-intensive models. In other scenarios, such as image classification, language modeling, and time series analysis, the features themselves have a more complicated structure and selecting an individual feature is not necessarily the inferential goal. For example, a pixel location being significant is not usually an interesting discovery in image classification. A conceptual layer needs to be added to extract meaningful insight in these types of models.

Finally, the HRT has the added benefit of allowing arbitrarily fine-grained questions to be asked. For instance, the scientist can partition the dataset based on some other feature, at which point the null hypothesis is $X_j \perp\!\!\!\perp Y | X_k = x, X_{-(j,k)}$. This is a common test in biology, where predictive models are often trained on all of the data but questions are then asked about feature dependencies for specific samples, such as a specific type of cancer or class of drug. We plan to explore these data-partitioning scenarios in future work.

Acknowledgments. The authors thank Yixin Wang, Daniel Hsu, Ioan Filip, Scott Linderman, Christopher Tosh, Jackson Loper, and Lucas Janson for helpful conversations in the early stages of this work.

References

- R. F. Barber and E. J. Candès. Controlling the false discovery rate via knockoffs. *The Annals of Statistics*, 43(5):2055–2085, 2015.
- J. Barretina, G. Caponigro, N. Stransky, K. Venkatesan, A. A. Margolin, S. Kim, C. J. Wilson, J. Lehár, G. V. Kryukov, and D. Sonkin. The cancer cell line encyclopedia enables predictive modelling of anticancer drug sensitivity. *Nature*, 483(7391):603, 2012.
- S. Basu, K. Kumbier, J. B. Brown, and B. Yu. Iterative random forests to discover predictive and stable high-order interactions. *Proceedings of the National Academy of Sciences*, 115(8):1943–1948, 2018.
- Y. Benjamini and Y. Hochberg. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 57(1):289–300, 1995.
- Y. Benjamini and D. Yekutieli. The control of the false discovery rate in multiple testing under dependency. *The Annals of Statistics*, 29(4):1165–1188, 2001.
- T. B. Berrett, Y. Wang, R. F. Barber, and R. J. Samworth. The conditional permutation test. *arXiv preprint arXiv:1807.05405*, 2018.

- C. M. Bishop. Mixture density networks. Technical report, Aston University, Birmingham, UK, 1994.
- L. Breiman. Statistical modeling: The two cultures. *Statistical Science*, 16(3):199–231, 2001.
- E. Candes, Y. Fan, L. Janson, and J. Lv. Panning for gold: ‘Model-X’ knockoffs for high dimensional controlled variable selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 80(3):551–577, 2018.
- J. Chen, L. Song, M. J. Wainwright, and M. I. Jordan. Learning to explain: An information-theoretic perspective on model interpretation. *International Conference on Machine Learning (ICML)*, 2018.
- L. Crawford, K. C. Wood, X. Zhou, and S. Mukherjee. Bayesian approximate kernel regression with variable selection. *Journal of the American Statistical Association*, 113(524):1–12, 2018.
- B. Efron and R. Tibshirani. The problem of regions. *The Annals of Statistics*, 26(5):1687–1718, 1998.
- A. Fisher, C. Rudin, and F. Dominici. Model class reliance: Variable importance measures for any machine learning model class, from the “Rashomon” perspective. *arXiv preprint arXiv:1801.01489*, 2018.
- A. Gandy and G. Hahn. QuickMMCTest: Quick multiple Monte Carlo testing. *Statistics and Computing*, 27(3):823–832, 2017.
- J. R. Gimenez, A. Ghorbani, and J. Zou. Knockoffs for the mass: New feature importance statistics with false discovery guarantees. *arXiv preprint arXiv:1807.06214*, 2018.
- I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. *Advances in Neural Information Processing Systems (NIPS)*, 2014.
- J. Jordon, J. Yoon, and M. van der Schaar. KnockoffGAN: Generating knockoffs for feature selection using generative adversarial networks. In *International Conference on Learning Representations*, 2019.
- A. Keller, R. C. Gerkin, Y. Guan, A. Dhurandhar, G. Turu, B. Szalai, J. D. Mainland, Y. Ihara, C. W. Yu, R. Wolfinger, C. Vens, L. Schietgat, K. De Grave, R. Norel, , G. Stolovitzky, G. A. Cecchi, L. B. Vosshall, and P. Meyer. Predicting human olfactory perception from chemical features of odor molecules. *Science*, 355(6327):820–826, 2017.

- J. Lei, M. G’Sell, A. Rinaldo, R. J. Tibshirani, and L. Wasserman. Distribution-free predictive inference for regression. *Journal of the American Statistical Association*, 113(523):1094–1111, 2018.
- F. Liang, Q. Li, and L. Zhou. Bayesian neural networks for selection of drug sensitive genes. *Journal of the American Statistical Association*, pages 1–18, 2018.
- Y. Y. Lu, J. Lv, Y. Fan, and W. S. Noble. DeepPINK: Reproducible feature selection in deep neural networks. *arXiv preprint arXiv:1809.01185*, 2018.
- S. M. Lundberg and S.-I. Lee. A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems (NIPS)*, 2017.
- N. Meinshausen, L. Meier, and P. Bühlmann. P-values for high-dimensional regression. *Journal of the American Statistical Association*, 104(488):1671–1681, 2009.
- J. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in Large Margin Classifiers*, 10(3):61–74, 1999.
- M. T. Ribeiro, S. Singh, and C. Guestrin. Why should I trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.
- Y. Romano, M. Sesia, and E. J. Candès. Deep knockoffs. *Journal of the American Statistical Association*, 2019.
- R. Sen, K. Shanmugam, H. Asnani, A. Rahimzamani, and S. Kannan. Mimic and classify: A meta-algorithm for conditional independence testing. *arXiv preprint arXiv:1806.09708*, 2018.
- M. Sesia, C. Sabatti, and E. J. Candès. Gene hunting with knockoffs for hidden markov models. *arXiv preprint arXiv:1706.04677*, 2017.
- R. D. Shah and J. Peters. The hardness of conditional independence testing and the generalised covariance measure. *arXiv preprint arXiv:1804.07203*, 2018.
- B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. De Freitas. Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2016.
- A. Shrikumar, P. Greenside, and A. Kundaje. Learning important features through propagating activation differences. *International Conference on Machine Learning (ICML)*, 2017.

- G. Stolovitzky, D. Monroe, and A. Califano. Dialogue on reverse-engineering assessment and methods. *Annals of the New York Academy of Sciences*, 1115(1):1–22, 2007.
- W. Tansey, Y. Wang, D. M. Blei, and R. Rabadian. Black box FDR. In *International Conference on Machine Learning*, pages 4874–4883, 2018.
- W. Tansey, K. Li, H. Zhang, S. W. Linderman, R. Rabadian, D. M. Blei, and C. H. Wiggins. Dose-response modeling in high-throughput cancer drug screenings: An end-to-end approach. *Biostatistics*, 2020a.
- W. Tansey, Y. Wang, R. Rabadian, and D. M. Blei. Double empirical bayes testing. *International Statistical Review*, 2020b.
- V. Vovk. Conditional validity of inductive conformal predictors. In *Asian Conference on Machine Learning*, pages 475–490, 2012.
- L. Wasserman and K. Roeder. High dimensional variable selection. *Annals of statistics*, 37(5A):2178, 2009.

A Conservative estimation by bootstrap sample reweighting

The error in the estimated complete conditional, $Q_{j|-j}$, can lead to inflated tails of the null p -values, causing a violation of the Type I or FDR error threshold. Here we develop a calibration technique that aims to produce conservative p -values by pessimistically weighting each null sample comparison.

We begin by casting Algorithm 2 as an importance sampling scheme with $Q_{j|-j}$ as a proposal distribution,

$$\hat{p}_j \approx \frac{1}{1 + \sum_{k=1}^K W^{(k)}} \left(1 + \sum_{k=1}^K \mathbb{1} \left\{ \mathcal{G}(\mathcal{D}', \pi_{\hat{\theta}}) \geq \mathcal{G}(\tilde{\mathcal{D}}^{(k)}, \pi_{\hat{\theta}}) \right\} W^{(k)} \right), \quad (6)$$

$$\tilde{\mathcal{D}}^{(k)} = [\tilde{\mathbf{X}}_{\cdot j}^{(k)}, \mathbf{X}'_{\cdot -j}], \quad \tilde{\mathbf{X}}_{\cdot j}^{(k)} \sim Q_{j|-j},$$

where $W^{(k)}$ is the importance weight,

$$W^{(k)} = \frac{P(X_j | X_{-j})(\tilde{\mathbf{X}}_{\cdot j}^{(k)})}{Q_{j|-j}(\tilde{\mathbf{X}}_{\cdot j}^{(k)})}, \quad (7)$$

where $P(X_j | X_{-j})(\tilde{\mathbf{X}}_{\cdot j}^{(k)}) = \prod_{i=1}^{n'} P(X_j | X_{-j})(\tilde{\mathbf{X}}_{ij}^{(k)})$, and analogously for $Q_{j|-j}(\tilde{\mathbf{X}}_{\cdot j}^{(k)})$. The estimate in (6) is consistent, but it still relies on the unknown true conditional $P(X_j | X_{-j})$, making it impossible to compute in practice.

Let $f_{j|-j}$ and $h_{j|-j}$ be functions that bound the true probability of the null sample: $0 \leq f_{j|-j}(\tilde{\mathbf{X}}_{\cdot j}) \leq P(X_j | X_{-j})(\tilde{\mathbf{X}}_{\cdot j}) \leq h_{j|-j}(\tilde{\mathbf{X}}_{\cdot j})$. If we have access to such a function, we can then bound p_j in expectation by biasing the importance weights to be conservative. This works by choosing the importance weight dynamically based on the outcome of each test statistic comparison,

$$\widehat{W}^{(k)} = \begin{cases} \frac{f_{j|-j}(\tilde{\mathbf{X}}_{\cdot j}^{(k)})}{Q_{j|-j}(\tilde{\mathbf{X}}_{\cdot j}^{(k)})} & \text{if } \mathcal{G}(\mathcal{D}', \pi_{\hat{\theta}}) < \mathcal{G}(\tilde{\mathcal{D}}^{(k)}, \pi_{\hat{\theta}}) \\ \frac{h_{j|-j}(\tilde{\mathbf{X}}_{\cdot j}^{(k)})}{Q_{j|-j}(\tilde{\mathbf{X}}_{\cdot j}^{(k)})} & \text{otherwise.} \end{cases} \quad (8)$$

Under this conservative weighting, $\mathbb{E}[\hat{p}_j] \geq p_j$. The tighter the bound from $(f_{j|-j}, h_{j|-j})$, the less conservative the p -value estimate and thus the more powerful the test statistic.

Trivially, for any true $P(X_j | X_{-j})$, we can always choose $f_{j|j} = 0$ and guarantee \hat{p} will be conservative. It will always be 1, however, so the method will be without power. To be effective, the bound must be relatively tight.

To generate approximate bounds on $P(X_j | X_{-j})$, we treat the CDE model as a black box and use the bootstrap to estimate quantiles [see Efron and Tibshirani, 1998, for a discussion on the technical issues with using the bootstrap for confidence intervals]. We create b bootstrap resamples of the dataset and fit b corresponding CDE models $B_{j|j} = (Q_{j|j}^{(1)}, Q_{j|j}^{(2)}, \dots, Q_{j|j}^{(b)})$, with $Q_{j|j}^{(1)}$ used as the proposal distribution. Given a choice of lower and upper quantile, (l, u) , we can replace the reweighting scheme in (8) with an approximate one,

$$\widehat{W}^{(k)} = \begin{cases} \frac{B_{j|j}^{(l)}(\tilde{\mathbf{X}}_{\cdot j}^{(k)})}{Q_{j|j}^{(1)}(\tilde{\mathbf{X}}_{\cdot j}^{(k)})} & \text{if } \mathcal{G}(\mathcal{D}', \pi_{\hat{\theta}}) < \mathcal{G}(\tilde{\mathcal{D}}^{(k)}, \pi_{\hat{\theta}}) \\ \frac{B_{j|j}^{(u)}(\tilde{\mathbf{X}}_{\cdot j}^{(k)})}{Q_{j|j}^{(1)}(\tilde{\mathbf{X}}_{\cdot j}^{(k)})} & \text{otherwise.} \end{cases} \quad (9)$$

where $B_{j|j}^{(i)}$ denotes the i^{th} quantile of the estimates in $B_{j|j}$.

We find that taking the ratio of the products as in Eq. (9) is numerically unstable for large n' and quickly goes to 0 or ∞ (numerically). To stabilize the procedure, we take the geometric mean of the ratio of constituent-to-proposal conditional probability estimate \mathbb{G} for each model across all n' samples,

$$\widehat{W}^{(k)} = \begin{cases} \left(\prod_{i=1}^{n'} \frac{B_{j|j}^{(l)}(\tilde{\mathbf{X}}_{ij}^{(k)})}{Q_{j|j}^{(1)}(\tilde{\mathbf{X}}_{ij}^{(k)})} \right)^{1/n'} & \text{if } \mathcal{G}(\mathcal{D}', \pi_{\hat{\theta}}) < \mathcal{G}(\tilde{\mathcal{D}}^{(k)}, \pi_{\hat{\theta}}) \\ \left(\prod_{i=1}^{n'} \frac{B_{j|j}^{(u)}(\tilde{\mathbf{X}}_{ij}^{(k)})}{Q_{j|j}^{(1)}(\tilde{\mathbf{X}}_{ij}^{(k)})} \right)^{1/n'} & \text{otherwise.} \end{cases} \quad (10)$$

Bootstrapping solves the first half of the calibration dilemma by providing a model-agnostic, nonparametric way to generate arbitrary quantile estimates. The other requirement is a way to choose the upper and lower quantiles. For this, we leverage the IID property of the conditional CDF values for each sample.

If a sampling distribution $Q_{j|j}$ is well-calibrated, the distribution of CDF values for all samples should be uniform. In practice, this does not happen when using highly-flexible

black box conditional density estimators trained via maximum likelihood estimation. While many approaches to recalibration have been explored in the literature, most tend to apply to classification or regression models (e.g. Platt scaling [Platt, 1999]). It is not immediately clear how one can calibrate quantiles for black box conditional density estimates. We use a data-adaptive heuristic described in Appendix B for continuous random variables; for other data types, we recommend simply using $(l, u) = (5, 95)$ as a conservative choice.

B Approximate calibration via a one-way KS statistic

For continuous random variables, we propose a one-way Kolmogorov-Smirnov (KS) statistic to choose the lower and upper quantiles of the bootstrapped models. An adaptation to discrete random variables is also available in our implementation. For other types of features, we recommend a conservative bound of $(l, u) = (5, 95)$; we find this is sufficient to control FDR in preliminary experiments.

We describe the approach in terms of finding a lower bound; the upper bound is determined analogously. To choose a lower quantile, we start with the median ($l = 50$) and iteratively lower it until the CDF estimates dominate the uniform CDF from above in a Q-Q plot. Let $C_{j|j}$ be the CDF for $B_{j|j}$. The one-way lower-KS statistic for l is then,

$$\text{KS}^+(l) = \underset{c \in [0,1]}{\text{maximum}} \left(0, c - \frac{1}{n} \sum_{i=1}^n \mathbb{I} \left[C_{j|j}^{(l)}(X_{ij}) \leq c \right] \right). \quad (11)$$

We then check if the current choice of l yields a sufficiently-small test statistic. Specifically, we conduct a hypothesis test by Monte Carlo, where we repeatedly draw samples from $U(0, 1)$ and only accept l if it is smaller than 10^5 draws from a true $U(0, 1)$. This corresponds to solving the following optimization problem:

$$\begin{aligned} & \underset{l}{\operatorname{argmax}} && \max(0, \text{KS}^+(l)) \\ & \text{subject to} && \text{KS}^+(l) < \tau^+, \end{aligned} \quad (12)$$

where τ^+ is the MC-derived threshold; the upper quantile is found analogously. While the calibration procedure may seem very conservative, in practice it yields reasonable bounds even for the high-dimensional datasets in Section 4.

C Conservativism of Bootstrap Heuristic

With $\hat{W}^{(k)}$ as defined in Eq. (8), relatively little is required for conservativism of the estimator of p_j (the true p -value),

$$\begin{aligned}\hat{p}_j^K &\approx \frac{1}{1 + \sum_{k=1}^K \hat{W}^{(k)}} \left(1 + \sum_{k=1}^K \mathbb{1} \left\{ \mathcal{G}(\mathcal{D}', \pi_{\hat{\theta}}) \geq \mathcal{G}(\tilde{\mathcal{D}}^{(k)}, \pi_{\hat{\theta}}) \right\} \hat{W}^{(k)} \right), \\ \tilde{\mathcal{D}}^{(k)} &= [\tilde{\mathbf{X}}_{\cdot j}^{(k)}, X'_{\cdot -j}], \quad \tilde{\mathbf{X}}_{\cdot j}^{(k)} \sim Q_{j| \cdot j}.\end{aligned}$$

In particular, the functions $f_{j| \cdot j}$ and $h_{j| \cdot j}$ only need to bound $P(X_j | X_{-j})$ in a very crude sense. Let $E_j = \left\{ \mathcal{G}(\mathcal{D}', \pi_{\hat{\theta}}) \geq \mathcal{G}(\tilde{\mathcal{D}}^{(k)}, \pi_{\hat{\theta}}) \right\}$ be the event that the empirical risk of a null sampler is less than the observed empirical risk. Then,

Theorem 2. *Let $f_{j| \cdot j}$ and $h_{j| \cdot j}$ be such that $0 \leq \mathbb{E}_{P(X_j | X_{-j})} \left[\frac{f_{j| \cdot j}(\tilde{\mathbf{X}}_{\cdot j}^{(k)})}{P(X_j | X_{-j})(\tilde{\mathbf{X}}_{\cdot j}^{(k)})} (1 - \mathbb{1}\{E_j\}) \right] \leq 1 - p_j$ and $p_j \leq \mathbb{E}_{P(X_j | X_{-j})} \left[\frac{h_{j| \cdot j}(\tilde{\mathbf{X}}_{\cdot j}^{(k)})}{P(X_j | X_{-j})(\tilde{\mathbf{X}}_{\cdot j}^{(k)})} \mathbb{1}\{E_j\} \right] < \infty$. Then, with importance weights as in Eq. (8), $\lim_{K \rightarrow \infty} \hat{p}_j^K \geq p_j$ almost surely (with respect to the proposal distribution $Q_{j| \cdot j}$).*

Proof. By the law of large numbers, almost surely,

$$\lim_{K \rightarrow \infty} \hat{p}_j^K = \frac{\mathbb{E}_{Q_{j| \cdot j}}[\hat{W}^{(k)} \mathbb{1}\{E_j\}]}{\mathbb{E}_{Q_{j| \cdot j}}[\hat{W}^{(k)}]} \tag{13}$$

$$= 1 / (1 + \frac{\mathbb{E}_{Q_{j| \cdot j}}[\hat{W}^{(k)}(1 - \mathbb{1}\{E_j\})]}{\mathbb{E}_{Q_{j| \cdot j}}[\hat{W}^{(k)} \mathbb{1}\{E_j\}]}), \tag{14}$$

where the second line is an algebraic manipulation. Notice that

$$\begin{aligned}\mathbb{E}_{Q_{j| \cdot j}}[\hat{W}^{(k)} \mathbb{1}\{E_j\}] &= \mathbb{E}_{Q_{j| \cdot j}} \left[\frac{h_{j| \cdot j}(\tilde{\mathbf{X}}_{\cdot j}^{(k)})}{Q_{j| \cdot j}(\tilde{\mathbf{X}}_{\cdot j}^{(k)})} \mathbb{1}\{E_j\} \right] \\ &= \mathbb{E}_{P(X_j | X_{-j})} \left[\frac{h_{j| \cdot j}(\tilde{\mathbf{X}}_{\cdot j}^{(k)})}{P(X_j | X_{-j})(\tilde{\mathbf{X}}_{\cdot j}^{(k)})} \mathbb{1}\{E_j\} \right] \\ &\geq p_j,\end{aligned}$$

where the last line is by assumption. Similarly, $\mathbb{E}_{Q_{j| \cdot j}}[\hat{W}^{(k)}(1 - \mathbb{1}\{E_j\})] \leq 1 - p_j$. The result then follows by comparing $p_j = 1 / (1 + \frac{1-p_j}{p_j})$ and Eq. (14). \square

Theorem 2 shows that, because the test statistic is fixed, we require only that the

upper and lower bounds are valid on average over the events E_j and $1 - E_j$. Note that our bootstrapping heuristic does not guarantee that Theorem 2 will hold. Rather, it states that if the heuristic succeeds in obtaining bounds (on average) then the p -values will be conservative high probability.

Model	Splits	$n = 20$	$n = 50$	$n = 100$	$n = 200$	$n = 500$	$n = 1000$
HRT (Algorithm 2)	1	0.05	0.03	0.05	0.04	0.06	0.06
	2	0.03	0.05	0.03	0.06	0.07	0.05
	3	0.04	0.03	0.06	0.05	0.05	0.05
CV-HRT (Algorithm 3)	4	0.08	0.06	0.03	0.05	0.03	0.03
	5	0.01	0.05	0.06	0.06	0.03	0.05
	10	0.02	0.06	0.04	0.07	0.05	0.06
	20	0	0	0.04	0.06	0.05	0.07
CV-HRT (Algorithm 4)	2	0.05	0.06	0.06	0.06	0.06	0.06
	3	0.04	0.06	0.05	0.08	0.07	0.06
	4	0.07	0.06	0.05	0.04	0.04	0.07
	5	0.07	0.06	0.05	0.05	0.06	0.07
	10	0.07	0.03	0.09	0.05	0.04	0.03
	20	0.05	0.04	0.05	0.04	0.06	0.06

Table 5: False discovery rate for the random forest predictive model in Section 4.1.

D Additional benchmark results for known conditionals

This section details additional results for the HRT algorithm and its extensions on the benchmarks in Section 4.1. First, we provide the corresponding false discovery rates for Tables 1 and 2. The results show that all experimental settings controlled the false discovery rate at or below the nominal 10% level.

We next provide the results of using different predictive models: ordinary least squares (Tables 7 and 10), lasso (Tables 8 and 11), and Bayesian ridge regression (Tables 9 and 12). These mirror the results using random forests presented in the main text.

Table 13 provides the corresponding result for the lasso and OLS models with different signal strengths for the nonnull features.

Finally, Figs. 3 to 7 present the Q-Q plots for p -values generated by the HRT and the four proposed extension algorithms. For the two cross-validation algorithms, we select $M = 5$ folds as an illustrative case. All results show that the methods generate empirically uniform or conservative null p -values (dotted lines in the figures). Further, the nonnull p -value distributions (solid lines) shift progressively to the left as the sample size grows.

Model	$n = 20$	$n = 50$	$n = 100$	$n = 200$	$n = 500$	$n = 1000$
FDR	HRT (Algorithm 2)	0.07	0.04	0.05	0.04	0.03
	HPT (Algorithm 5)	0.06	0.07	0.05	0.07	0.04
	HGT (Algorithm 6)	0.06	0.03	0.05	0.05	0.03

Table 6: False discovery rate for the HRT, HPT, and HGT using random forests as the predictive model.

Model	Splits	$n = 20$	$n = 50$	$n = 100$	$n = 200$	$n = 500$	$n = 1000$
HRT (Algorithm 2)	1	0.03	0.31	0.59	0.7	0.71	0.75
	2	0.04	0.36	0.67	0.69	0.74	0.77
	3	0.08	0.3	0.64	0.7	0.69	0.75
	4	0.06	0.33	0.63	0.7	0.72	0.72
	5	0.05	0.28	0.51	0.68	0.73	0.73
	10	0.04	0.15	0.35	0.56	0.71	0.71
	20	0.02	0.08	0.15	0.42	0.68	0.71
CV-HRT (Algorithm 3)	2	0.04	0.43	0.68	0.72	0.74	0.75
	3	0.09	0.48	0.67	0.69	0.71	0.75
	4	0.15	0.53	0.68	0.7	0.72	0.76
	5	0.17	0.54	0.68	0.69	0.74	0.79
	10	0.14	0.52	0.66	0.69	0.72	0.78
	20	0.12	0.54	0.67	0.68	0.73	0.78

Table 7: True positive rate for the ordinary least squares predictive model in Section 4.1.

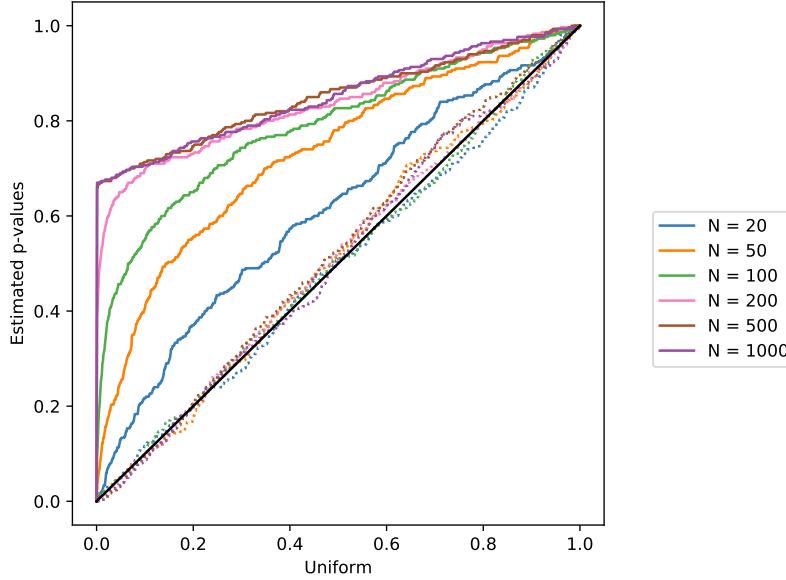


Figure 3: Q-Q plot for the HRT (Algorithm 1) simulations in Section 4.1. Solid lines show the CDFs of nonnull p -values; dotted lines show the CDFs of null p -values.

Model	Splits	$n = 20$	$n = 50$	$n = 100$	$n = 200$	$n = 500$	$n = 1000$
HRT (Algorithm 2)	1	0.08	0.28	0.61	0.69	0.7	0.72
	2	0.06	0.35	0.65	0.69	0.71	0.73
	3	0.09	0.36	0.62	0.68	0.72	0.72
CV-HRT (Algorithm 3)	4	0.04	0.24	0.58	0.66	0.69	0.69
	5	0.06	0.24	0.51	0.67	0.69	0.73
	10	0.01	0.15	0.32	0.62	0.68	0.71
	20	0.04	0.04	0.17	0.36	0.67	0.69
CV-HRT (Algorithm 4)	2	0.1	0.42	0.67	0.7	0.72	0.73
	3	0.12	0.48	0.66	0.7	0.71	0.73
	4	0.09	0.46	0.65	0.7	0.7	0.71
	5	0.13	0.47	0.66	0.69	0.71	0.72
	10	0.17	0.49	0.67	0.68	0.71	0.73
	20	0.19	0.51	0.68	0.69	0.71	0.75

Table 8: True positive rate for the lasso predictive model in Section 4.1.

Model	Splits	$n = 20$	$n = 50$	$n = 100$	$n = 200$	$n = 500$	$n = 1000$
HRT (Algorithm 2)	1	0.08	0.35	0.6	0.7	0.69	0.72
	2	0.09	0.43	0.68	0.69	0.7	0.72
	3	0.1	0.34	0.63	0.69	0.68	0.73
CV-HRT (Algorithm 3)	4	0.08	0.35	0.62	0.68	0.72	0.7
	5	0.08	0.31	0.53	0.67	0.71	0.72
	10	0.04	0.16	0.33	0.57	0.69	0.7
	20	0.02	0.06	0.15	0.44	0.68	0.7
CV-HRT (Algorithm 4)	2	0.12	0.47	0.67	0.7	0.7	0.73
	3	0.13	0.5	0.67	0.68	0.7	0.73
	4	0.18	0.52	0.68	0.69	0.7	0.74
	5	0.14	0.55	0.67	0.69	0.73	0.76
	10	0.15	0.55	0.67	0.68	0.71	0.76
	20	0.17	0.53	0.68	0.68	0.72	0.76

Table 9: True positive rate for the Bayesian ridge predictive model in Section 4.1.

Model	Splits	$n = 20$	$n = 50$	$n = 100$	$n = 200$	$n = 500$	$n = 1000$
HRT (Algorithm 2)	1	0.06	0.05	0.05	0.06	0.06	0.06
	2	0.05	0.06	0.05	0.06	0.03	0.06
	3	0.04	0.05	0.02	0.06	0.04	0.06
CV-HRT (Algorithm 3)	4	0.06	0.05	0.05	0.04	0.05	0.04
	5	0.05	0.02	0.05	0.04	0.06	0.04
	10	0.04	0.05	0.08	0.06	0.05	0.07
	20	0.04	0.04	0.03	0.06	0.05	0.04
CV-HRT (Algorithm 4)	2	0.05	0.05	0.04	0.05	0.06	0.05
	3	0.06	0.05	0.04	0.05	0.04	0.05
	4	0.08	0.06	0.05	0.05	0.04	0.03
	5	0.07	0.03	0.03	0.03	0.04	0.04
	10	0.07	0.03	0.05	0.04	0.05	0.04
	20	0.05	0.05	0.04	0.04	0.03	0.04

Table 10: False discovery rate for the ordinary least squares predictive model in Section 4.1.

Model	Splits	$n = 20$	$n = 50$	$n = 100$	$n = 200$	$n = 500$	$n = 1000$
HRT (Algorithm 2)	1	0	0.01	0.02	0.02	0.01	0.02
	2	0	0.02	0.02	0.03	0.01	0.02
	3	0.02	0	0.02	0.01	0.01	0.03
CV-HRT (Algorithm 3)	4	0.02	0	0.01	0.01	0.01	0.01
	5	0.03	0.02	0.02	0.01	0.02	0.02
	10	0.02	0.02	0.02	0.02	0.01	0.04
	20	0	0.02	0.02	0.02	0.02	0.03
CV-HRT (Algorithm 4)	2	0.02	0.02	0.03	0.03	0.01	0.02
	3	0.01	0.03	0.02	0.02	0.02	0.04
	4	0.04	0.03	0.03	0.01	0.01	0.02
	5	0.02	0.02	0.01	0.03	0.04	0.03
	10	0.03	0.06	0.02	0.02	0.02	0.04
	20	0.02	0.03	0.02	0.02	0.03	0.02

Table 11: False discovery rate for the lasso predictive model in Section 4.1.

Model	Splits	$n = 20$	$n = 50$	$n = 100$	$n = 200$	$n = 500$	$n = 1000$
HRT (Algorithm 2)	1	0.06	0.04	0.04	0.03	0.03	0.02
	2	0.04	0.02	0.05	0.04	0.01	0.02
	3	0.06	0.05	0.01	0.04	0.01	0.02
CV-HRT (Algorithm 3)	4	0.04	0.04	0.04	0.03	0.02	0.01
	5	0.01	0.03	0.04	0.03	0.02	0.02
	10	0.09	0.03	0.03	0.04	0.03	0.02
	20	0.03	0.02	0.02	0.04	0.02	0.01
	2	0.05	0.03	0.02	0.04	0.03	0.03
	3	0.04	0.02	0.02	0.03	0.03	0.03
CV-HRT (Algorithm 4)	4	0.06	0.03	0.03	0.03	0.02	0.02
	5	0.06	0.01	0.02	0.01	0.03	0.02
	10	0.02	0.01	0.03	0.01	0.02	0.02
	20	0.05	0.02	0.02	0.02	0.02	0.02

Table 12: False discovery rate for the Bayesian ridge predictive model in Section 4.1.

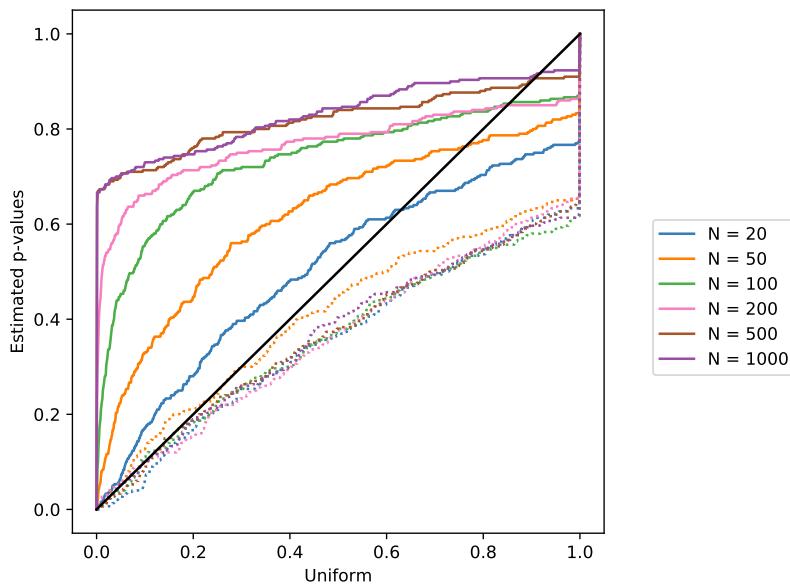


Figure 4: Q-Q plot for the CV-HRT (Algorithm 2) simulations in Section 4.1. Solid lines show the CDFs of nonnull p -values; dotted lines show the CDFs of null p -values.

Signal Strength	Algorithm	TPR		FDR	
		OLS	Lasso	OLS	Lasso
0.1	HRT (Algorithm 2)	0.55	0.58	0.03	0.01
	CV-HRT (Algorithm 3)	0.69	0.66	0.05	0.03
	CV-HRT (Algorithm 4)	0.69	0.68	0.04	0.04
0.2	HRT (Algorithm 2)	0.70	0.69	0.02	0.01
	CV-HRT (Algorithm 3)	0.73	0.69	0.03	0.02
	CV-HRT (Algorithm 4)	0.72	0.73	0.05	0.02
0.3	HRT (Algorithm 2)	0.71	0.71	0.03	0.01
	CV-HRT (Algorithm 3)	0.76	0.72	0.04	0.02
	CV-HRT (Algorithm 4)	0.76	0.76	0.05	0.02
0.4	HRT (Algorithm 2)	0.71	0.72	0.03	0.01
	CV-HRT (Algorithm 3)	0.75	0.72	0.04	0.02
	CV-HRT (Algorithm 4)	0.80	0.80	0.06	0.02
0.5	HRT (Algorithm 2)	0.72	0.72	0.04	0.01
	CV-HRT (Algorithm 3)	0.76	0.75	0.02	0.02
	CV-HRT (Algorithm 4)	0.83	0.82	0.05	0.02
0.6	HRT (Algorithm 2)	0.75	0.72	0.05	0.02
	CV-HRT (Algorithm 3)	0.77	0.77	0.03	0.01
	CV-HRT (Algorithm 4)	0.85	0.85	0.05	0.02
0.7	HRT (Algorithm 2)	0.74	0.75	0.06	0.03
	CV-HRT (Algorithm 3)	0.77	0.78	0.03	0.02
	CV-HRT (Algorithm 4)	0.87	0.88	0.04	0.03
0.8	HRT (Algorithm 2)	0.76	0.75	0.06	0.03
	CV-HRT (Algorithm 3)	0.79	0.79	0.03	0.02
	CV-HRT (Algorithm 4)	0.90	0.89	0.03	0.04
0.9	HRT (Algorithm 2)	0.77	0.77	0.06	0.03
	CV-HRT (Algorithm 3)	0.83	0.80	0.03	0.02
	CV-HRT (Algorithm 4)	0.93	0.90	0.03	0.03
0.9	HRT (Algorithm 2)	0.78	0.78	0.06	0.03
	CV-HRT (Algorithm 3)	0.83	0.82	0.03	0.02
	CV-HRT (Algorithm 4)	0.94	0.92	0.03	0.03

Table 13: OLS and lasso results as the signal strength of the features changes. All simulations used $n = 200$ data points and $M = 5$ folds for the cross-validation methods.

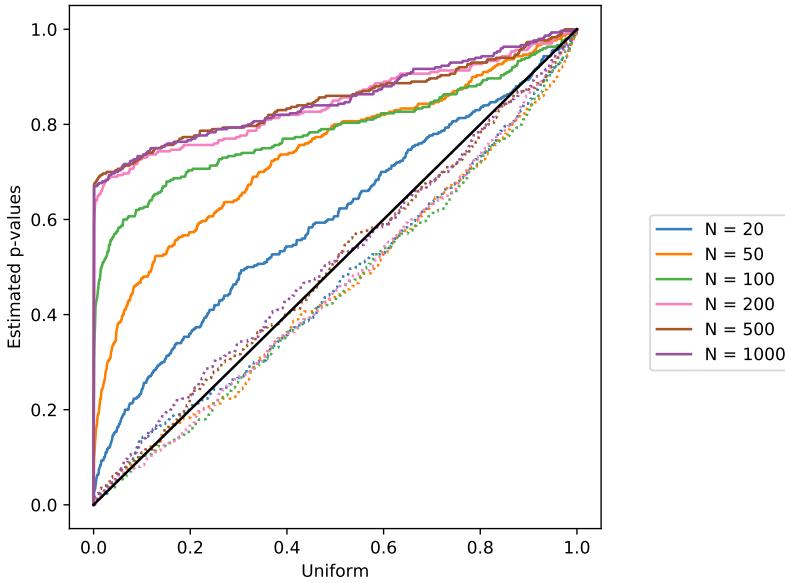


Figure 5: Q-Q plot for the approximate CV-HRT (Algorithm 3) simulations in Section 4.1. Solid lines show the CDFs of nonnull p -values; dotted lines show the CDFs of null p -values.

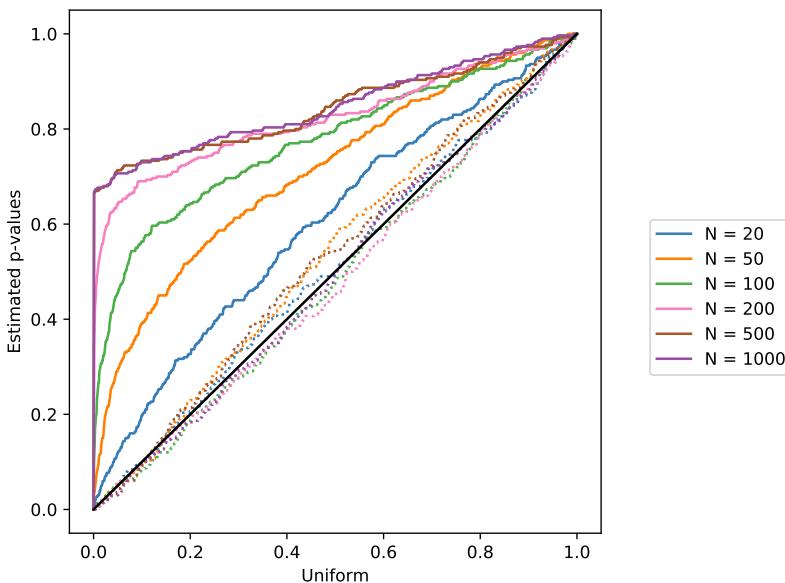


Figure 6: Q-Q plot for the HPT (Algorithm 4) simulations in Section 4.1. Solid lines show the CDFs of nonnull p -values; dotted lines show the CDFs of null p -values.

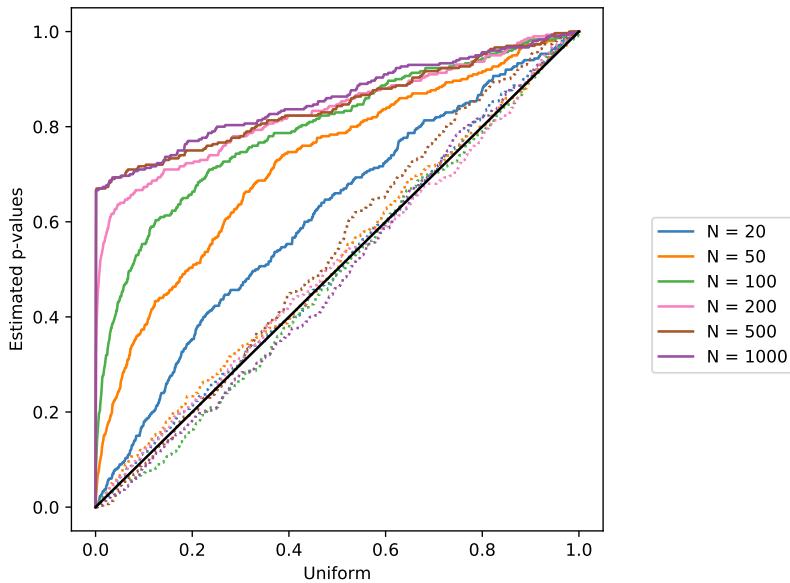


Figure 7: Q-Q plot for the HGT (Algorithm 5) simulations in Section 4.1. Solid lines show the CDFs of nonnull p -values; dotted lines show the CDFs of null p -values.

E Additional benchmark results for unknown conditionals

Fig. 8 provides a closer look at the calibration procedure results for the cross-validation HRT. The top two panels show a sweep of possible choices of the upper and lower bounds for both power and empirical FDR. Even at a stringent 90% confidence interval (top right corner of both heat maps), power is still 28%; empirical FDR declines relatively rapidly, being only $\approx 5\%$ at a 90% interval. The bottom left panel shows the CDF values of the p -values for the null features using different bounds. The panel is zoomed in to show the $[0, 0.1]$ region where p -values may be rejected under the target FDR threshold. Without any calibration, even the bootstrapped median (i.e. the $[50, 50]$ interval) still shows an inflation in the lower portion of the CDF, leading to invalid null p -values. A 20–30% confidence interval biases the null p -values down sufficiently to control the empirical FDR near the target level. The bottom right panel shows the distribution of bounds chosen by the data-adaptive procedure in each of the 100 independent trials; the procedure chooses roughly a 25% confidence region for most trials.

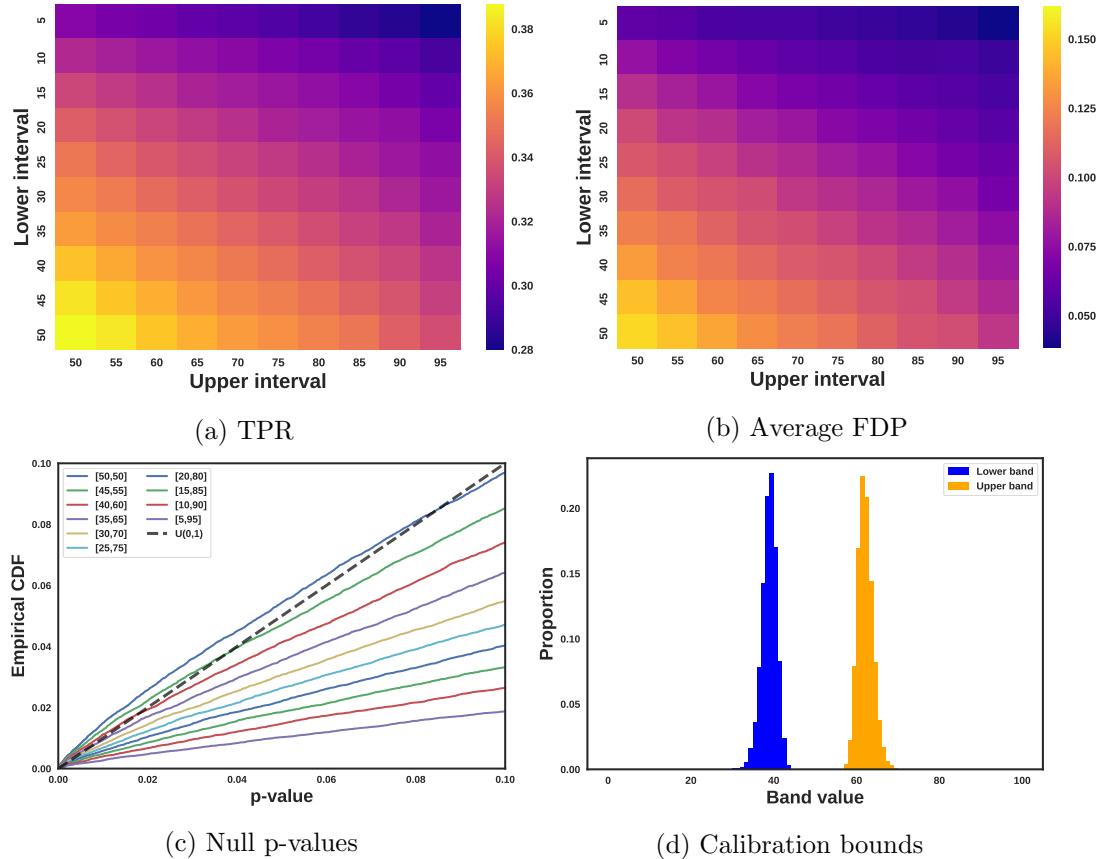


Figure 8: (a) TPR and (b) empirical FDR for different fixed upper and lower quantile intervals on the benchmarks. (c) CDF of the true null p -values across all benchmark trials for different interval selections. (d) distribution of bounds selected by the calibration technique.

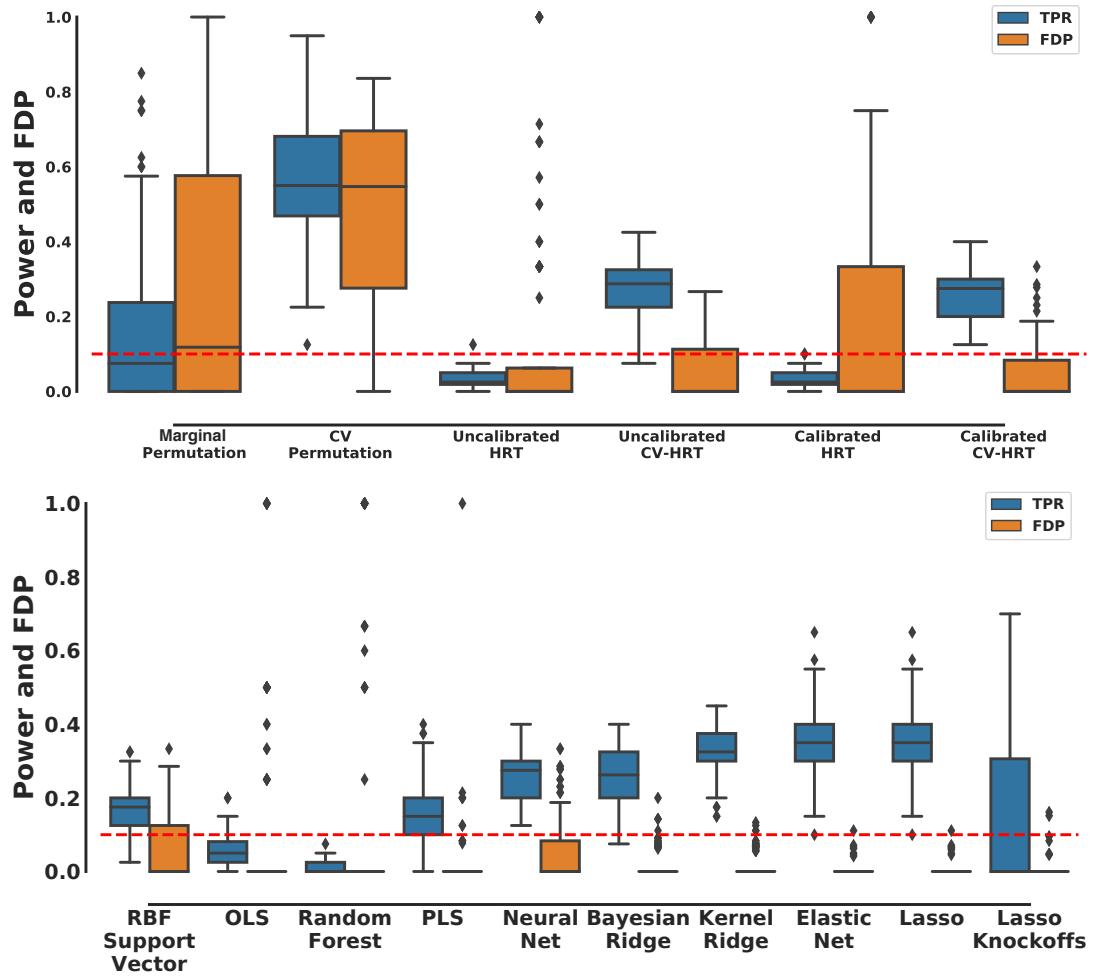


Figure 9: Top: Power and FDP results for each HRT variant on the benchmark simulation. Bottom: Power and FDP for different choices of predictive model used in the CV-HRT, ordered by empirical risk of the predictive model. The right-most result is for a lasso model using model-X knockoffs for feature selection. All results use BY correction rather than simple BH.

F Benchmark results with BY correction

Fig. 9 shows the results of the same benchmarks from Section 4, but using Benjamini-Yekutieli (BY) correction [Benjamini and Yekutieli, 2001] instead of BH correction. Although the BY correction is technically necessary to ensure FDR control, the results show that for our simulation the actual correction is conservative. Nevertheless, the BY-corrected HRT p-values still have high power in our simulations.

Algorithm 7 A knockoff-style empirical risk selection heuristic

```
1: procedure HYBRID(training data  $\mathcal{D}$ , test data  $\mathcal{D}'$ , model  $\pi$ , training objective  $\mathcal{L}_\pi$ ,  
empirical risk function  $\mathcal{G}$ , FDR threshold  $\alpha$ )  
2:   Fit  $\hat{\theta}$  by optimizing  $\mathcal{L}_\pi(\mathcal{D}, \theta)$ .  
3:   Compute the empirical risk on held out data,  $t \leftarrow \mathcal{G}(\mathcal{D}', \pi_{\hat{\theta}})$ .  
4:   for each feature  $j$  do  
5:     Sample  $\tilde{X}'_j \sim P(X_j | X_{-j})$ .  
6:     Create a new dataset  $\tilde{\mathcal{D}}$  by replacing  $X'_j$  in  $\mathcal{D}'$  with  $\tilde{X}'_j$ .  
7:     Compute the change in empirical risk,  $w^{(j)} \leftarrow \mathcal{G}(\tilde{\mathcal{D}}, \pi_{\hat{\theta}}) - t$ .  
8:      $\omega^* \leftarrow \min_{\omega \geq 0} \omega$ , subject to  $\left[ \frac{1 + \# w^{(j)} \leq -\omega}{\# w^{(j)} \geq \omega} \leq \alpha \right]$ .  
9:   return discoveries at the  $\alpha$  level:  $\{j : w^{(j)} \geq \omega^*\}$ .
```

G Disentangling knockoffs from empirical risk testing

In our experiments, the HRT has higher power than the lasso knockoffs of Candes et al. [2018]. Intuitively, there could be two reasons for this power. First, the HRT computes p -values and performs selection using BH; knockoffs work in a single-shot setting with simultaneous selection via a martingale-based filter [Barber and Candès, 2015]. Second, the lasso knockoffs use a coefficient magnitude test statistic rather than held out empirical risk. Either of these differences in procedures could conceivably be causing the discrepancy in power.

To investigate this discrepancy, we consider a hybrid knockoff-HRT heuristic approach. The hybrid approach uses the cross-validation empirical risk test statistic of the CV-HRT, but performs selection via the knockoff filter. Algorithm 7 presents the full algorithm.

Unlike the HRT and knockoffs, this hybrid approach does not produce valid test statistics, even with the true complete conditionals. The knockoff filter requires that the signs of each test statistic are i.i.d. coin flips under the null. The hybrid approach, by only conditioning on the complete conditionals and not on the joint, creates a dependency between all of the signs. Thus, applying the knockoff filter to the one-shot empirical risk statistics is not guaranteed to control FDR. The closely related “swap” knockoff statistic [Gimenez et al., 2018] would be a valid knockoff approach that uses empirical risk, but it would require changing the predictive model. Since we are only interested in determining power here, FDR control is less important than retaining the same predictive model.

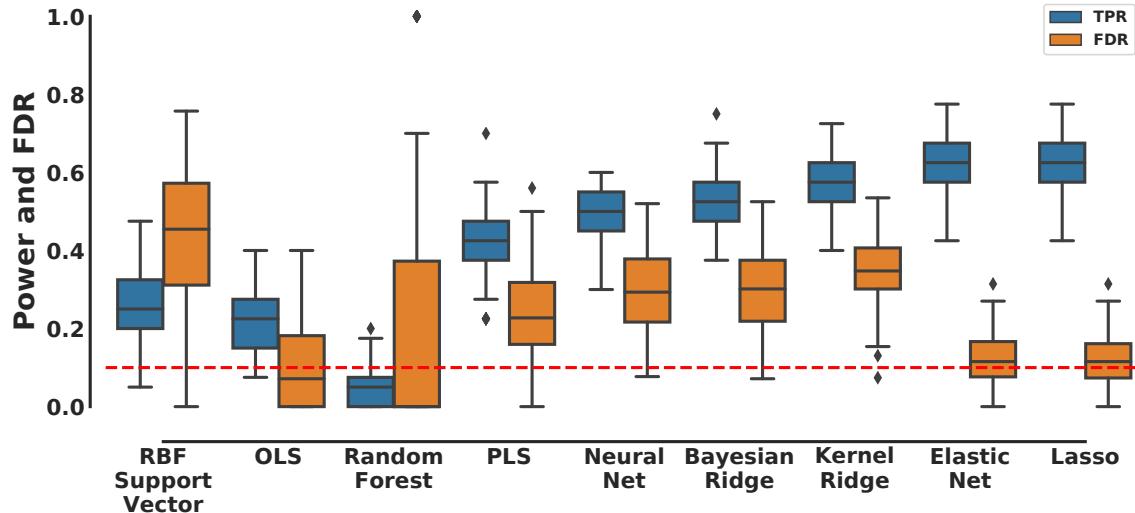


Figure 10: Power and FDP for different choices of predictive model used with Algorithm 7, ordered by empirical risk of the predictive model.

Fig. 10 shows the results of the same simulations, but using Algorithm 7 instead of the HRT. The method maintains the correlation between predictive performance and power, as shown in Fig. 11, and achieves similar power to the HRT. This suggests (at least for this simulation) the low power of lasso coefficient knockoffs is likely due to the choice of test statistic. However, unlike the HRT, the method is unable to control FDR in all predictive models. In addition to the theoretical issues with dependency between test statistics, knockoffs also lack a way to calibrate of the complete conditional, which may contribute to the FDP inflations in this case. Finally, it is still possible that the power discrepancy between the lasso coefficient statistic and empirical risk is specific to our simulations; establishing the relative power of empirical risk to other statistics is future work.

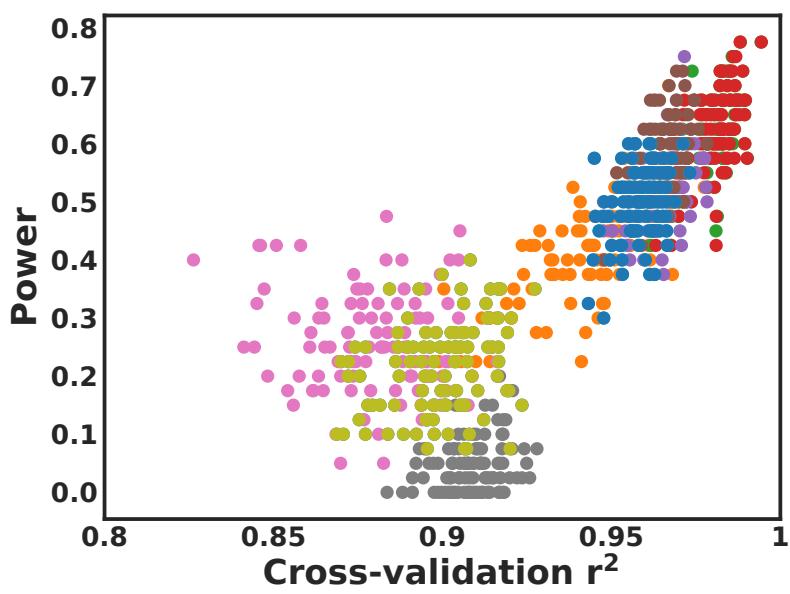


Figure 11: Predictive model performance versus feature selection power for each model and independent trial using Algorithm 7.