Causal Machine Learning and Experiments

Jack Rametta & Sam Fuller

March 4, 2026

Outline

Today's presentation is divided into two components:

1. The presentation of our first paper in this series.
2. A broad overview of our plans for projects in this space.

$\hookrightarrow$ Context: We have a book project (w/ Chris Hare) in the works, we greedily want as much feedback as we can get.

Questions During the Presentation

If you have any questions during the course of the presentation, do not hesitate to ask! We're happy to clarify during or after the presentation.

The Balance Permutation Test: A Machine Learning Replacement for Balance Tables

Motivation and Background

This project began with a simple question: **Should you include a balance table + balance tests with your experiment?**

Example balance table (2-arm, N = 1,000):

	Control		Treatment			
	Mean	Std. Dev.	Mean	Std. Dev.	Diff. in Means	Std. Error
Female	0.49	0.50	0.43	0.50	-0.06	0.05
Black	0.26	0.44	0.19	0.39	-0.07	0.04
Income	51318	10787	49345	9820	-1972*	949
Democrat	0.47	0.50	0.43	0.50	-0.03	0.05
Age	53.10	17.79	49.78	19.49	-3.32*	1.69

↪ It turns out this is a highly contested question!

The Balance Debate

Pro Balance Table (Gerber et al. 2014)

- Inference in experiments *depends* on successful random assignment.
- Randomization *can* fail for all kinds of idiosyncratic reasons.
- Approximate marginal covariate balance is a testable implication of random assignment (*multiple testing!).
- Balance tables can catch assignment/balance issues.

Anti Balance Table (Mutz, Pemantle, and Pham 2019)

- Balance tables not useful for detecting assignment errors in practice.
- Balance tables encourage post-hoc reg. adjustments, open door to specification search.
- The "balance test and adjust" approach isn't a sound estimation procedure relative to PAP.

Balance Tables: What Are They Good For?

Returning to the balance table for simple experiment with 5 covariates, N = 1,000:

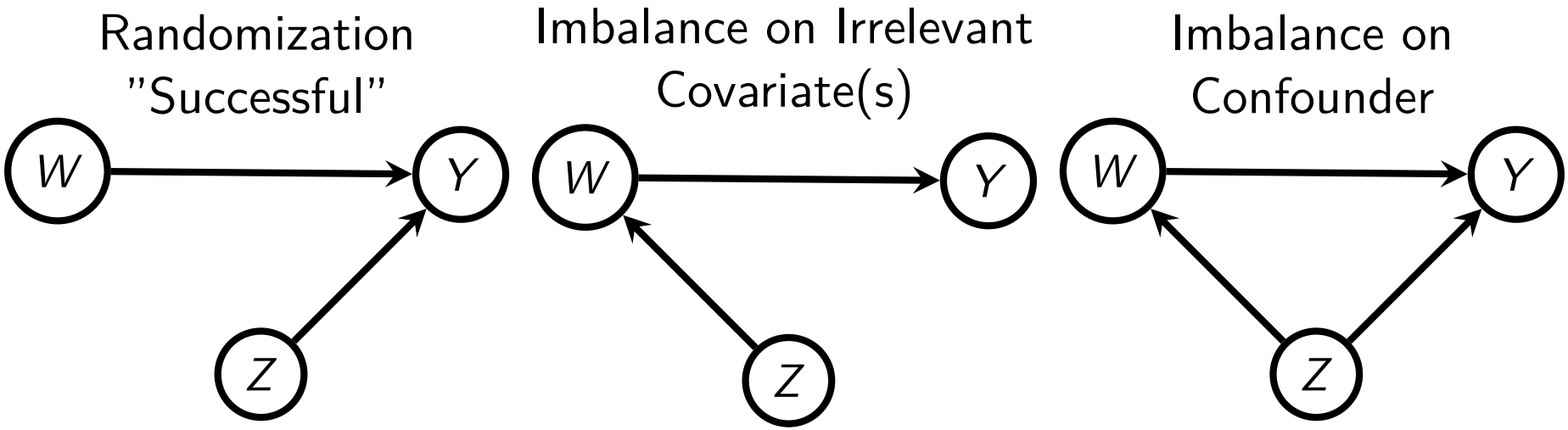
	Control		Treatment			
	Mean	Std. Dev.	Mean	Std. Dev.	Diff. in Means	Std. Error
Female	0.49	0.50	0.43	0.50	-0.06	0.05
Black	0.26	0.44	0.19	0.39	-0.07	0.04
Income	51318	10787	49345	9820	-1972*	949
Democrat	0.47	0.50	0.43	0.50	-0.03	0.05
Age	53.10	17.79	49.78	19.49	-3.32*	1.69

$\hookrightarrow$ What if I told you there are no black, female observations in the treatment group?

Assignment is conditioned by female, black, and income. Interactive imbalances not detected!

Why Imbalance Can Matter

So do we need balance tables at all? Why not do away with them?



Imbalance $\Rightarrow$ can induce confounding.

The Baby in the Bathwater

Another POV...

Eckles (2022)

"Thomas Aquinas argued that God's knowledge of the world upon creation of it is a kind of practical knowledge: knowing something is the case because you made it so. One might think that in randomized experiments we have a kind of practical knowledge: we know that treatment was randomized because we randomized it. **But unlike Aquinas's God, we are not infallible, we often delegate, and often we are in the position of consuming reports on other people's experiments.**"

We agree. Echoes arguments from Gerber et al. 2014 among others.

$\hookrightarrow$ We should interrogate assignment for possible issues, but balance tables (& tests) not great.

Our Argument: The Best of Both Worlds

Can we have our cake and eat it too? Yes!

1. Take identification seriously even if we control assignment mechanism.
2. Balance tables/tests are weak tests of assignment failure. Like a knife in a gun fight.
3. Instead load for bear: **test for systematic imbalance using machine learning.**
4. To adjust for imbalance & improve precision, estimate unadjusted & "doubly-robust" effects with a double-robust machine learning estimator (DRML).

$\hookrightarrow$ DRML is our catchall for AIPW, TMLE, DoubleML, etc.

Benefits of Our Approach

Taken together, our approach ameliorates concerns of Gerber, Mutz...

1. Balance table replacement that works. Marginal and joint distributions examined. Data-driven modeling, no post-hoc user-spec needed.

2. Works w/ preregistration paradigm **and** allows for efficiency improvements. No post-hoc adjustments!

3. Works off-the-shelf with one line of code. Model tuning, sample-splitting, variable importance calculations all done for you. `MLbalance::random_check`

4. Detect experiments w/ fabricated/manipulated data.

The Balance Permutation Test

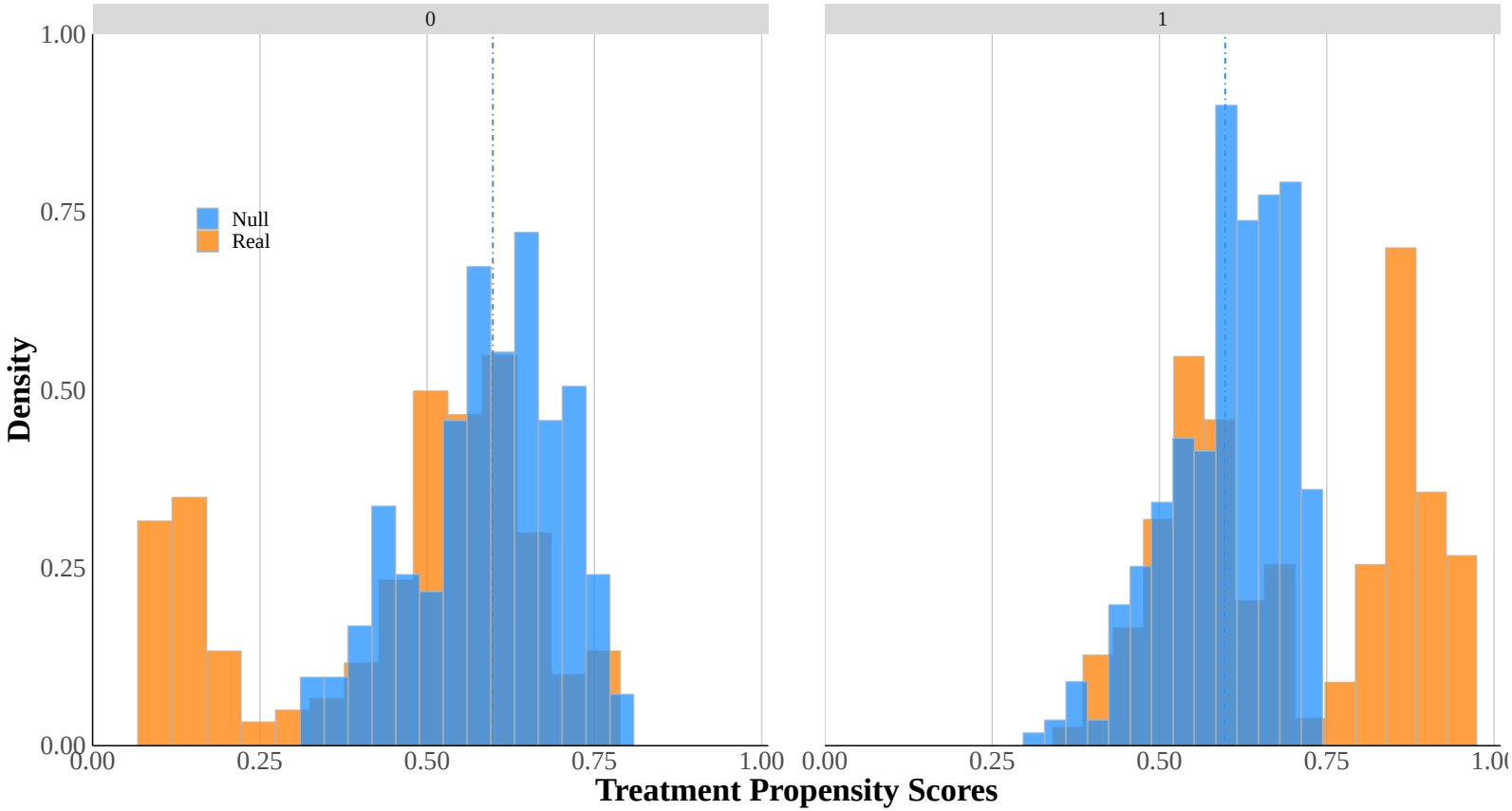
We propose the "balance permutation test" (inspired by Breiman (2001)) as follows:

1. Build and tune black box model of real treatment assignment using relevant pre-treatment covariates. **No colliders or irrelevant variables!**
2. Build black box model of *permuted*, or simulated, treatment assignment using pre-treatment covariates & the same tuning parameters.
3. Ocular test: compare the treatment propensity score distributions generated by models 1 and 2.
   $\hookrightarrow$ What to look for: differences in central tendency or variance, extreme/deterministic propensity scores, strange clusters of observations, or bimodality. In other words, weird stuff.

That's basically it. No single test statistic proposed (more on this later).

Balance Permutation Example: Problem

Let's reconsider the motivating example. Here's the BPT result:



Note: Dotted lines represent mean values for the null and real treatment propensity distributions.

↪ Notice the extreme propensity scores!

Balance Permutation Example Cont'd

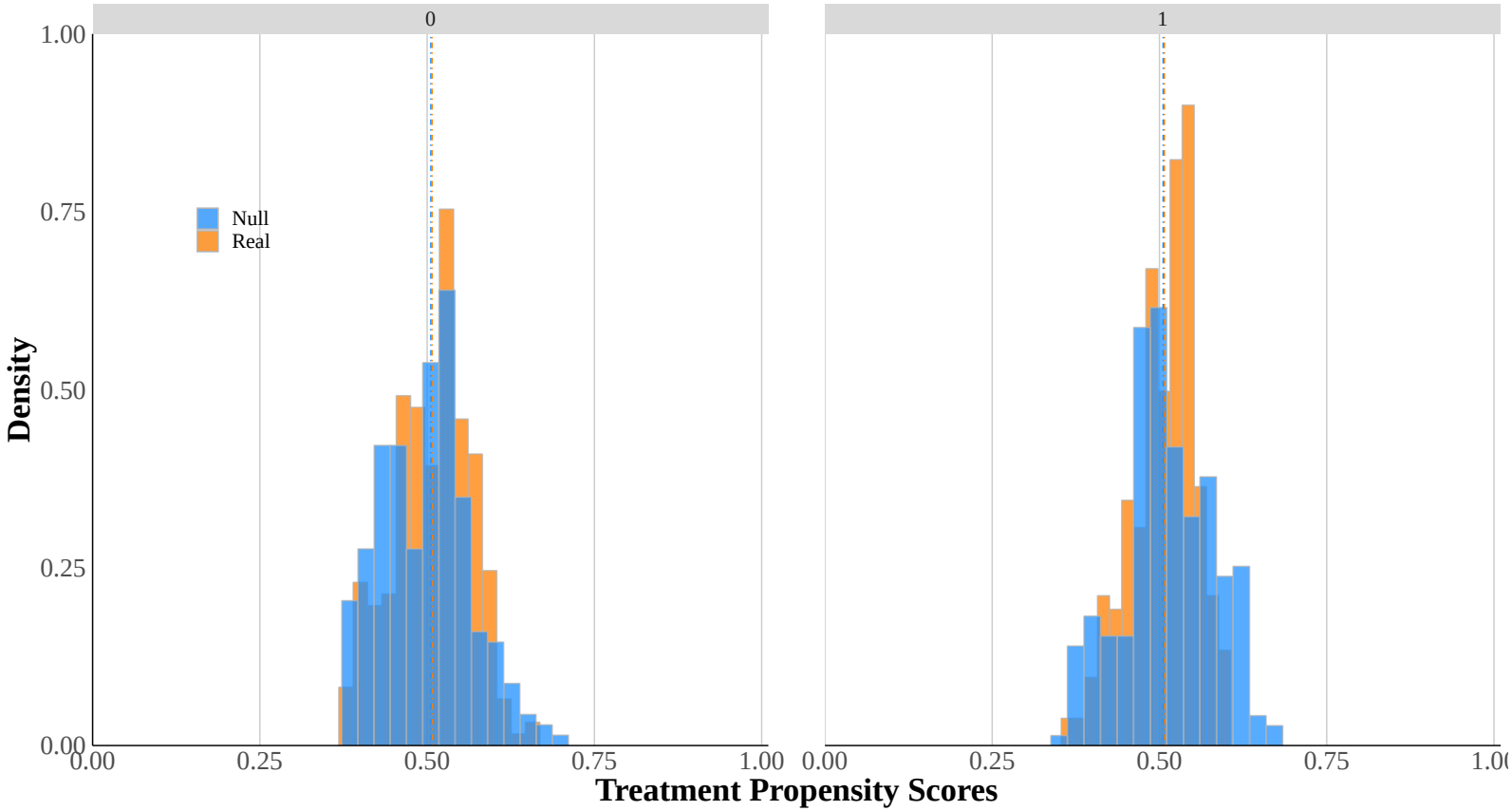
Examining observations with propensity scores close to zero yields:

Female	Black	Income	Democrat	Age	W	Treat Props
1	1	44308	1	55.11	0	0.080
1	1	45320	1	73.52	0	0.081
1	1	39735	1	54.22	0	0.083
1	1	47080	1	62.19	0	0.088
1	1	44725	1	45.83	0	0.095
1	1	58138	1	53.52	0	0.082
1	1	60283	0	62.75	0	0.090
1	1	41606	1	50.68	0	0.079
1	1	46838	1	53.41	0	0.069
1	1	69806	0	61.37	0	0.095

$\hookrightarrow$ Indeed, the bottom quartile of treatment propensity scores is exclusively black and female.

Balance Permutation Example: No Problem

What if the assignment process isn't contaminated? The figure should look something like this:

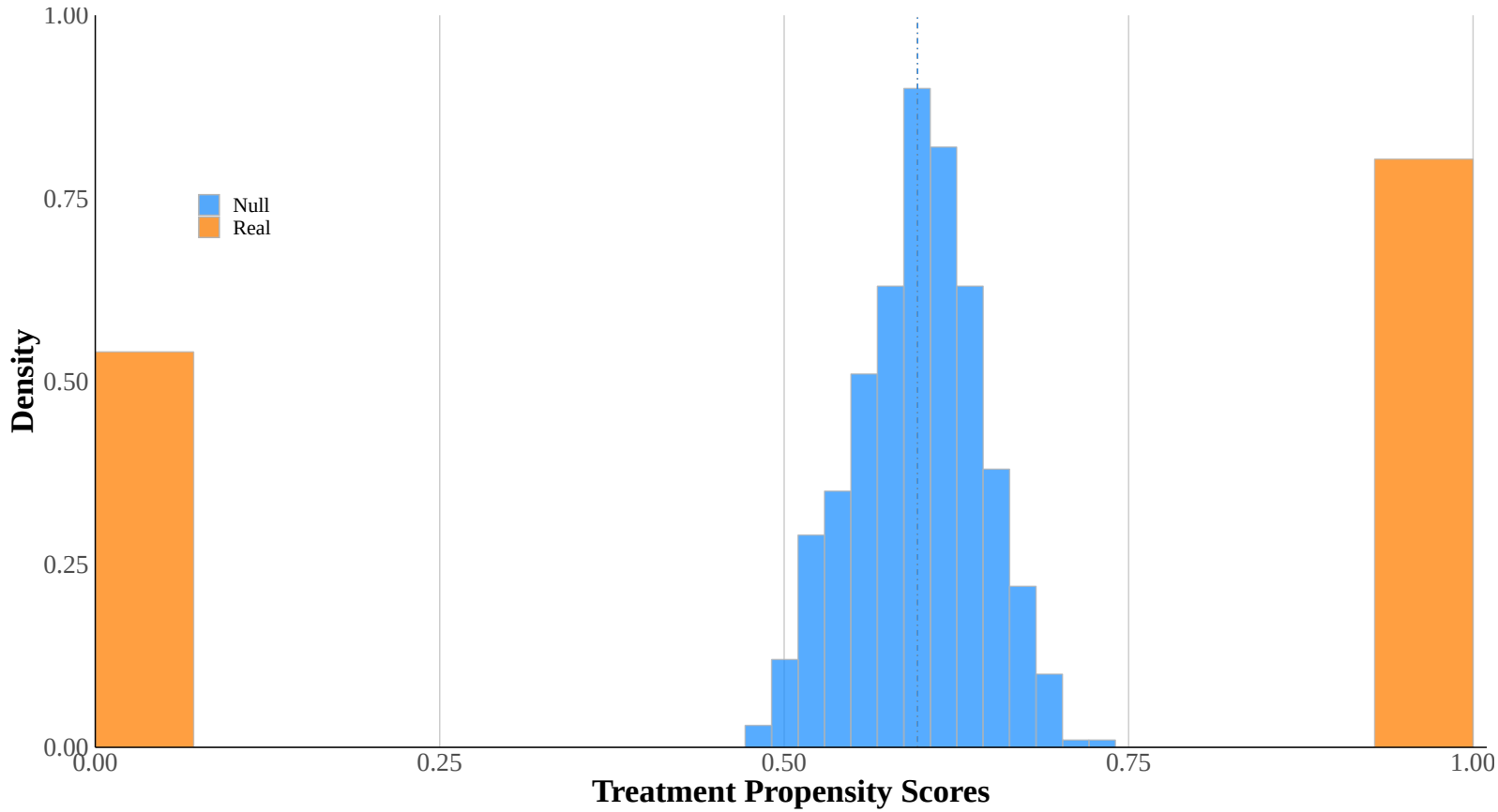


Note: Dotted lines represent mean values for the null and real treatment propensity distributions.

↪ Looks good.

Balance Permutation Example: Left Hand on the Right

By the way, if you accidentally include treatment assignment as a predictor?

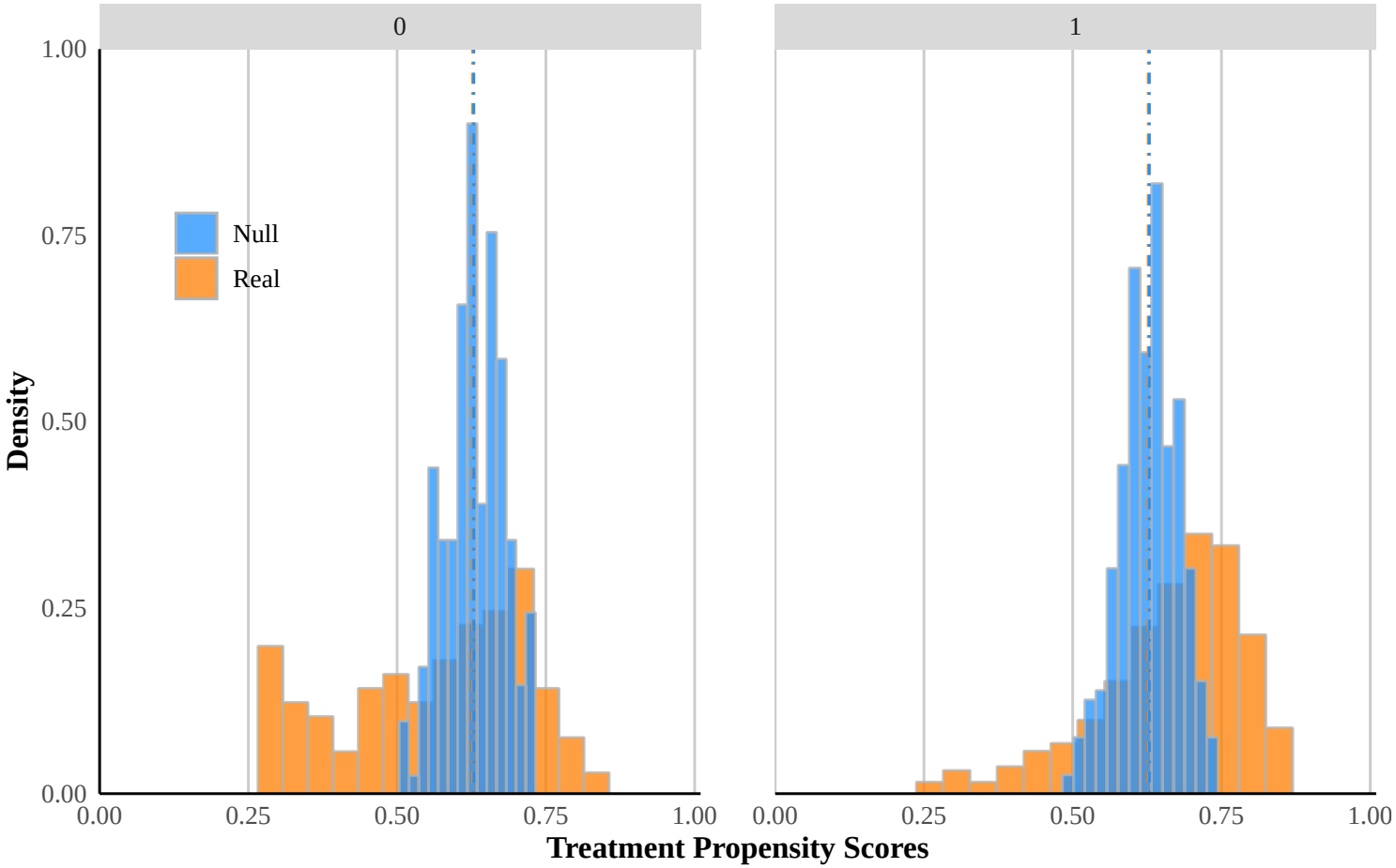


Note: Dotted lines represent mean values for the null and real treatment propensity distributions.

↪ Bimodal around 0 and 1. Very bad something horribly wrong.

BPT Example: Karim 2020 APSR

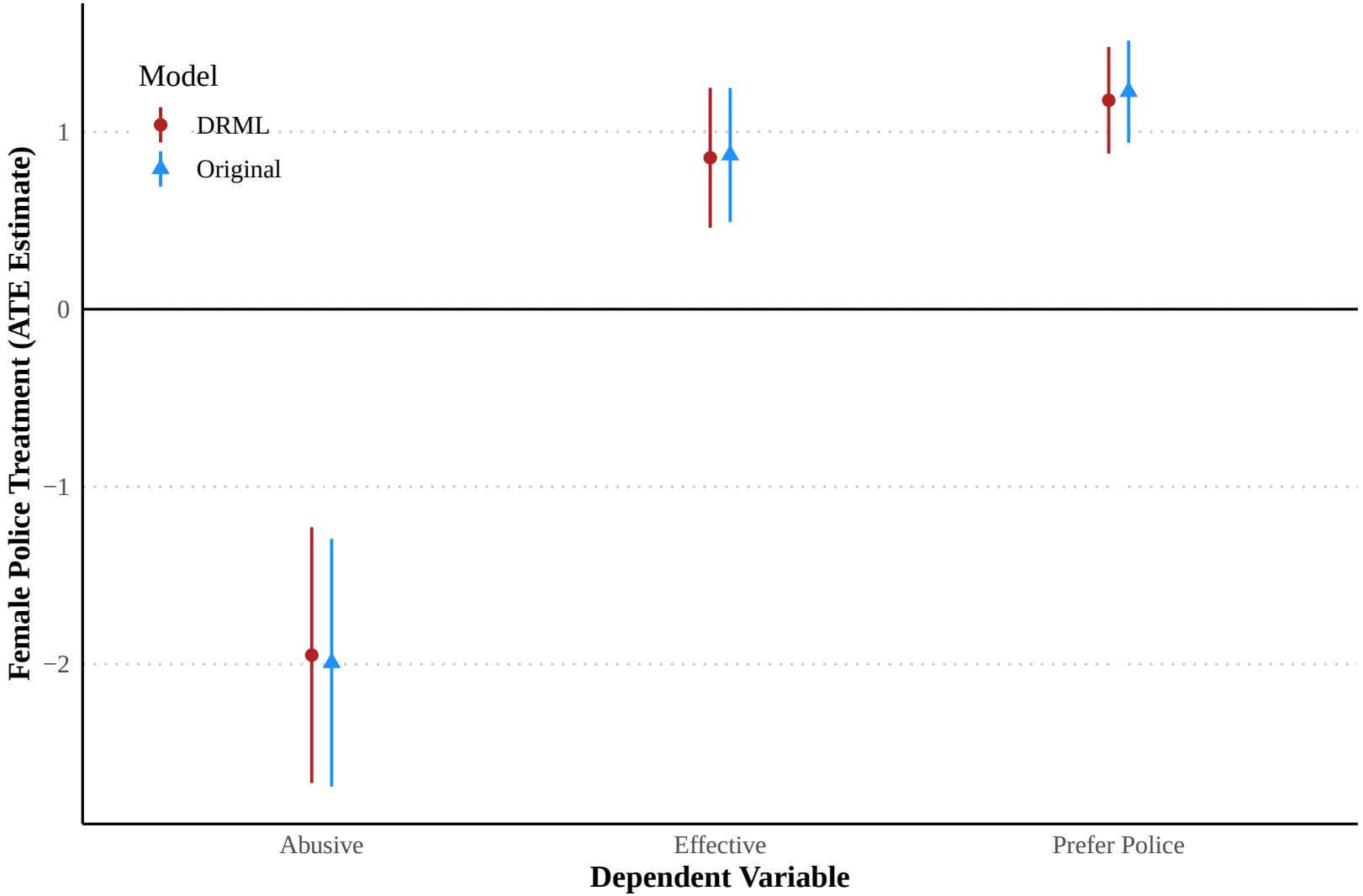
A recent field experiment noted unusual imbalance, our method recovers.



Note: Dotted lines represent mean values for the null and real treatment propensity distributions.

BPT Example: Karim 2020 APSR

DRML ATE estimates reassure their main effects are fine!



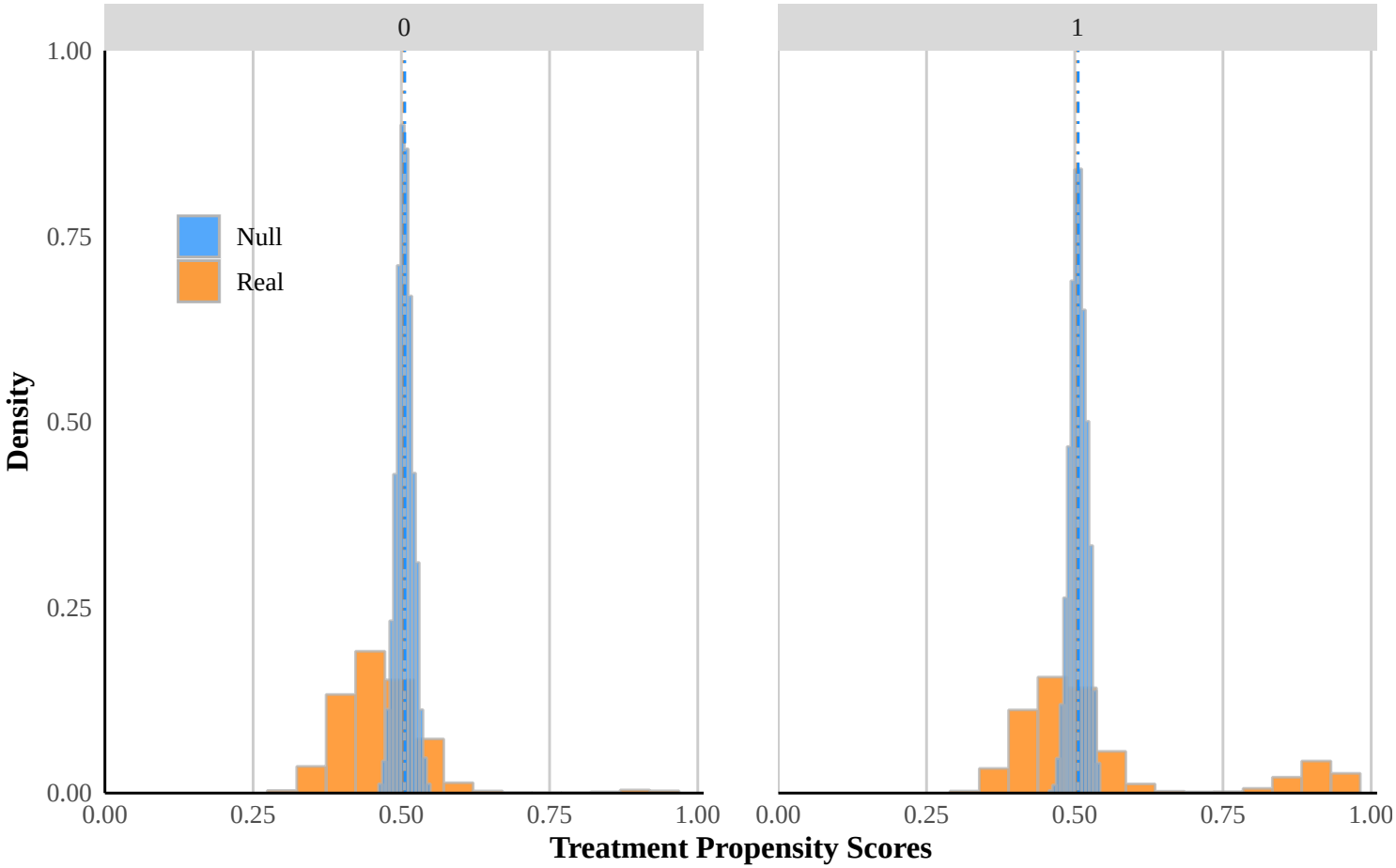
Side Benefit: BPT for Fabrication Checks

Our approach has another useful function: as a post facto check for data fabrication or manipulation.

This works when fabrication results in an unusual relationship between assignment and a pre-treatment covariate.

$\hookrightarrow$ Turns out, this is frequently the case. (see e.g., Berenbaum 2021)

Uncovering Data Manipulation: Shu et al. (2012)



Note: Dotted lines represent mean values for the null and real treatment propensity distributions.

↪ Won't always catch fabrication, but works in every case we've tested it on. There's probably some selection bias going on there...

Why No Test Statistic?

We considered several test statistics and methods to compare the treatment propensities:

- Location and/or scale tests: diff-in-means, diff-in-variance, Cocconi test, Kolmogorov–Smirnov test, etc.
- Model fit stats on null vs. real models.
- Comparison of model fitness over many permutations (Gagnon-Bartsch and Shem-Tov 2019).

Costs and benefits of each of these: **No test works in every instance.**

$\hookrightarrow$ Ultimately, "ocular test" with a check for extreme propensity scores is fast and robust. Works generally.

Technical Details, Under the Hood

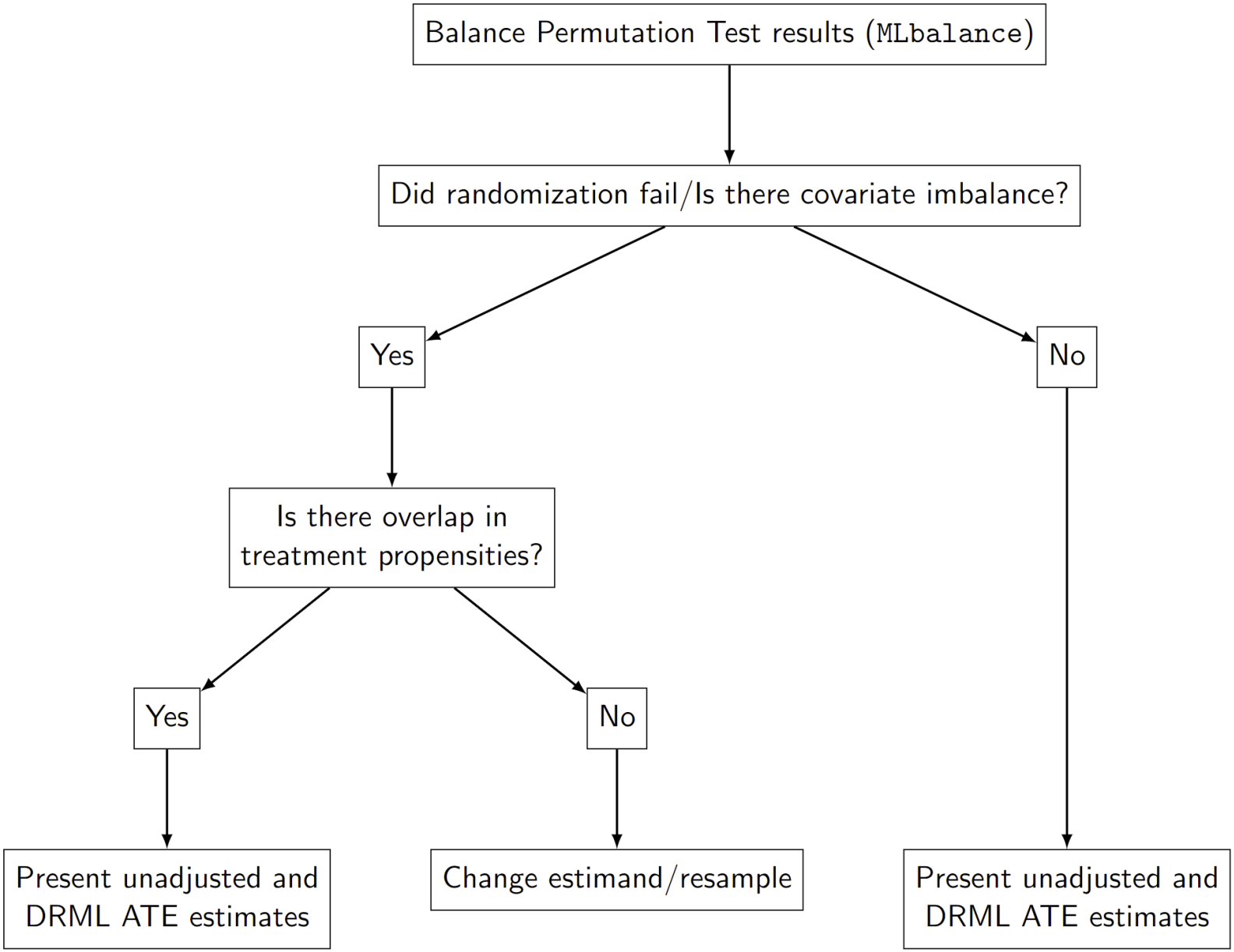
More vignettes: Gerber and Green (2000), Moehler and Conroy-Krutz (2016), etc.

What's the black box? Honest n-step boosted random forest model (Ghosal and Hooker 2020)

- Naturally models interactions & non-linearities.
- Honest sub-sample splitting: half the data to choose splits, half to fill out leaves.
- Scrutability of random forest *and* predictive power of boosting.
- Model tuning less important relative to gbms, nets, etc.
- Built on `grf` architecture. Fast (C++), native multi-threading, no GPU needed, few external dependencies.

$\hookrightarrow$ Other algorithms possible, this one has advantages.

Putting the Pieces Together



Package Description

The balance permutation test implemented in our package `MLbalance`.

It has some nice features:

- Native parallel, fast, no GPU required (built on `grf` architecture).
- Built-in variable importance.
- No manual splitting or tuning required.

Try it out!

Causal Machine Learning for Experiments: Guidelines for Effective and Transparent Research

Motivation

Rapid expansion of literature applying machine learning for causal inference tasks (Grimmer, Roberts, and Stewart 2021).

One promising area is the adaptation of ML for subgroup analysis.

Methods like...

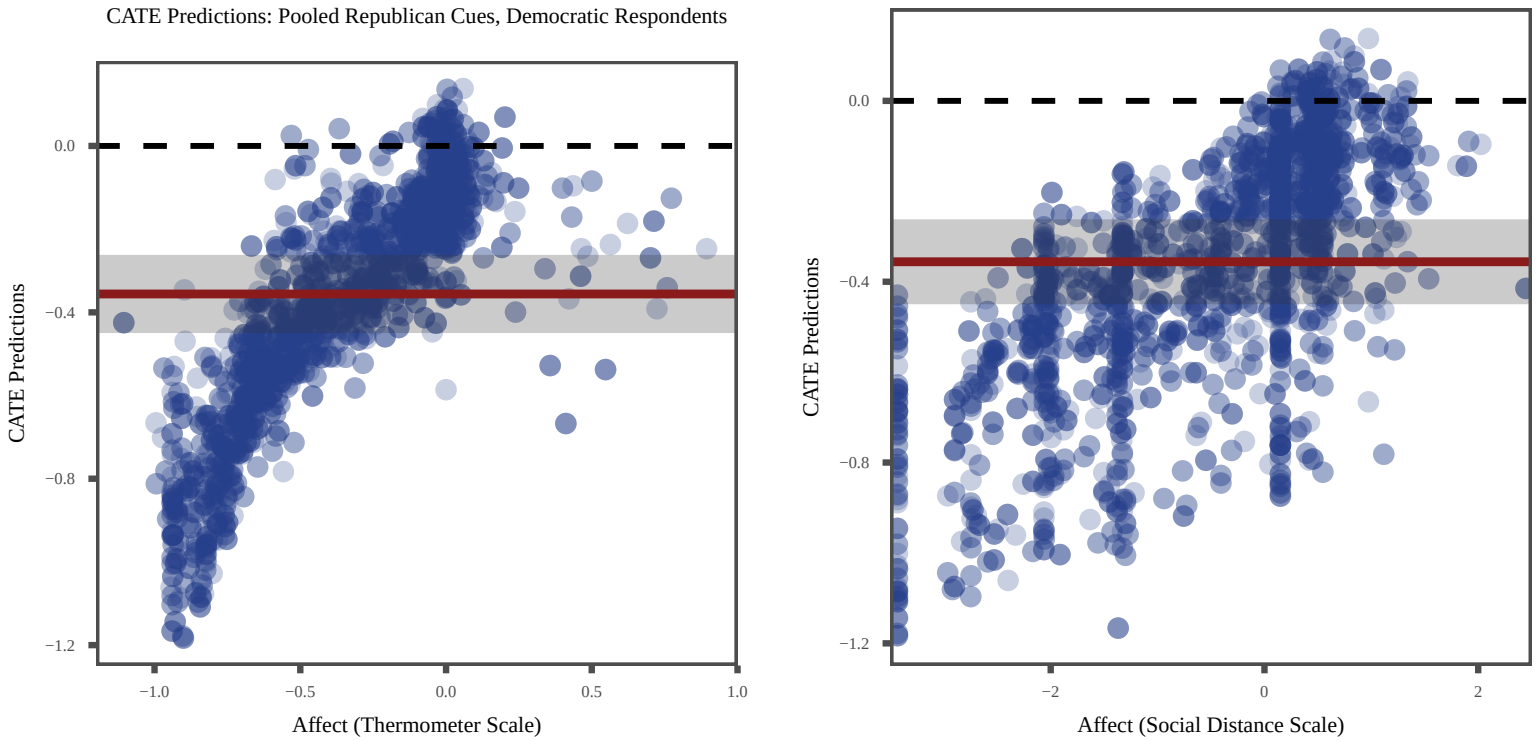
- Causal Forests (Wager and Athey 2018)
- Bayesian Additive Regression Trees (BART, many flavors) (Hahn, Murray, and Carvalho 2020; Dorie et al. 2022)
- Split weighted conformal quantile regression (Lei and Candès 2021)

$\hookrightarrow$ Offer powerful approaches for investigating heterogeneity in treatment effects across subgroups.

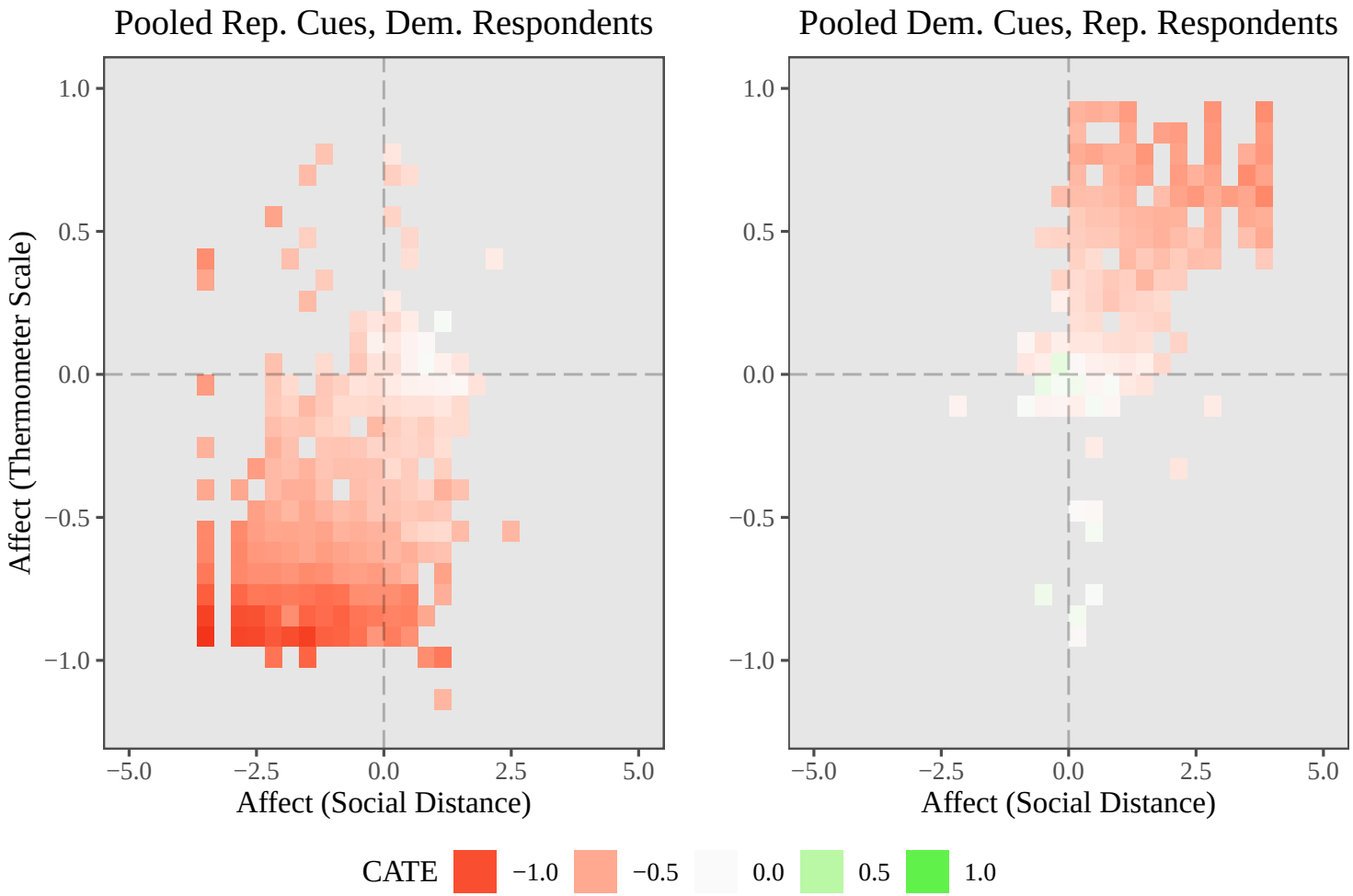
Motivation: An Example

When employed correctly, these methods allow researchers to estimate CATEs over many dimensions + increase resolution of experimental findings.

An example:



Motivation: An Interaction in Affective Space



↪ These methods are great, but potential for misuse (Ratkovic 2021).

Overview

Goal: Provide a clear framework for researchers to apply ML tools to experimental data in a principled way.

↪ **Approach**:

1. Generate clear guidelines for experimentalists to use ML for subgroup analysis, detecting HTEs, and estimate CATEs.
2. Outline other helpful applications of ML for measurement, balance, ATE estimation, etc.
3. Develop a package family `MLC` (Machine Learning for Causal inference) that implements these approaches in a user friendly, robust way.
   - Works "off-the-shelf," follows *DeclareDesign* framework.
   - One component already complete: `MLbalance`.

A Simple (?) Question

How should we calculate conditional average treatment effects?

---

1. Multiple interactions in a simple regression?
   - Could overfit, likely misspecified (Beiser-McGrath and Beiser-McGrath 2020). Need lots of power.
2. Separate single interaction regression models?
   - Could be omitted interaction bias! (Blackwell and Michael P. Olson 2022b)
3. Off-the-shelf machine learning methods (RF, boosting, etc.)?
   - Not designed for causal inference, likely overfit (Ratkovic and Tingley 2023).
4. But wait... do we even have heterogeneity?

And our journey down the rabbithole began...

Lit: Why, How, & When Should We Use ML for Subgroup Analysis?

Two works in PS have covered this topic in depth:

Green and White (2023) & Ratkovic (2021)

- Green and White provide a detailed overview of a few methods, including applications, benefits, and drawbacks.
- Ratkovic offers a detailed general checklist for conducting principled subgroup analysis with ML.
  - Recommends the use of (sub)sample splitting + ML to test and estimate heterogeneity in a wide variety of instances.

However, general guidelines for how ML should be used in experiments, from design to analysis, are nonexistent.

The Experimentalist's Checklist: *Transparency Over Simplicity*

1. Design a clear experiment and check expectations
2. Liberally and accurately measure the relevant covariate space
3. Test for assignment failures and/or test for systematic attrition using ML
4. Create and tune an accurate model
5. Test for and identify treatment effect heterogeneity using the *entire* covariate space
   - For each treatment with identified heterogeneity report variable importance
   - Only present CATEs if the variable is identified as important
6. Present results from DRML estimation, both ATEs and CATEs, alongside standard ATE estimates

The Experimentalist's Checklist: *Transparency Over Simplicity* Cont.

1. Design a clear experiment and check expectations
2. **Liberally and accurately measure the relevant covariate space**
3. Test for assignment failures and/or test for systematic attrition using ML
4. **Create and tune an accurate model**
5. **Test for and identify treatment effect heterogeneity using the *entire* covariate space**
   - For each treatment with identified heterogeneity report variable importance
   - Only present CATEs if the variable is identified as important
6. **Present results from DRML estimation, both ATEs and CATEs, alongside standard ATE estimates**

Measuring Covariates

Given the ability of causal ML algorithms to handle large covariate spaces, researchers should more broadly measure covariates.

However, measurement is still incredibly important, causal ML methods cannot fix bad questions or scales: $\hookrightarrow$ Garbage in, garbage out.

Measurement should be taken more seriously and unsupervised ML (scaling) should be used more.

The final section of this presentation shows the benefits of accurate and broad measurement in a real experiment.

Creating and Tuning an Accurate Model

Making sure that a model provides (relatively) accurate predictions is underappreciated.

However, this is directly related to quality inference: if a model cannot predict an outcome, then the statistically significant relationships between IVs and DVs can be meaningless.

Published work using CF has already made the mistake of not tuning models well (see Zheng and Yin (2023)).

There are numerous tests for ensuring model accuracy, and multiple should be reported.

Testing for Heterogeneity

Researchers should always test for treatment effect heterogeneity using all covariates and a sample-splitting procedure (like that implemented in CF).

Tests for heterogeneity include the best linear projection test from Chernozhukov et al. (2022), the bound test from Athey and Wager (2019), recent sequential RATE test Wager (2024).

Conditional average treatment effects should only be investigated (and eventually calculated) if these tests reveal evidence of treatment effect heterogeneity.

Doubly-Robust Machine Learning

DRML treatment effect estimation works within the standard linear model framework. Specifically, it uses a partially linear model:

DRML

$$Y = D\theta_0 + g_0(X) + \zeta, \quad \mathbb{E}(\zeta|D, X) = 0,$$
$$D = m_0(X) + V, \quad \mathbb{E}(V|X) = 0,$$

- Y is our outcome variable.
- $D = \{0, 1\}$ is an indicator variable of treatment.
- The vector $X = (X_1, ..., X_p)$ is composed of all pre-treatment covariates.
- ζ and V are random errors for each equation.

$\hookrightarrow$ We model both $g_0(X)$ (outcome $\sim$ covs) and $m_0(X)$ (treatment $\sim$ covs) using some flavor of ML.

DRML Cont.

In this setup, $g_0(X)$ and $m_0(X)$ are nuisance parameters. Only need good predictions!

In experimental context, DRML with good randomization turns into a fancy regression adjustment.

Depending on the implementation, DRML employs cross-fitting or other forms of sample-splitting to guard against overfitting.

$\hookrightarrow$ We employ a variant of DRML where the plug-ins are honest random forests. Some gains possible from other plug-ins but this is a fast and high-performance default.

DRML Benefits

1. Model specification doesn't fall from the sky!
2. High dimensional relations and interactions? No problem.
3. Nonlinear function form? No problem.
4. Kills temptation for table 2 fallacy.
5. Sample splitting/cross-validation turns out to be good for prediction AND inference.
   - Athey and Imbens (2016), Ratkovic and Tingley (2023), and Blackwell and Michael P Olson (2022a), etc.!

$\hookrightarrow$ In our sims, DRML significantly improves precision relative to unadjusted, reg. adjusted, and the popular Lin estimator.

ML for Treatment Effect Estimation: Simulation Results

Figure: Average reduction in absolute error of the DRML treatment effect estimates versus unadjusted difference in means, regression-adjusted, and Lin-adjusted effects.

480 permutations x 2,000 seeds = 960,000 simulations covering many DGPs.

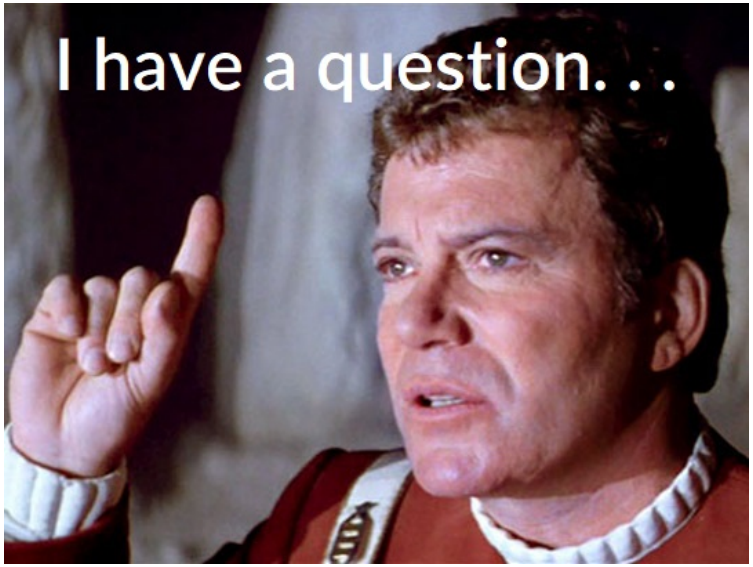
↪ (*untuned*) DRML wins in $\sim 99\%$ **of cases.**

Future Work

We plan on applying ML to a host of other topics, some of which are chapters of our book project with Chris Hare.

1. ML for Systematic Attrition Detection
2. ML for Treatment Effect Estimation
3. Unifying Unsupervised and Supervised ML
4. Implications of ML for Experimental Design
5. A Meta-Reanalysis of CATEs in Experimental Political Science

Questions ***For You!***



1. Other prediction tasks we should evaluate in an experimental context?

2. What are you skeptical of? What do we need to do to convince you of the benefits of these methods?

3. What's confusing? Jack and I have spent two years in this space, we know these methods can appear (or be) incredibly esoteric.

Thank You! We Look Forward to Your Feedback.