Testing the Fairness-Accuracy Improvability of Algorithms

Eric Auerbach
(Northwestern)

Annie Liang
(Northwestern)

Kyohei Okumura
(Northwestern)

Max Tabord-Meehan
(UChicago)

introduction

- algorithms are used by organizations to guide high-stakes decisions
  - which patients receive treatment? which borrowers are granted a loan?
- many of these algorithms have a disparate impact
  - their benefits/harms fall disproportionately on specific social groups
  - however organizations value other objectives (e.g., accuracy, profit)
- can we reduce disparate impact without compromising other objectives?
- **legal relevance**: under US federal law, a policy with disparate impact may be permissible if it is necessary to achieve a legitimate business interest

three-part legal process

codified under Title VII of the Civil Rights Act of 1964

(cf. 42 U.S.C. § 2000e–2(k); Title VI Manual of DoJ)

PART 1:

ESTABLISHING DISPARATE IMPACT

PART 2:

ESTABLISHING BUSINESS NECESSITY

PART 3:

IS THERE A VALID LESS-DISCRIMINATORY ALTERNATIVE?

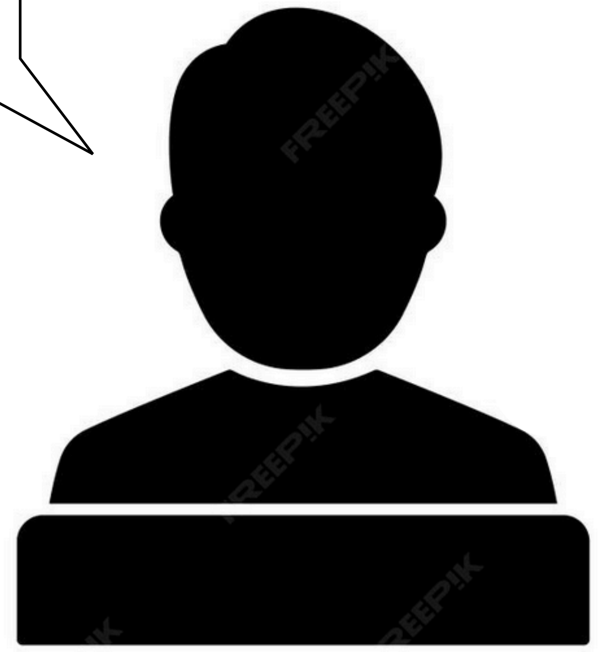
PART 1:

ESTABLISHING DISPARATE IMPACT

ORGANIZATION

(employs an algorithm, e.g., to make hiring decisions)

the existing algorithm has disproportionate harms for a certain group of people



CHALLENGER

(e.g., a regulator or private individual)

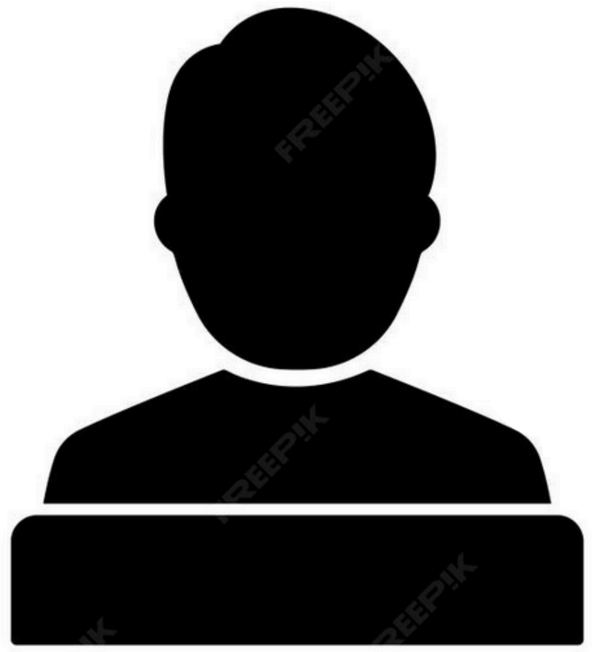
PART 2:

ESTABLISHING BUSINESS NECESSITY

such disparate impact is necessary to achieve a legitimate business interest

ORGANIZATION

(employs an algorithm, e.g., to make hiring decisions)



CHALLENGER

(e.g., a regulator or private individual)

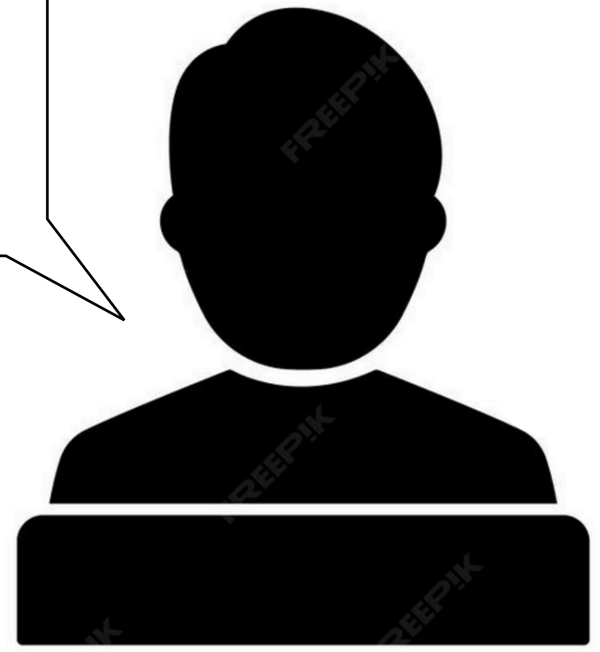
ORGANIZATION

(employs an algorithm, e.g., to make hiring decisions)

PART 3:

IS THERE A VALID LESS-DISCRIMINATORY ALTERNATIVE?

this **alternative algorithm** would achieve those same business objectives, and has less disparate impact



WINS →

CHALLENGER

(e.g., a regulator or private individual)

PART 1:

ESTABLISHING DISPARATE IMPACT

PART 2:

ESTABLISHING BUSINESS NECESSITY

our framework is useful for evaluating this final part →

PART 3:

IS THERE A VALID LESS-DISCRIMINATORY ALTERNATIVE?

other potential applications

can we reduce disparate impact without compromising other objectives?

- organization itself may ask this (e.g., integrity, reputation, risk mitigation)
- regulator may seek to provide guidance on algorithms that should be avoided

contribution

this paper:

- introduce a **conceptual framework** for assessing the existence of less discriminatory alternatives, building on Liang, Lu, Mu, and Okumura (2024)

- develop a simple and practical **test** for testing the "fairness-improvability" of a status-quo algorithm given data
  - a new econometric result on bootstrap consistency specifically tailored to AI settings
  - a game-theoretic foundation for repeated sample splitting

- apply the test to a healthcare algorithm used in the U.S., and find strong statistical evidence of the existence of less discriminatory alternative

bootstrap

some relevant literature

- finding less discriminatory algorithms:
  - Coston et al. (2021), Viviano and Bradic (2023), Blattner and Spiess (2023), Gillis et al. (2024) …
  - primarily focus on how to find a good algorithm by solving an optimization problem
  - **our focus**: test if the improvement of the new algorithm is statistically significant
  - complementary: any method developed in the literature can be used with our test
- closely related work: Liu and Molinari (2024)
  - study estimation of the entire "fairness-accuracy frontier"
  - **our focus:** a narrower question "is there a better alternative?"
    - accommodates any exogenous constraints on algorithm class
      - e.g., capacity constraints, shape restrictions (linear, monotone, etc.)

outline

- model
- testing procedure
- microfoundation
- empirical application

model

testing procedure

microfoundation

empirical application

setup

- each subject i is described by three variables:
  - **outcome** Y_i taking values in $\mathscr{Y} \subseteq \mathbb{R}$
  - **covariate vector** X_i taking values in $\mathscr{X} \subseteq \mathbb{R}^d$
  - **group** $G_i \in \mathscr{G} := \{r, b\}$

e.g.

need for medical procedure

image scans
# of past hospital visits
blood tests

race (Black or White)

setup

- each subject i is described by three variables:
  - **outcome** Y_i taking values in $\mathcal{Y} \subseteq \mathbb{R}$
  - **covariate vector** X_i taking values in $\mathcal{X} \subseteq \mathbb{R}^d$
  - **group** $G_i \in \mathcal{G} := \{r, b\}$

e.g.

need for medical procedure

image scans
# of past hospital visits
blood tests

race (Black or White)

- in the population, $(X_i, Y_i, G_i) \sim_{iid} P$
- an **algorithm** is a mapping $a : \mathcal{X} \to \mathcal{D}$ from the covariate vector into a decision
  - $a(X_i)$: decision for subject i

G_i can be included in X_i

setup

- there is a **status quo algorithm** a_0 which is under contention
- **analyst**'s goal is to assess whether it is possible to reduce the "disparate impact" of a_0 without compromising on another objective
- we will call these two objectives simply fairness and accuracy

an umbrella term for any objective of the organization

how we define accuracy and fairness

- accuracy utility function $u_A : \mathscr{X} \times \mathscr{Y} \times \mathscr{D} \rightarrow \mathbb{R}_+$
- fairness utility function $u_F : \mathscr{X} \times \mathscr{Y} \times \mathscr{D} \rightarrow \mathbb{R}_+$

u_F is possibly identical to u_A, but can be different

how we define accuracy and fairness

- accuracy utility function $u_A : \mathcal{X} \times \mathcal{Y} \times \mathcal{D} \to \mathbb{R}_+$
- fairness utility function $u_F : \mathcal{X} \times \mathcal{Y} \times \mathcal{D} \to \mathbb{R}_+$

- consider expected utility under algorithm a for each group $g \in \{r, b\}$:

$$U_A^g(a) := E_P\left[u_A(X, Y, a(X)) \mid G = g\right], \quad U_F^g(a) := E_P\left[u_F(X, Y, a(X)) \mid G = g\right]$$

- accuracy for group g of algorithm a is defined as $U_A^g(a)$
- (un)fairness (or disparate impact) of algorithm a is defined as $|\, U_F^r(a) - U_F^b(a) \,|$

how we define accuracy and fairness

- accuracy utility function $u_A : \mathcal{X} \times \mathcal{Y} \times \mathcal{D} \to \mathbb{R}_+$
- fairness utility function $u_F : \mathcal{X} \times \mathcal{Y} \times \mathcal{D} \to \mathbb{R}_+$

- consider expected utility under algorithm a for each group $g \in \{r, b\}$:

$$U_A^g(a) := E_P\left[u_A(X, Y, a(X)) \mid G = g\right], \quad U_F^g(a) := E_P\left[u_F(X, Y, a(X)) \mid G = g\right]$$

definition:

- algorithm a_1 is more accurate than a_0 if $U_A^g(a_1) > U_A^g(a_0)$ for each group $g \in \{r, b\}$
- algorithm a_1 is more fair than a_0 if $|U_F^r(a_1) - U_F^b(a_1)| < |U_F^r(a_0) - U_F^b(a_0)|$

examples: fairness and accuracy criteria

1. **correct classification rate**:

$$U^g(a) := P(Y = a(X) \mid G = g)$$

average probability of correct diagnosis for patients in group g

Y: sick or not

$a(X)$: treat or not

G: race (white or black)

$$U_A^g = U_F^g =: U^g$$

examples: fairness and accuracy criteria

1. **correct classification rate**:

$$U^g(a) := P(Y = a(X) \mid G = g)$$

2. **correct positive rate**:

$$U^g(a) := P(Y = a(X) \mid Y = 1, G = g)$$

average probability of correct diagnosis for patients in group g **who are sick**

Y:	sick or not
$a(X)$:	treat or not
G:	race (white or black)

$$U_A^g = U_F^g =: U^g$$

examples: fairness and accuracy criteria

Y:	sick or not
$a(X)$:	treat or not
G:	race (white or black)

$$U_A^g = U_F^g =: U^g$$

1. **correct classification rate**:

$$U^g(a) := P(Y = a(X) \mid G = g)$$

2. **correct positive rate**:

$$U^g(a) := P(Y = a(X) \mid Y = 1, G = g)$$

By changing u_A and u_F, our framework can accommodate most metrics proposed in the literature

magnitude considerations

- with these definitions, we can formally discuss the existence of less discriminatory alternatives
  - is there any more accurate and more fair algorithm?
- not only the existence but also the magnitude of potential gains may matter
- Title VI legal manual by the Department of Justice writes:

"*investigating agencies must determine whether the disparity is large enough to matter, i.e., it is sufficiently significant to establish a legal violation.*"

- our framework can allow for such magnitude considerations

magnitude considerations

definition: fix any magnitude parameters $\Delta_r, \Delta_b, \Delta_f \in \mathbb{R}$.

algorithm a_1 constitutes a $(\Delta_r, \Delta_b, \Delta_f)$-improvement on a_0 if

- $U_A^r(a_1) > (1 + \Delta_r)U_A^r(a_0)$
- $U_A^b(a_1) > (1 + \Delta_b)U_A^b(a_0)$
- $|U_F^r(a_1) - U_F^b(a_1)| < (1 - \Delta_f)|U_F^r(a_0) - U_F^b(a_0)|$

accuracy for group r increases by Δ_r percent

accuracy for group b increases by Δ_b percent

disparity decreases by Δ_f percent

magnitude considerations

definition: fix any magnitude parameters $\Delta_r, \Delta_b, \Delta_f \in \mathbb{R}$.

algorithm a_1 constitutes a $(\Delta_r, \Delta_b, \Delta_f)$-improvement on a_0 if

- $U_A^r(a_1) > (1 + \Delta_r)U_A^r(a_0)$
- $U_A^b(a_1) > (1 + \Delta_b)U_A^b(a_0)$
- $|U_F^r(a_1) - U_F^b(a_1)| < (1 - \Delta_f)|U_F^r(a_0) - U_F^b(a_0)|$

- NB: $(0,0,0)$-improvement $\Leftrightarrow$ more accurate and more fair

what we want to evaluate

- our goal is to evaluate the improvability of **a status quo algorithm** a_0 within a given class $\mathscr{A}$ of algorithms
  - $\mathscr{A}$: **a class of permissible algorithms** (e.g., shape or capacity constraints)
- formally, we will test the following null hypothesis:

H_0: there is no algorithm within class $\mathscr{A}$ that $(\Delta_r, \Delta_b, \Delta_f)$-improves on a_0

model

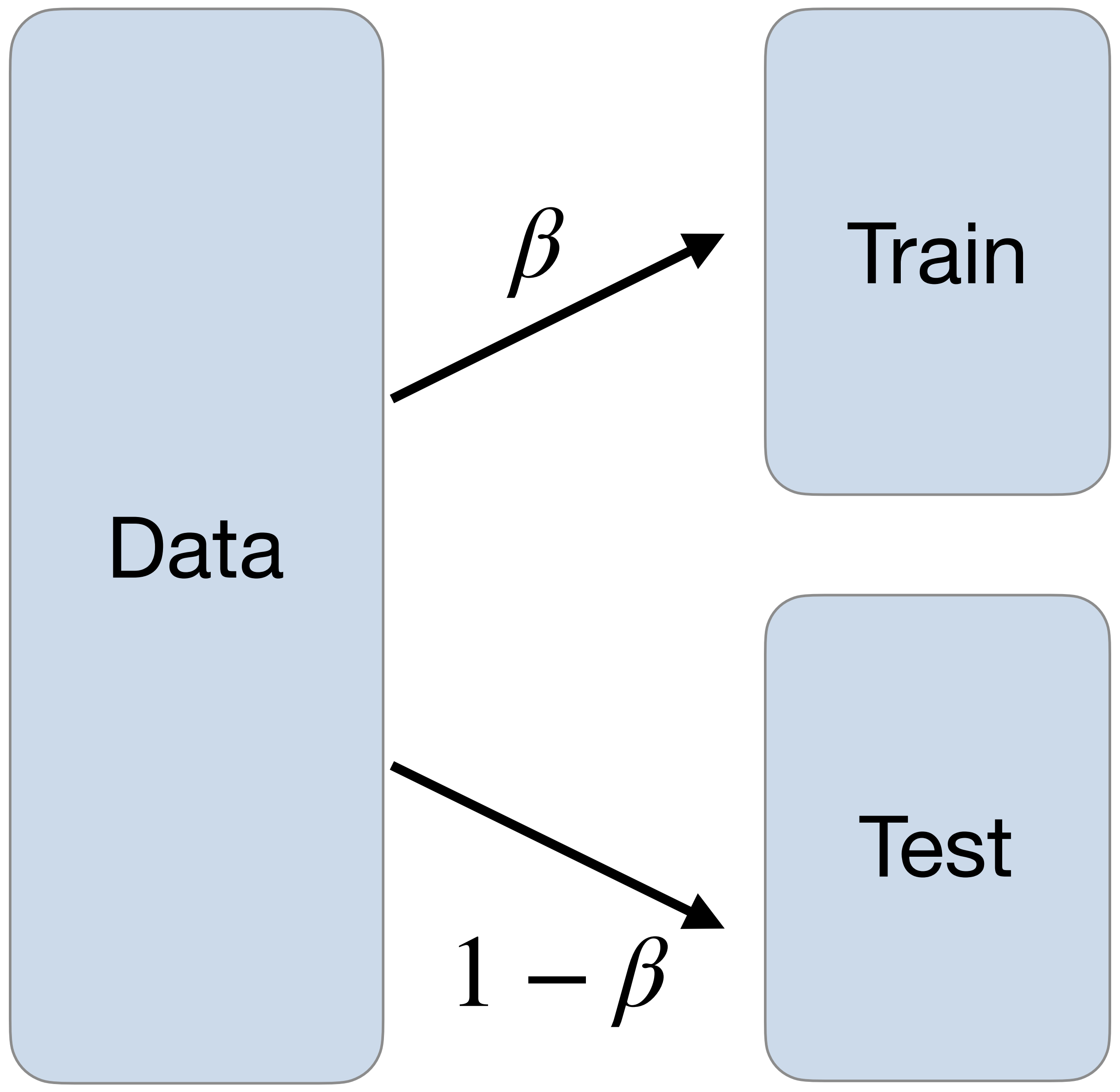
testing procedure

microfoundation

empirical application

our proposed procedure

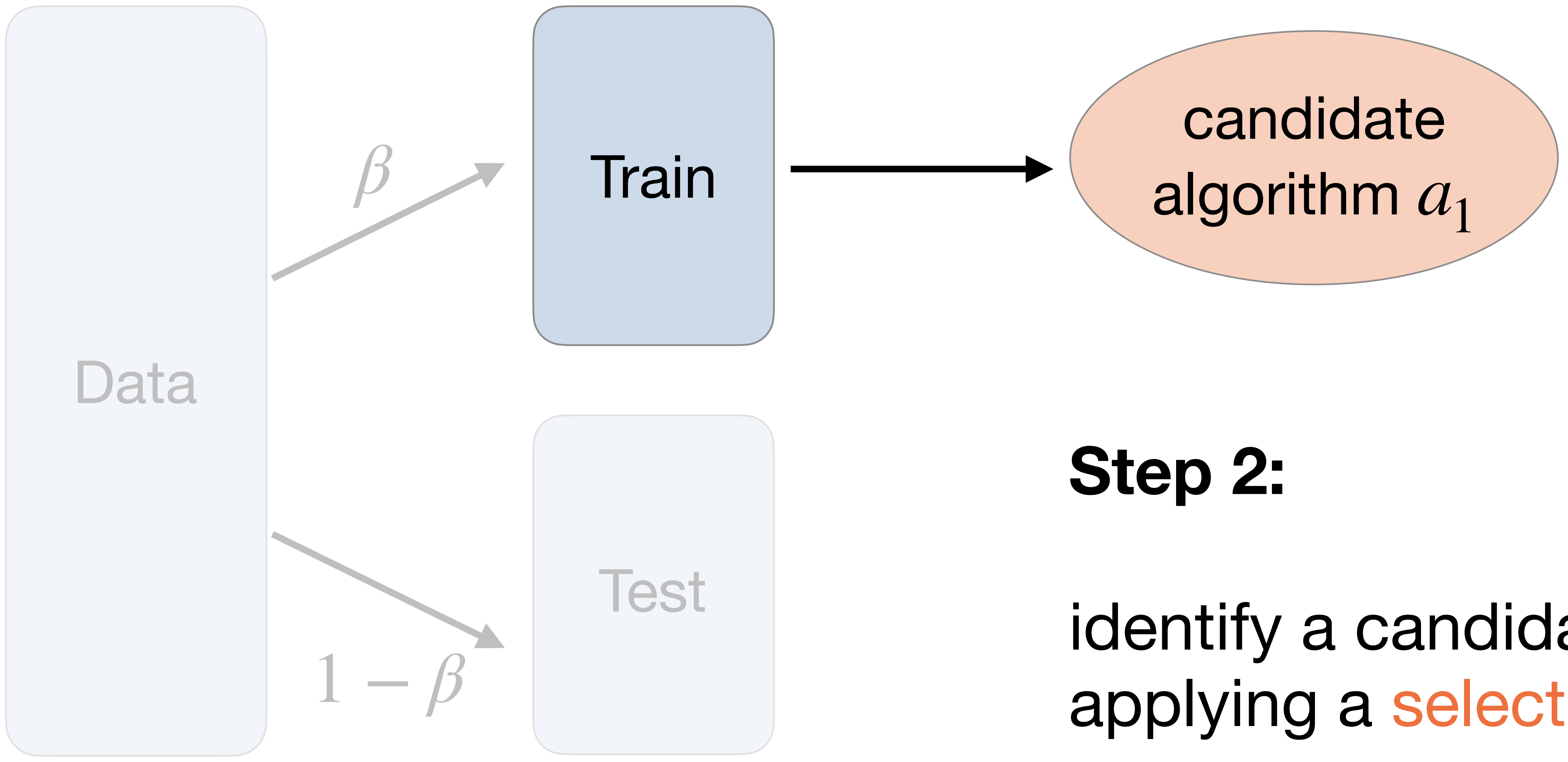
- the analyst does not know P,
  but has access to a dataset consisting of n i.i.d. observations $(Y_i, X_i, G_i)_{1 \leq i \leq n}$ from P



Step 1:

randomly split the data into train and test sets

our proposed procedure

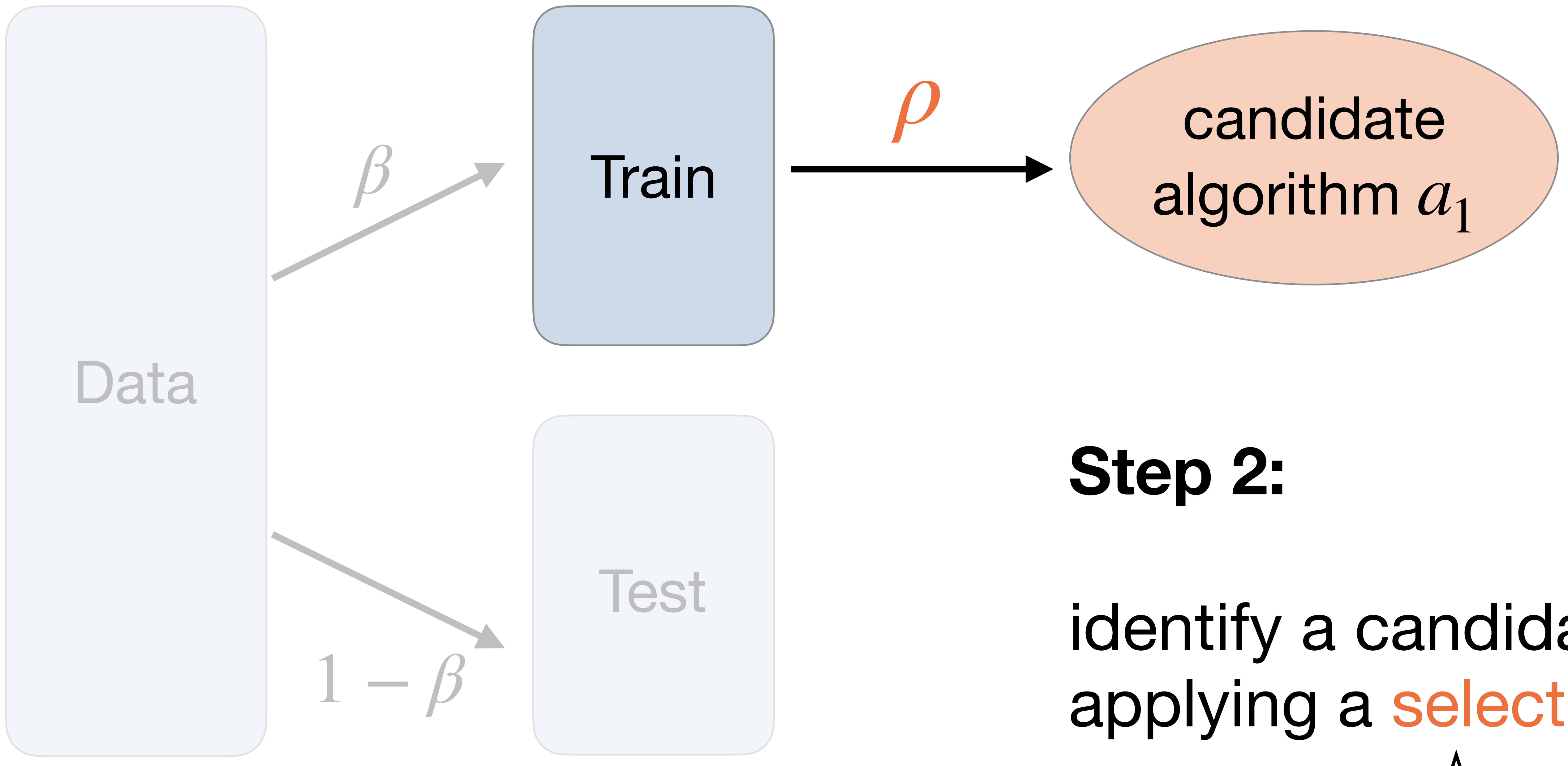


Step 2:

identify a candidate algorithm a_1
applying a selection rule to the training data

our proposed procedure

- general-purpose ML method (e.g., random forests)
- method proposed in algorithmic fairness literature

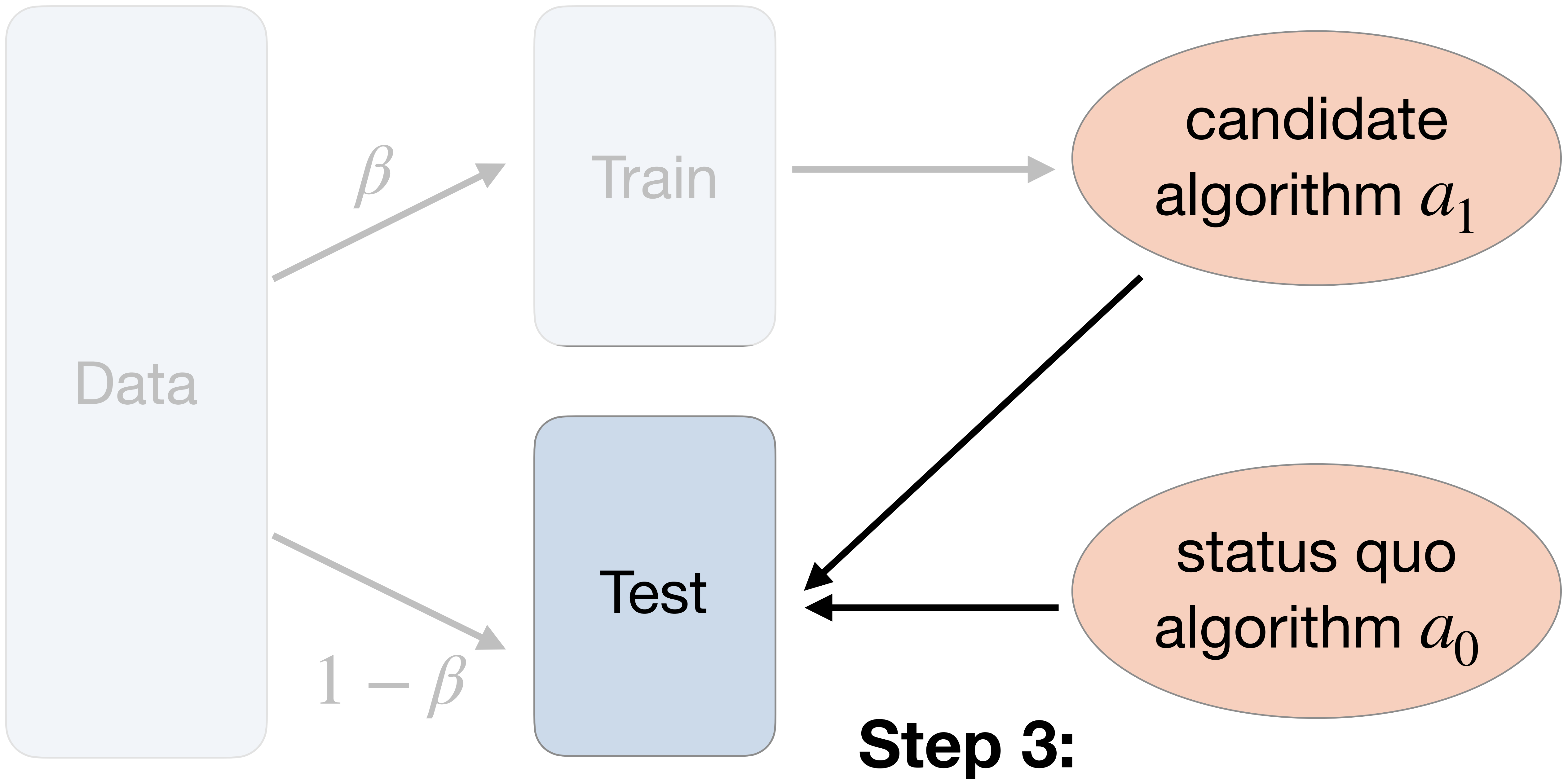


Step 2:

identify a candidate algorithm a_1
applying a selection rule to the training data

a mapping ρ from a training data into an algorithm

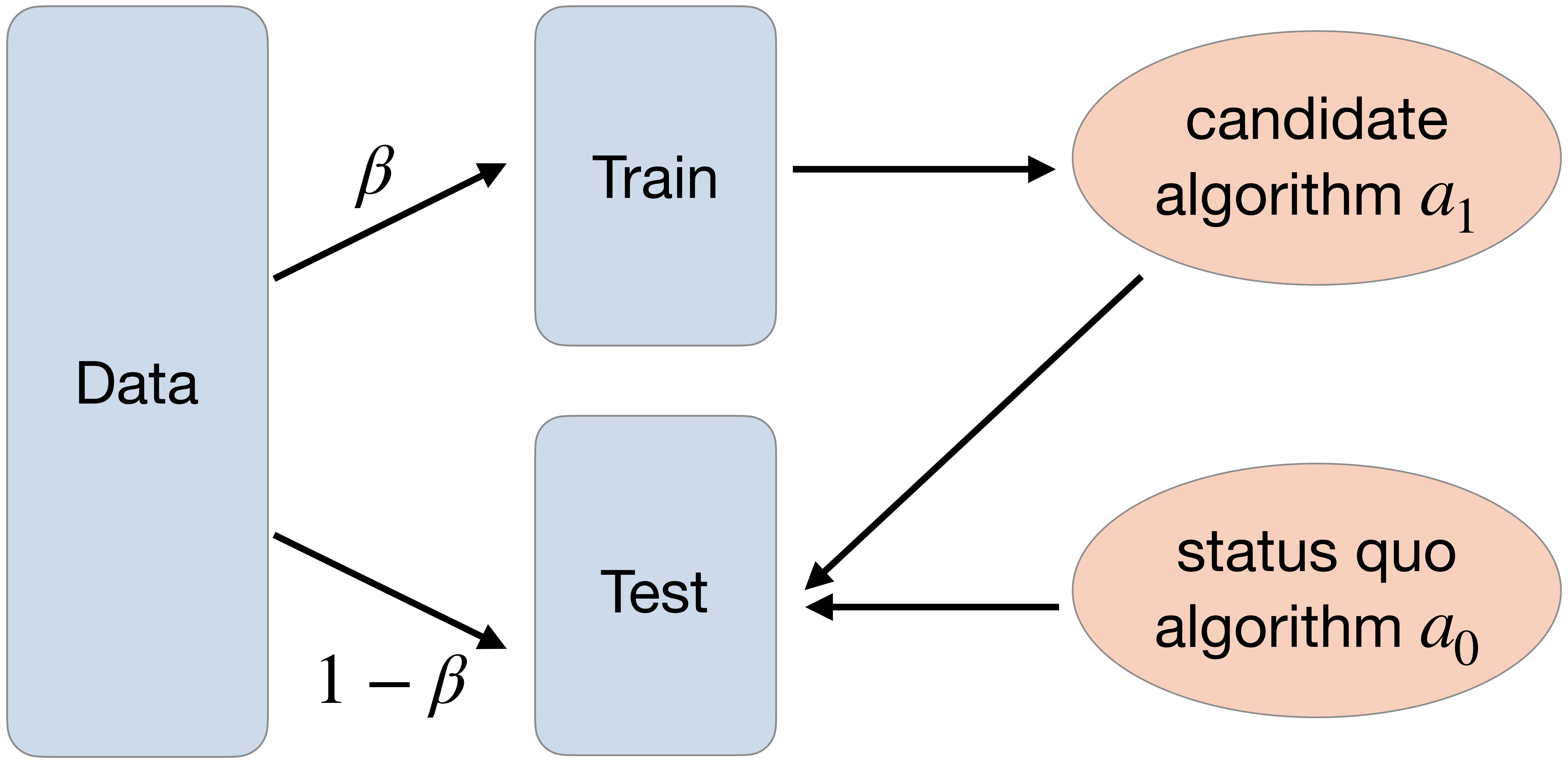
our proposed procedure



Step 3:

test whether a_1 $(\Delta_r, \Delta_b, \Delta_f)$-improves on a_0 computing a p-value (details come next)

our proposed procedure



Step 4:

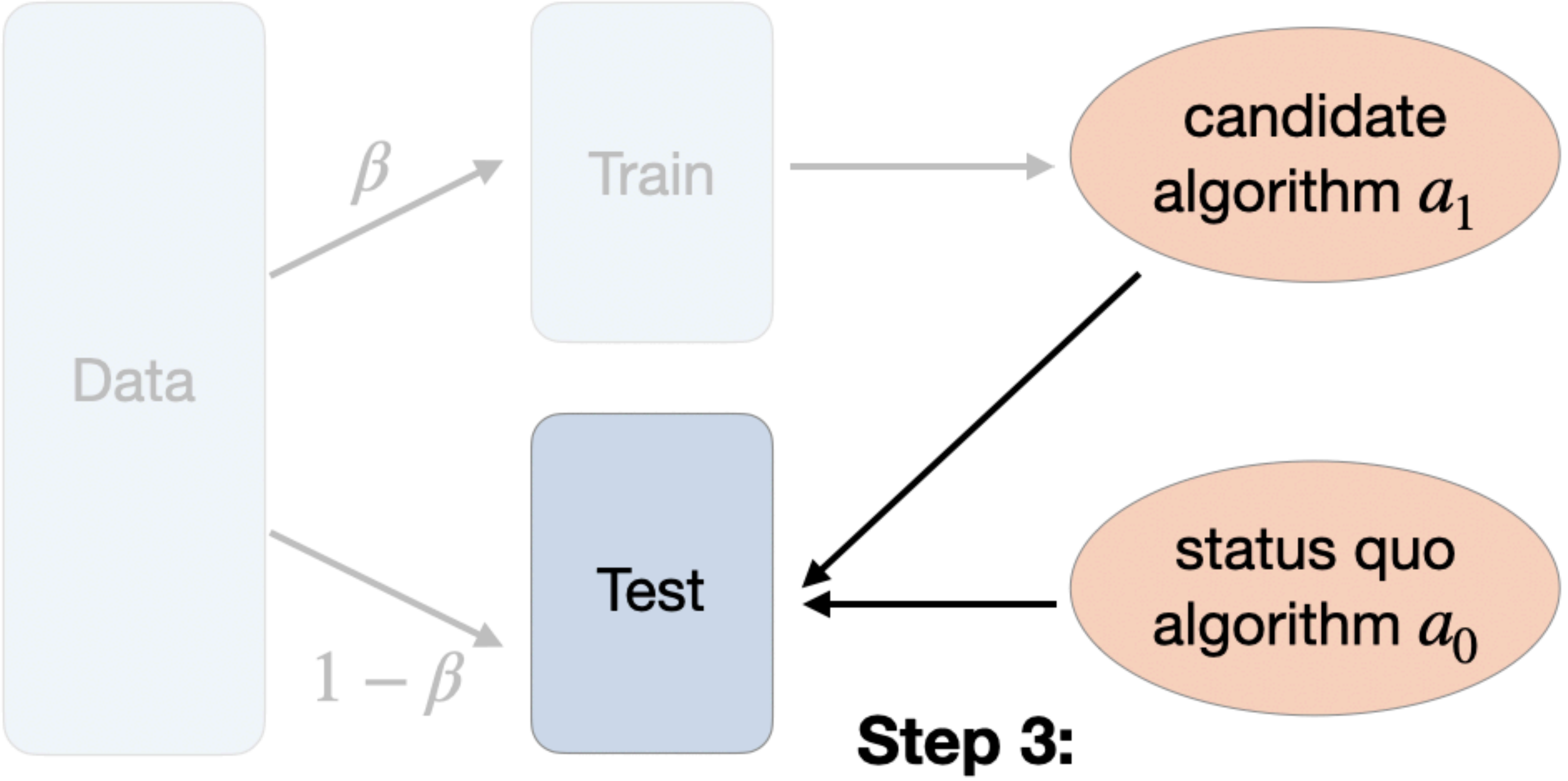
repeat steps 1-3 K times, and obtain p-values $(p_1, \ldots, p_K)$

aggregate the result by computing the **median p-value** $p := \text{median}\{p_1, \ldots, p_K\}$

and reject the null if $p < \dfrac{\alpha}{2}$

more details on step 3

- **step 3**: test whether a_1 is $(\Delta_r, \Delta_b, \Delta_f)$-improves on a_0



Step 3:

test whether a_1 $(\Delta_r, \Delta_b, \Delta_f)$-improves on a_0
computing a p-value

more details on step 3

assume $(\Delta_r, \Delta_b, \Delta_f) := (0,0,0)$
for simplicity

- **step 3**: test whether a_1 is more accurate and more fair than a_0

more details on step 3

assume $(\Delta_r, \Delta_b, \Delta_f) := (0,0,0)$ for simplicity

- **step 3**: test whether a_1 is more accurate and more fair than a_0

null hypothesis H_0

$$U_A^r(a_1) \leq U_A^r(a_0)$$

OR

$$U_A^b(a_1) \leq U_A^b(a_0)$$

OR

$$|\, U_F^r(a_1) - U_F^b(a_1) \,| \geq |\, U_F^r(a_0) - U_F^b(a_0) \,|$$

alternative H_1

$$U_A^r(a_1) > U_A^r(a_0)$$

AND

$$U_A^b(a_1) > U_A^b(a_0)$$

AND

$$|\, U_F^r(a_1) - U_F^b(a_1) \,| < |\, U_F^r(a_0) - U_F^b(a_0) \,|$$

a_1 is more accurate and more fair than a_0

more details on step 3

- **step 3**: test whether a_1 is more accurate and more fair than a_0

null hypothesis H_0

$$U_A^r(a_1) \leq U_A^r(a_0)$$

OR

$$U_A^b(a_1) \leq U_A^b(a_0)$$

OR

$$|U_F^r(a_1) - U_F^b(a_1)| \geq |U_F^r(a_0) - U_F^b(a_0)|$$

union of three conditions

alternative H_1

$$U_A^r(a_1) > U_A^r(a_0)$$

AND

$$U_A^b(a_1) > U_A^b(a_0)$$

AND

$$|U_F^r(a_1) - U_F^b(a_1)| < |U_F^r(a_0) - U_F^b(a_0)|$$

more details on step 3

- **step 3**: test whether a_1 is more accurate and more fair on a_0

null hypothesis H_0	**alternative** H_1	
$U_A^r(a_1) \leq U_A^r(a_0)$	$U_A^r(a_1) > U_A^r(a_0)$	p_k^r
OR	AND	
$U_A^b(a_1) \leq U_A^b(a_0)$	$U_A^b(a_1) > U_A^b(a_0)$	p_k^b
OR	AND	
$\lvert U_F^r(a_1) - U_F^b(a_1) \rvert \geq \lvert U_F^r(a_0) - U_F^b(a_0) \rvert$	$\lvert U_F^r(a_1) - U_F^b(a_1) \rvert < \lvert U_F^r(a_0) - U_F^b(a_0) \rvert$	p_k^f

we will construct a p-value for each part individually → combine these by taking the maximum (intersection-union method)

$$p_k := \max \left\{ p_k^r, p_k^b, p_k^f \right\}$$

construting p-value for a subhypothesis

null hypothesis H_0^r	alternative H_1^r	
$U_A^r(a_1) \leq U_A^r(a_0)$	$U_A^r(a_1) > U_A^r(a_0)$	p_k^r

- define $\widehat{U_A^r}(a)$ as the sample analog of $U_A^r(a)$; compute it using the test set
- define a test statistics $\hat{T}_{r,n} := \widehat{U_A^r}(a_1) - \widehat{U_A^r}(a_0)$

  expected to be negative if H_0^r is true
- generate a p-value p_k^r using the **nonparametric bootstrap**

bootstrap

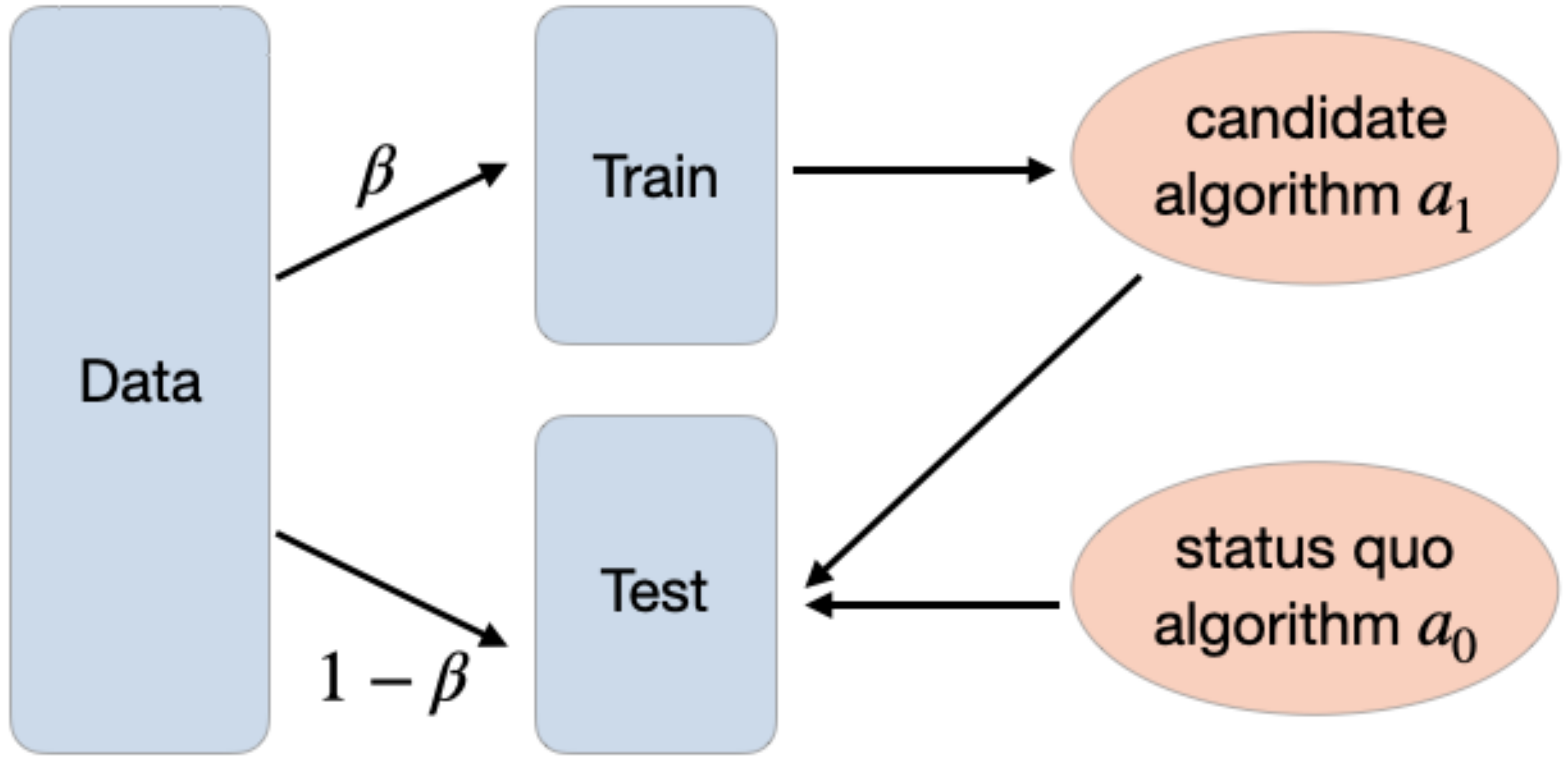
construting p-value for a subhypothesis

null hypothesis H_0^r	alternative H_1^r	
$U_A^r(a_1) \leq U_A^r(a_0)$	$U_A^r(a_1) > U_A^r(a_0)$	p_k^r

- define $\widehat{U_A^r}(a)$ as the sample analog of $U_A^r(a)$; compute it using the test set
- define a test statistics $\hat{T}_{r,n} := \widehat{U_A^r}(a_1) - \widehat{U_A^r}(a_0)$

  expected to be negative if H_0^r is true
- generate a p-value p_k^r using the **nonparametric bootstrap**
  - avoid analytically computing standard errors case-by-case for each utility function
- (p-values for other two parts are defined similarly)

three practical objectives



1. the appropriate definitions of disparate impact and business-relevant criteria vary substantially across applications
   - we want a framework that is flexible enough to accommodate any such definitions that may emerge in practice

2. there are often exogenous constraints on the algorithm space (e.g., capacity constraints, monotonicity in some variable, linearity)
   - we want a procedure that accommodates any such constraints

3. transparency and simplicity of use for practitioners

guarantees for this procedure (informal)

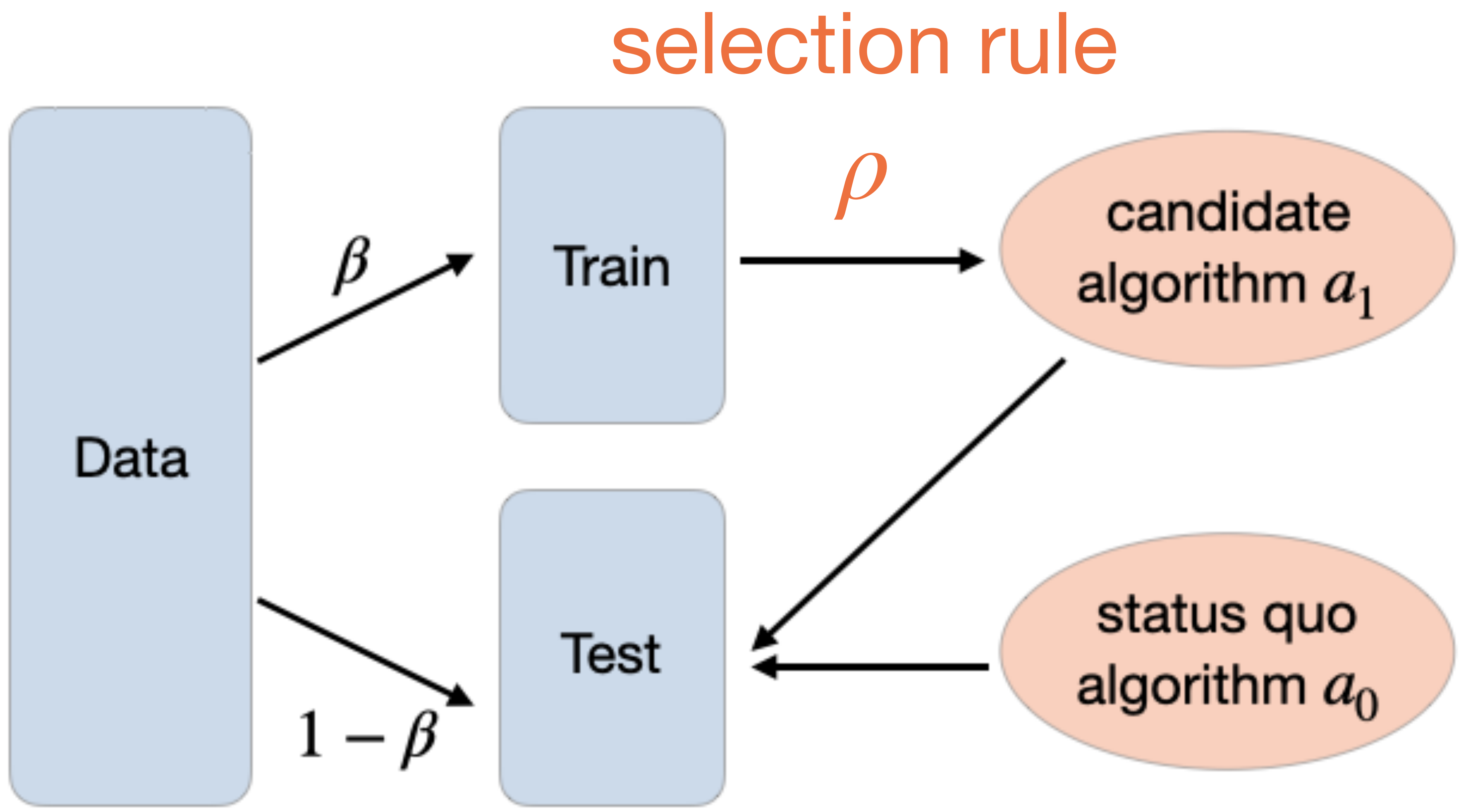
recall the null hypothesis:

H_0: there is no algorithm within class $\mathscr{A}$ that $(\Delta_r, \Delta_b, \Delta_f)$-improves on a_0

- under regularity conditions, our test is **asymptotically valid**
  - **valid**: if the null is true, then we can control the probability of incorrect rejection

- when the selection rule is "improvement-convergent," then the test is **consistent**
  - **consistent**: if the null is false, we can correctly reject it
    with probability converging to 1 as the sample grows large

guarantees for this procedure (informal)



- when the selection rule is "improvement-convergent," then the test is **consistent**
  - **consistent**: if the null is false, we can correctly reject it with probability converging to 1 as the sample grows large
  - improvement-convergent: the selection rule can find a better candidate when the sample size is large and a_0 is improvable within class $\mathscr{A}$
  - NB: validity does not require improvement-convergence

comments

- the procedure tests the existence of an alternative that achieves improvement
- strictly speaking, the procedure does not identify a specific alternative
- however, if we reject the null, our procedure implies that the used selection rule can find a better alternative
  - if we need a single algorithm to use after rejecting the null, we recommend applying the selection rule to the entire dataset and using its output

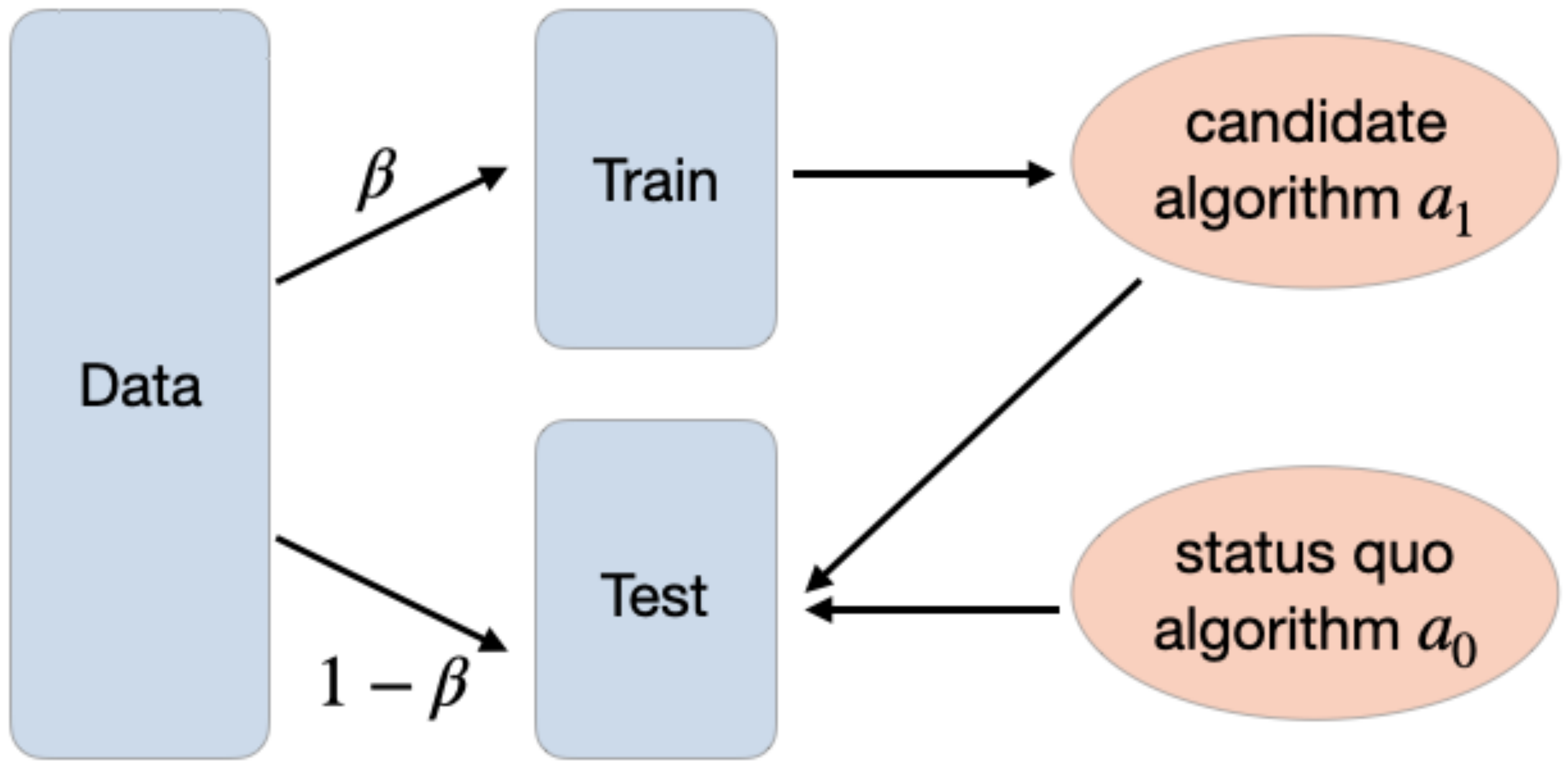
model

testing procedure

microfoundation

empirical application

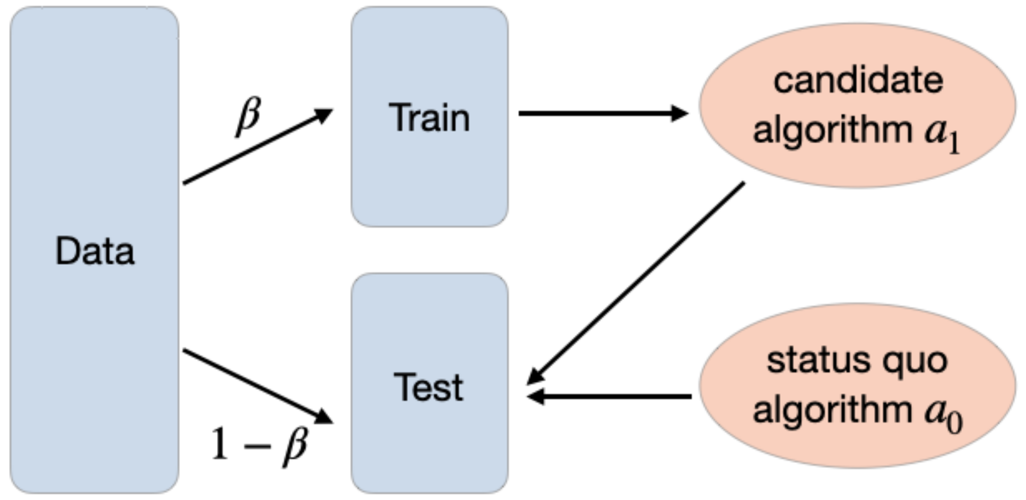
microfoundation for repeated sample-splitting



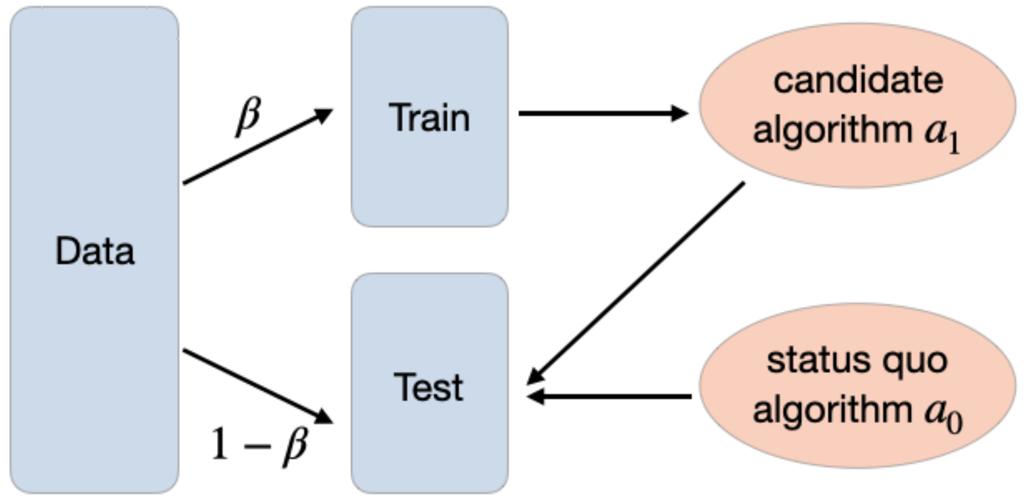
- we recommend using the median p-value across $K \geq 2$ train-test splits
- why not just conduct a test with a single train-test split ($K := 1$)?
  - both valid
  - no known statistical advantage (e.g., power) for repeated sample splitting
  - ...then why?

microfoundation for repeated sample-splitting

- resulting p-value can vary substantially across different splits

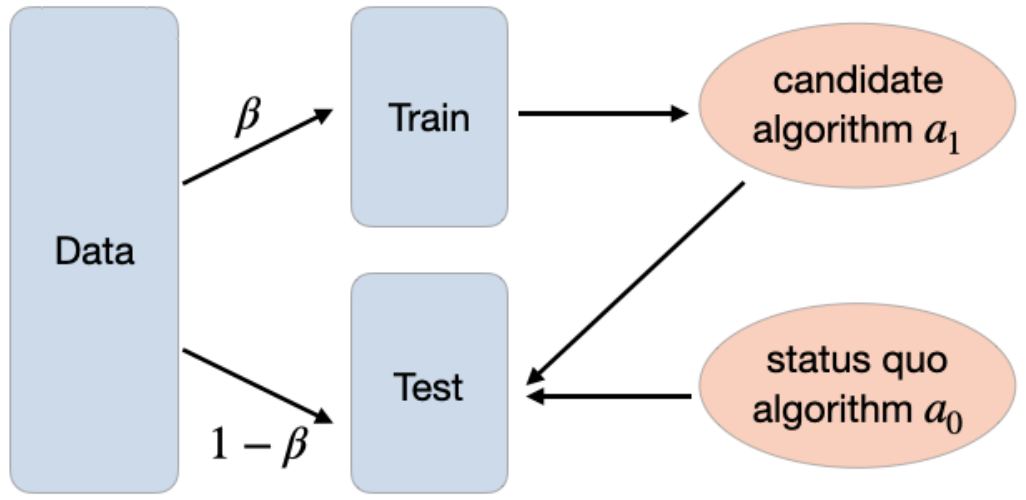


$$p^{(1)} = 0.08$$



$$p^{(2)} = 0.11$$

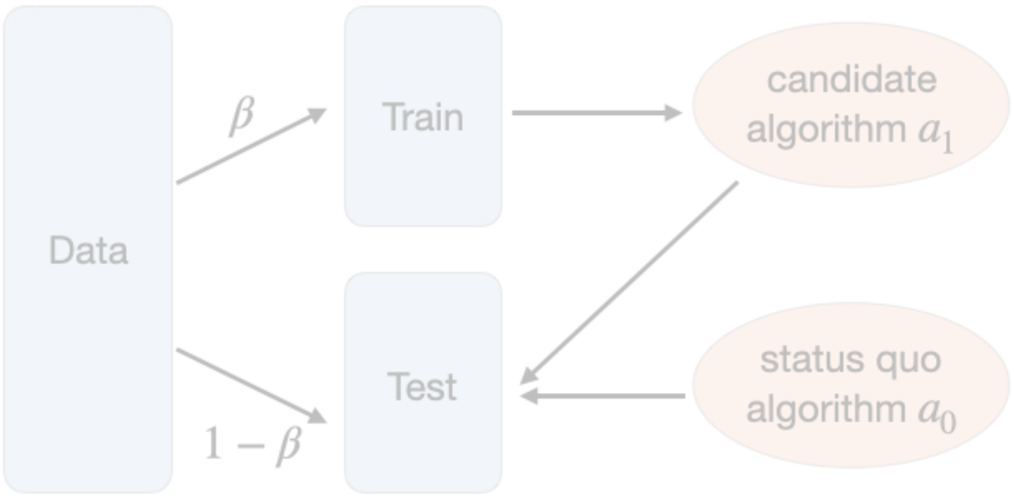
...



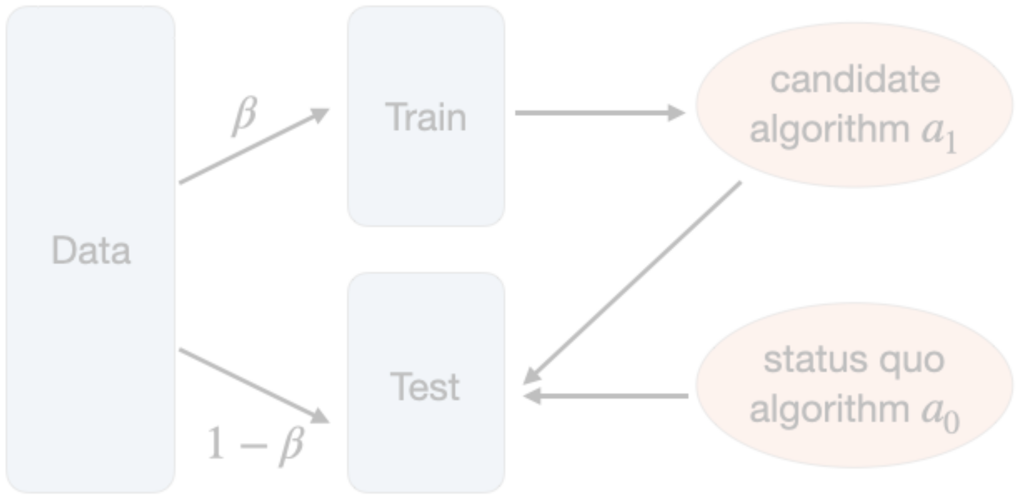
$$p^{(100)} := 0.04$$

microfoundation for repeated sample-splitting

- resulting p-value can vary substantially across different splits

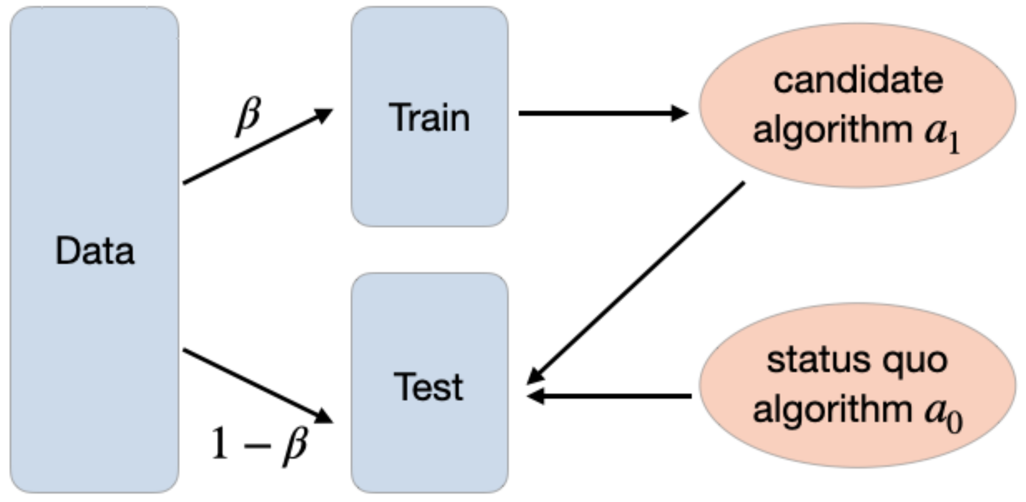


$p^{(1)} = 0.08$



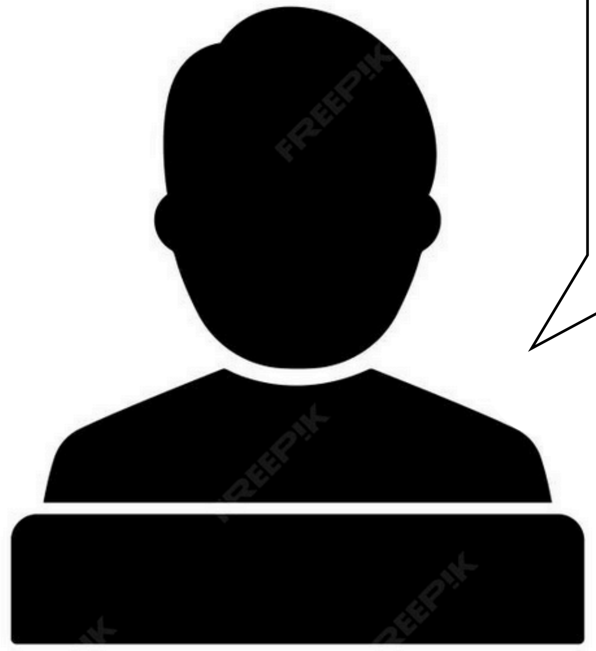
$p^{(2)} = 0.11$

…



$p^{(100)} := 0.04$

this is the split I used
we can reject the null with $\alpha := 0.05$



analyst

microfoundation for repeated sample-splitting

- resulting p-value can vary substantially across different splits
- relying on a single split introduces the possibility of manipulation by the analyst
- Ritzwoller and Romano (2023) put:

"Researchers are incentivized to report significant results.
If there is scope to materially alter the statistics that they report
through the choice of the split of their sample,
should this choice be left to chance?"

microfoundation for repeated sample-splitting

- how can we address this cherry-picking problem?
- our naive intuition says:

repeated sample-splitting reduces the sensitivity to the choice of splits, and provides stronger safeguards against manipulation

formalize this intuition!

setup

- game played by two players: an **analyst** and a **policymaker**

- there is **a fixed statistical test** of exact size α
  - the test produces a p-value given train-test split (e.g., step 1-3 of our test)

- policymaker first chooses between two procedures
  1. single train-test split: reject the null if $p < \alpha$
  2. K train-test splits (our proposed method): reject if $p < \alpha/2$

- analyst must follow the chosen procedure, and
  - repeats it m times at a cost of $c_\ell(m)$ for procedure $\ell \in \{1,2\}$

e.g. constant cost C per repetition

setup

- game played by two players: an **analyst** and a **policymaker**
- there is **a fixed statistical test** of exact size α
  - the test produces a p-value given train-test split (e.g., step 1-3 of our test)
- policymaker first chooses between two procedures
  1. single train-test split: reject the null if $p < \alpha$
  2. K train-test splits (our proposed method): reject if $p < \alpha/2$
- analyst must follow the chosen procedure, and
  - repeats it m times at a cost of $c_\ell(m)$ for procedure $\ell \in \{1,2\}$

increasing, weakly convex

setup

- game played by two players: an **analyst** and a **policymaker**

- there is **a fixed statistical test** of exact size α
  - the test produces a p-value given train-test split (e.g., step 1-3 of our test)

- policymaker first chooses between two procedures
  1. single train-test split: reject the null if $p < \alpha$
  2. K train-test splits (our proposed method): reject if $p < \alpha/2$

- analyst must follow the chosen procedure, and
  - repeats it m times at a cost of $c_\ell(m)$ for procedure $\ell \in \{1,2\}$
  - reports the p-value from one of these repetitions

- the reported p-value determines whether the null is rejected **(as if $m = 1$)**

setup

- we are interested in settings where the analyst wants to reject the null even when it holds

status quo is not improvable

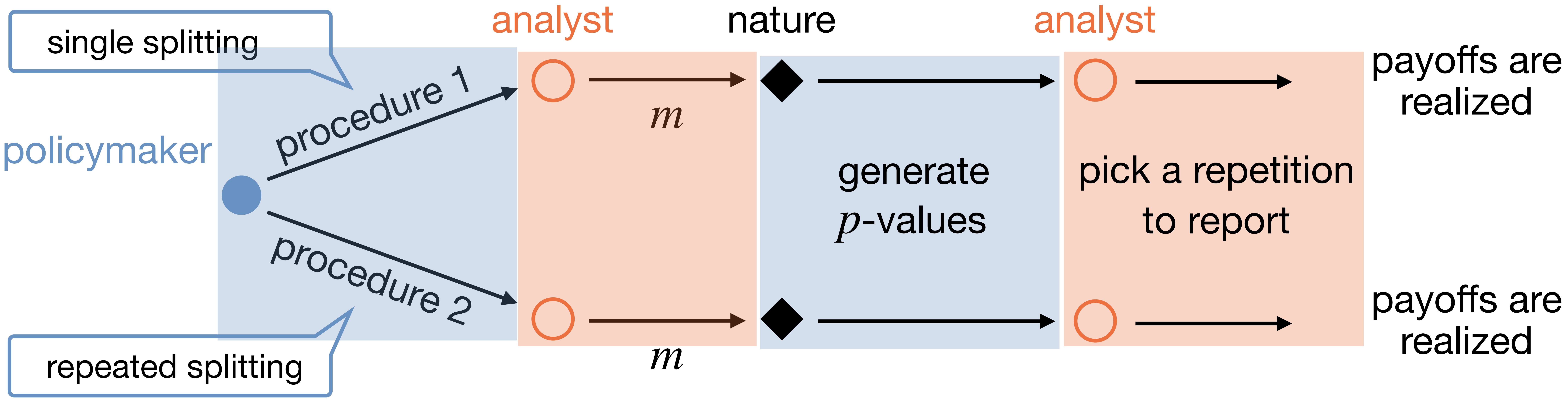
- we condition on the state of the world in which the null hypothesis holds

player \ action	reject	not reject
analyst	$1 - c_\ell(m)$	$-c_\ell(m)$
policymaker	0	1

analyst wants to reject

policymaker does not want to incorrectly reject

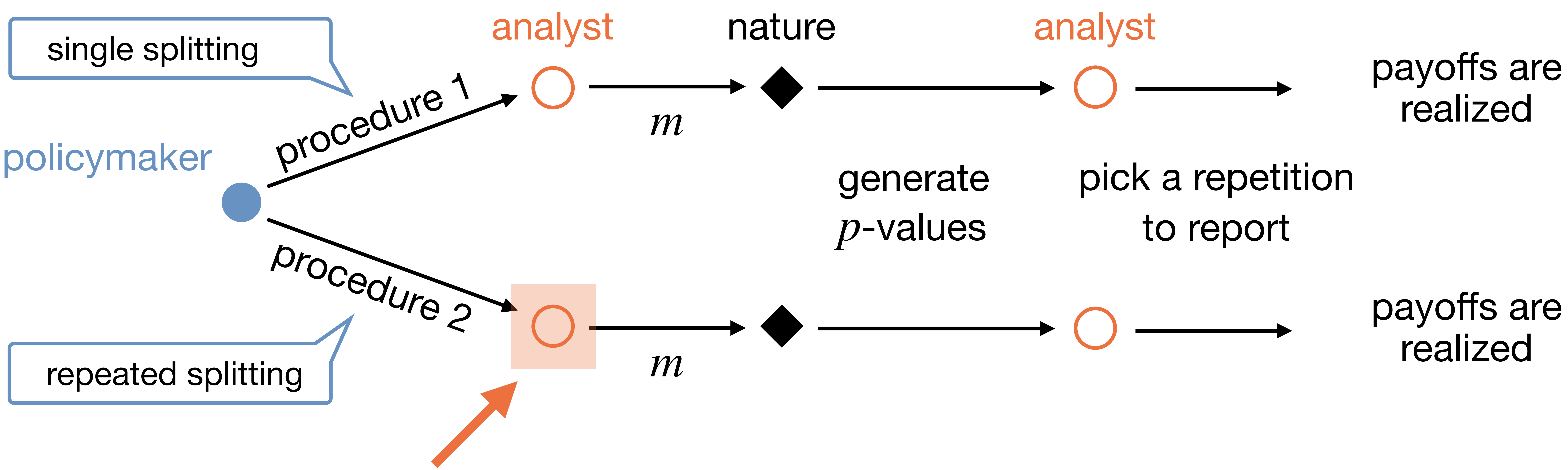
game tree



we consider subgame-perfect equilibria

backward induction: analyst's problem

if policymaker chooses **procedure 2 (K repeated sample splitting)**



backward induction: analyst's problem

if policymaker chooses **procedure 2 (K repeated sample splitting)**

- suppose that the analyst chooses # of repetition m
- $m \times K$ p-values are generated:

not i.i.d. (positively correlated)

$$\text{repetition } 1 \le i \le m \left\{ \underbrace{\begin{bmatrix} p_1^1 & p_1^2 & \dots & p_1^{K-1} & p_1^K \\ p_2^1 & p_2^2 & \dots & p_2^{K-1} & p_2^K \\ & & \vdots & & \\ p_{m-1}^1 & p_{m-1}^2 & \dots & p_{m-1}^{K-1} & p_{m-1}^K \\ p_m^1 & p_m^2 & \dots & p_m^{K-1} & p_m^K \end{bmatrix}}_{\text{split } 1 \le k \le K} \right.$$

for each repetition,
K p-values are generated

backward induction: analyst's problem

if policymaker chooses **procedure 2 (K repeated sample splitting)**

- suppose that the analyst chooses # of repetition m
- $m \times K$ p-values are generated:

median p-values

$$
\text{repetition } 1 \leq i \leq m \left\{ \underbrace{\begin{bmatrix} p_1^1 & p_1^2 & \cdots & p_1^{K-1} & p_1^K \\ p_2^1 & p_2^2 & \cdots & p_2^{K-1} & p_2^K \\ & & \vdots & & \\ p_{m-1}^1 & p_{m-1}^2 & \cdots & p_{m-1}^{K-1} & p_{m-1}^K \\ p_m^1 & p_m^2 & \cdots & p_m^{K-1} & p_m^K \end{bmatrix}}_{\text{split } 1 \leq k \leq K} \right. \begin{array}{l} \longrightarrow p_1 \\ \longrightarrow p_2 \\ \\ \longrightarrow p_{K-1} \\ \longrightarrow p_K \end{array}
$$

if any of them
is less than $\alpha/2$,
then reject the null

backward induction: analyst's problem

if policymaker chooses **procedure 2 (K repeated sample splitting)**

- suppose that the analyst chooses # of repetition m
- $m \times K$ p-values are generated:

median p-values

$$
\text{repetition } 1 \le i \le m \left\{
\begin{bmatrix}
p_1^1 & p_1^2 & \dots & p_1^{K-1} & p_1^K \\
p_2^1 & p_2^2 & \dots & p_2^{K-1} & p_2^K \\
 & & \vdots & & \\
p_{m-1}^1 & p_{m-1}^2 & \dots & p_{m-1}^{K-1} & p_{m-1}^K \\
p_m^1 & p_m^2 & \dots & p_m^{K-1} & p_m^K
\end{bmatrix}
\right.
\begin{array}{c}
\longrightarrow p_1 \\
\longrightarrow p_2 \\
\\
\longrightarrow p_{K-1} \\
\longrightarrow p_K
\end{array}
$$

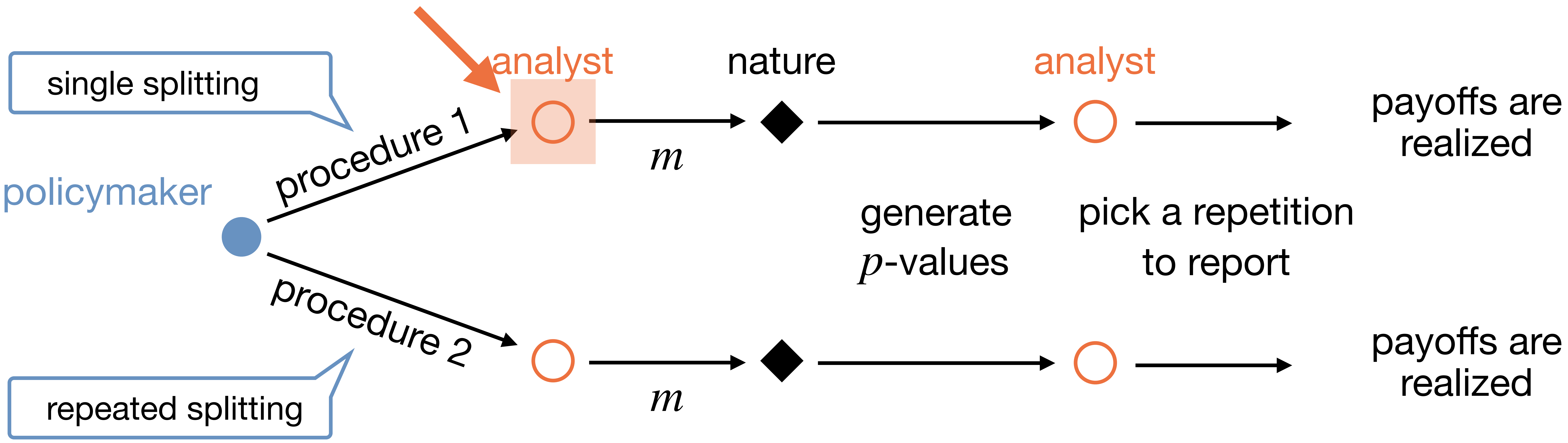
if any of them is less than $\alpha/2$, then reject the null

- analyst chooses m to maximize

$$P(\text{rejection} \mid m, \text{procedure 2}) - c_2(m)$$

backward induction: analyst's problem

if policymaker chooses **procedure 1 (single sample splitting)**



backward induction: analyst's problem

if policymaker chooses **procedure 1 (single sample splitting)**

- suppose that the analyst chooses # of repetition m
- $m \times 1$ p-values are generated:

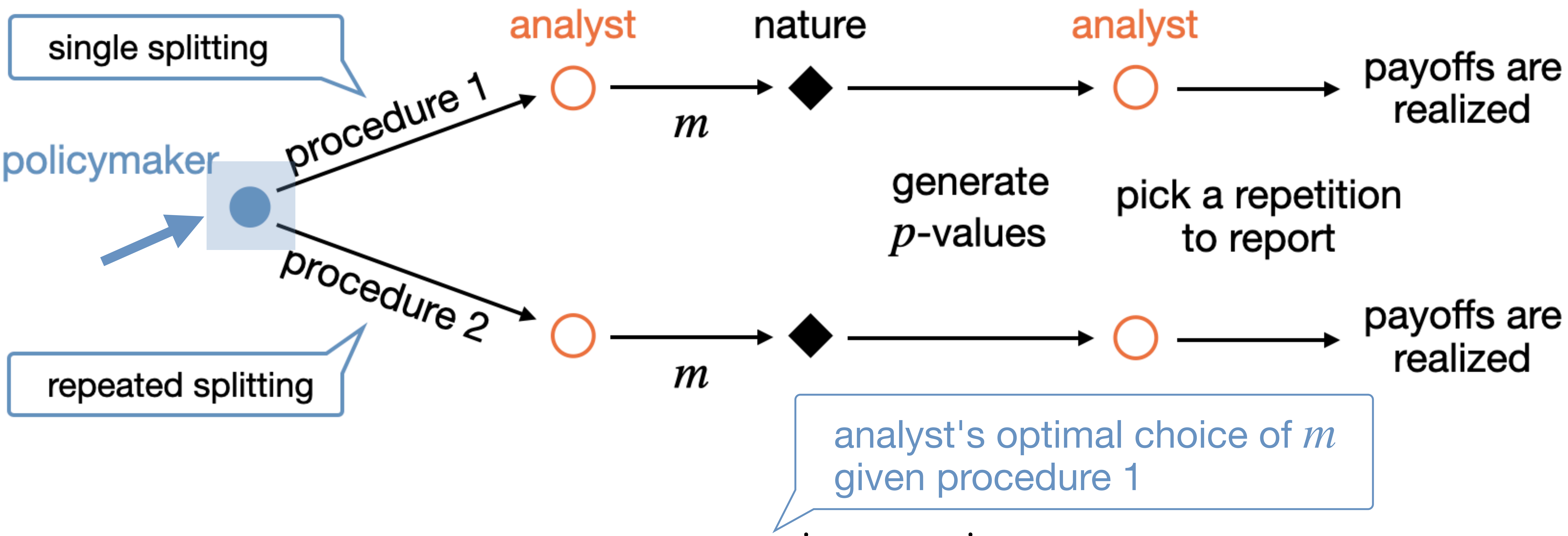
$$\text{repetition } 1 \leq i \leq m \left\{ \begin{bmatrix} p_1^1 & p_1^2 & \cdots & p_1^{K-1} & p_1^K \\ p_2^1 & p_2^2 & \cdots & p_2^{K-1} & p_2^K \\ & & \vdots & & \\ p_{m-1}^1 & p_{m-1}^2 & \cdots & p_{m-1}^{K-1} & p_{m-1}^K \\ p_m^1 & p_m^2 & \cdots & p_m^{K-1} & p_m^K \end{bmatrix} \right.$$

if any of them is less than α, then reject the null

- analyst chooses m to maximize

$$P(\text{rejection} \mid m, \text{procedure 1}) - c_1(m)$$

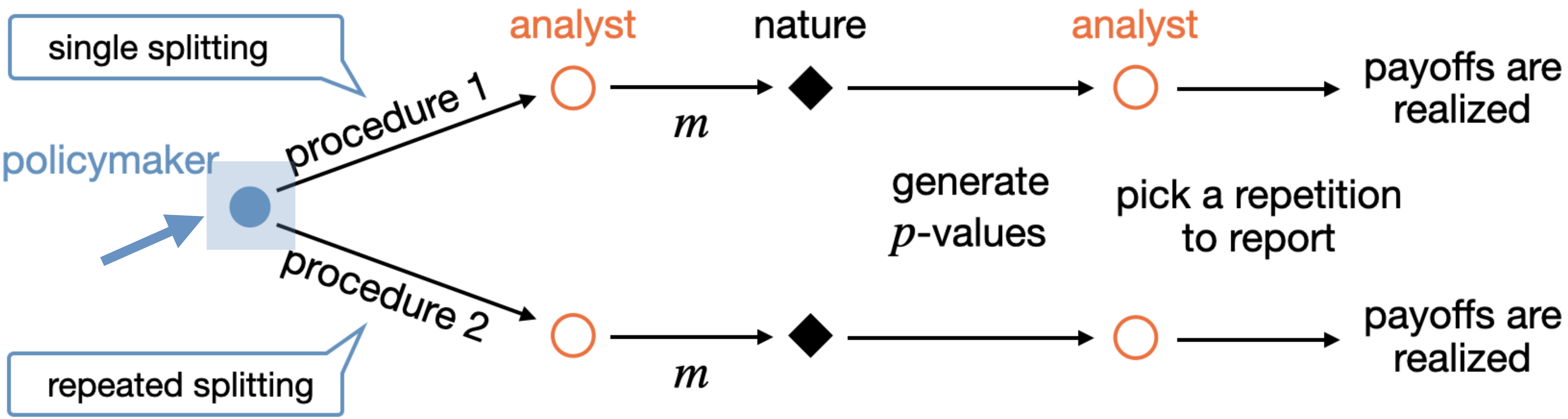
backward induction: policymaker's problem



given these analyst's best responses m_1^* and m_2^*,

policymaker chooses the procedure that minimizes the probability that the null is incorrectly rejected

backward induction: policymaker's problem



definition: procedure ℓ is more robust to manipulation than procedure $\ell' \neq \ell$ if the probability of incorrect rejection is lower for ℓ in equilibrium

result (informal)

- p-values are not perfectly correlated
- cost of an extra sample splitting is not large

under a mild assumption, for K sufficiently large,
procedure 2 (repeated sample splitting) is more robust to manipulation
than procedure 1 (single sample splitting)

proof idea

- why would we expect the result to be true? -- **concentration of the median**
  - to cherry-pick given a single sample split, analyst just needs to reject under one split
  - to cherry-pick given K sample splits, analyst needs to reject under at least half of them
    - # of rejections is "almost deterministic" if p-values are i.i.d. across random splits
    - leaving little room for manipulation

- formalizing this is not straightforward because p-values are NOT i.i.d.
  - they are positively correlated. can't use the most standard concentration inequalities

- however, note that p-values are exchangeable; we can leverage de Finetti's theorem to show the result

model

testing procedure

microfoundation

empirical application

empirical application

- we consider a dataset from Obermeyer et al. (2019)
  - X: patient's medical profile
  - G: race (Black or White)
  - Y: the number of active chronic illnesses in the next year
  - D: whether to automatically enroll the patient in a care management program

- status quo algorithm a_0: the hospital's algorithm (assign 3% of patients to care)

- we apply our approach to evaluate the improvability of this algorithm within the class of algorithms $a : \mathscr{X} \to \{0,1\}$ that also enrolls 3% of patients
  - class of permissible algorithms $\mathscr{A}$ is restricted by capacity constraint

accuracy and fairness

- similar to Obermeyer et al. (2019), let

$$U_A^g(a) = U_F^g(a) := E[Y \mid a(X) = 1, G = g]$$

expected number of illnesses for patients in group g
who are assigned to the program

- an algorithm is:
  - **more accurate** if the expected number of health conditions is higher among both Black and White patients assigned to the program

high $U^g(a)$: the algorithm successfully identifies sick patients
who are likely to derive greater benefits from the care

accuracy and fairness

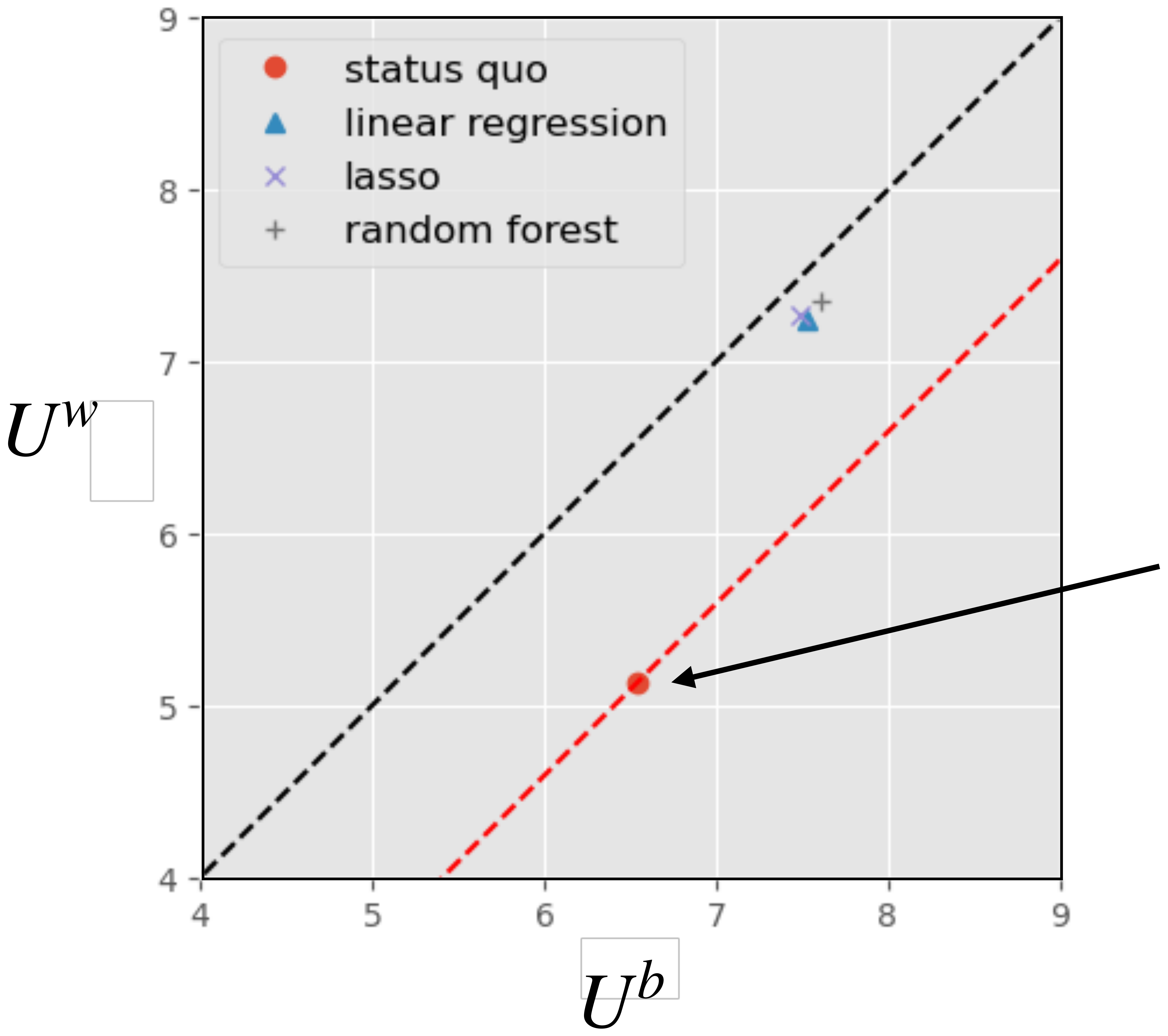
- similar to Obermeyer et al. (2019), let

$$U_A^g(a) = U_F^g(a) := E[Y \mid a(X) = 1, G = g]$$

> expected number of illnesses for patients in group g who are assigned to the program

- an algorithm is:
  - **more accurate** if the expected number of health conditions is higher among both Black and White patients assigned to the program
  - **more fair** if it reduces the disparity in the expected number of health conditions among Black and White patients assigned to the program

status quo algorithm

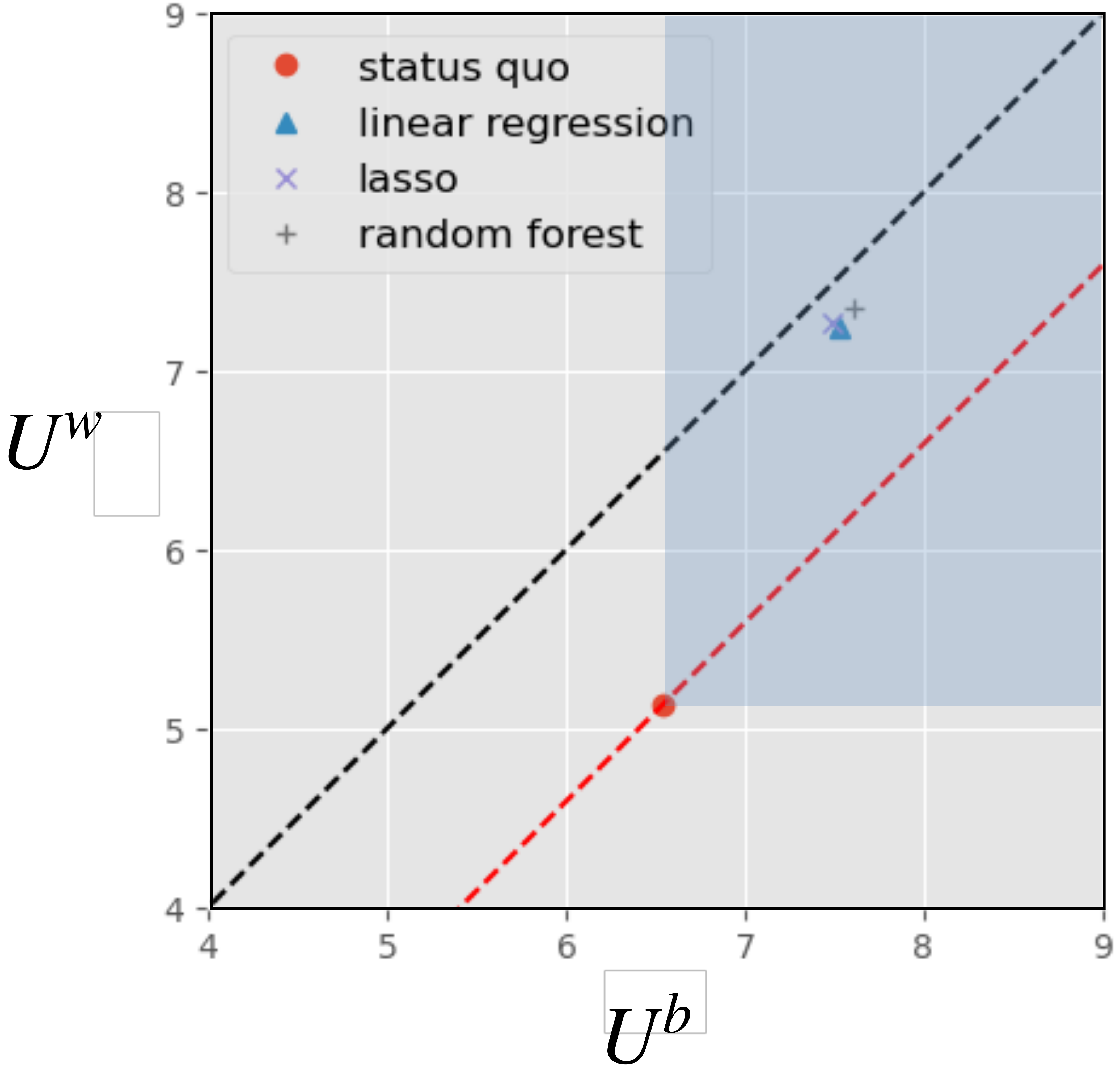


(average of $K := 7$ repetitions)

$U^b > U^w$: Black patients need to have more illnesses to enroll in care program

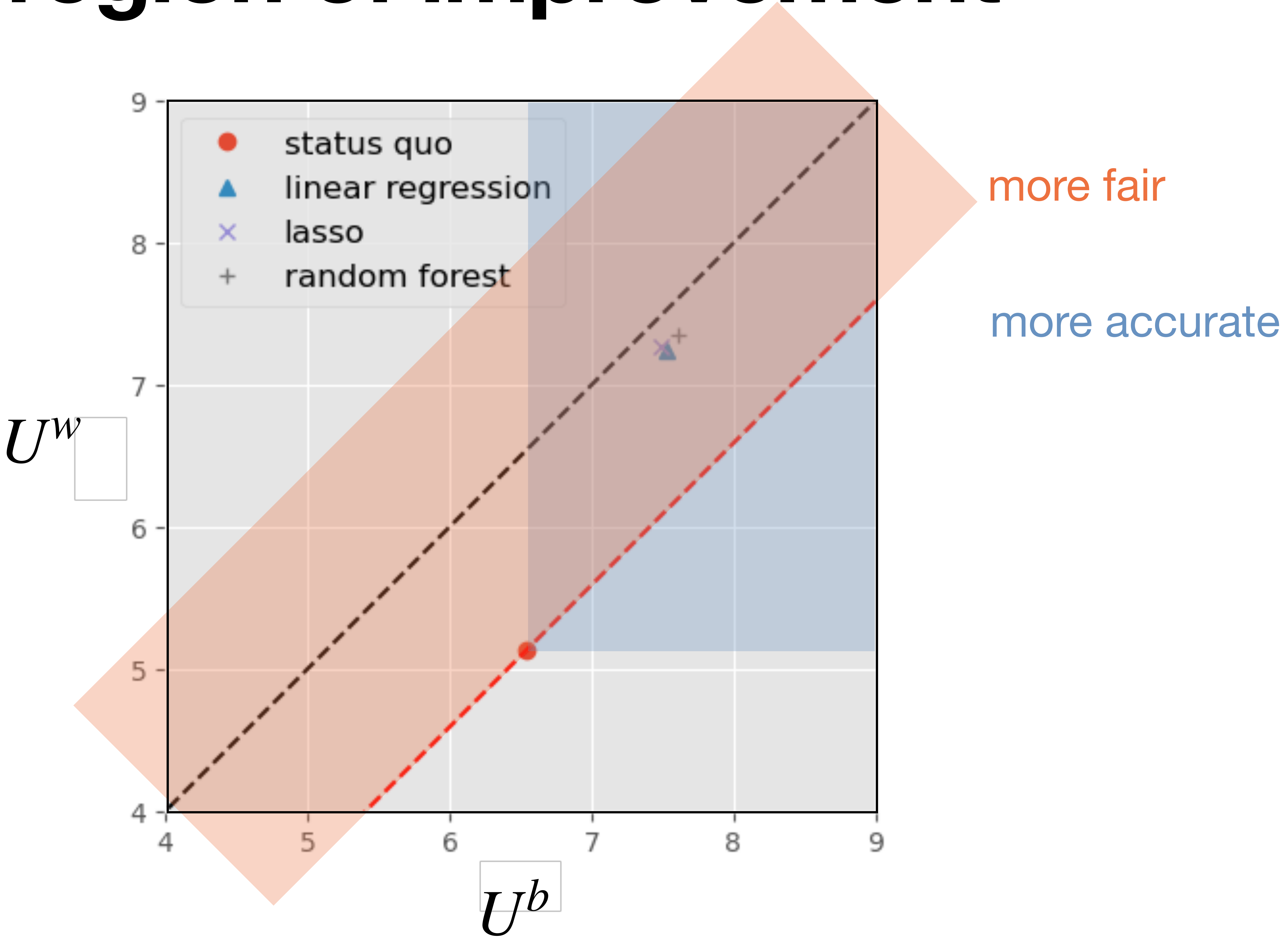
status quo algorithm favors White patients

region of improvement

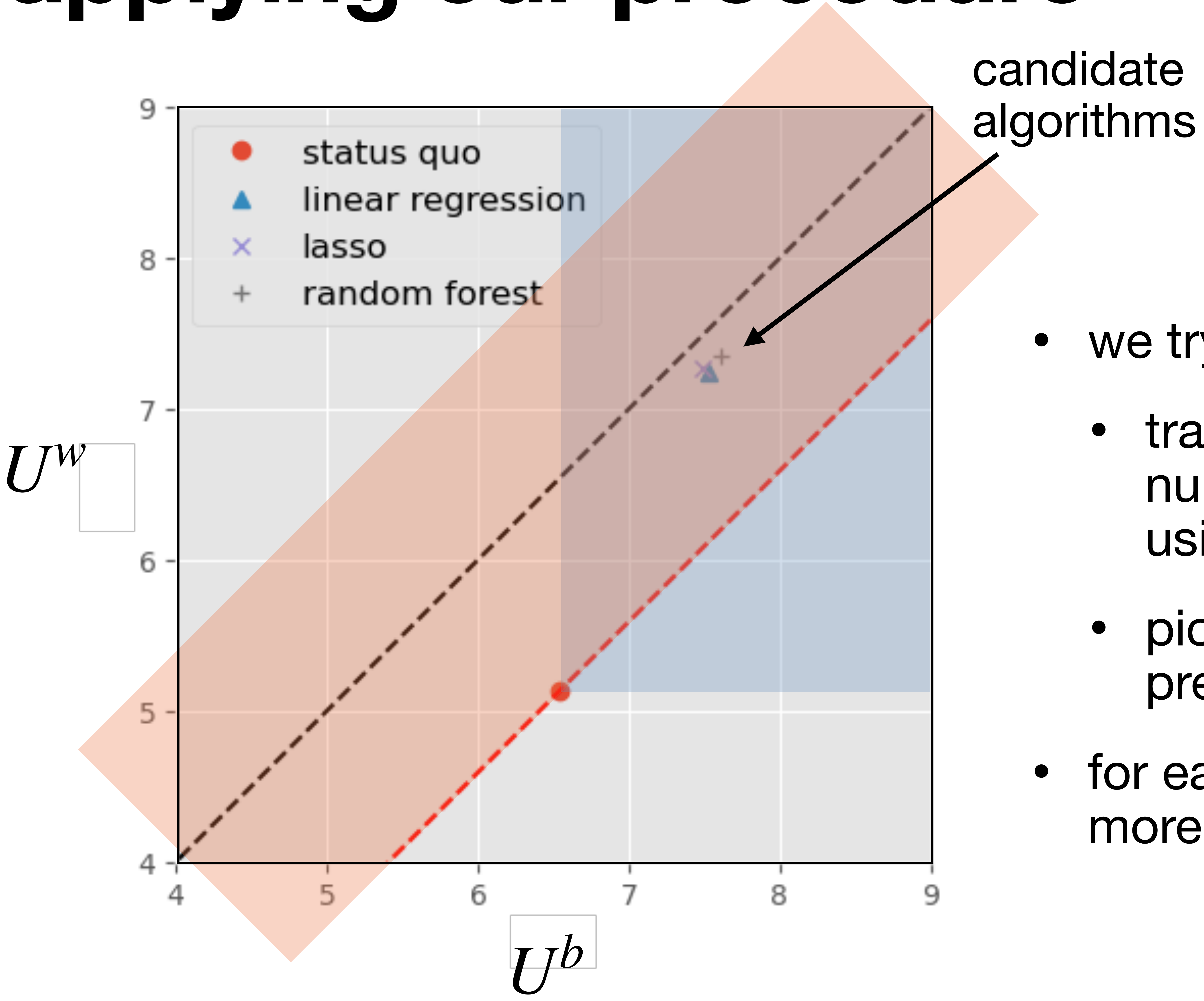


more accurate

region of improvement

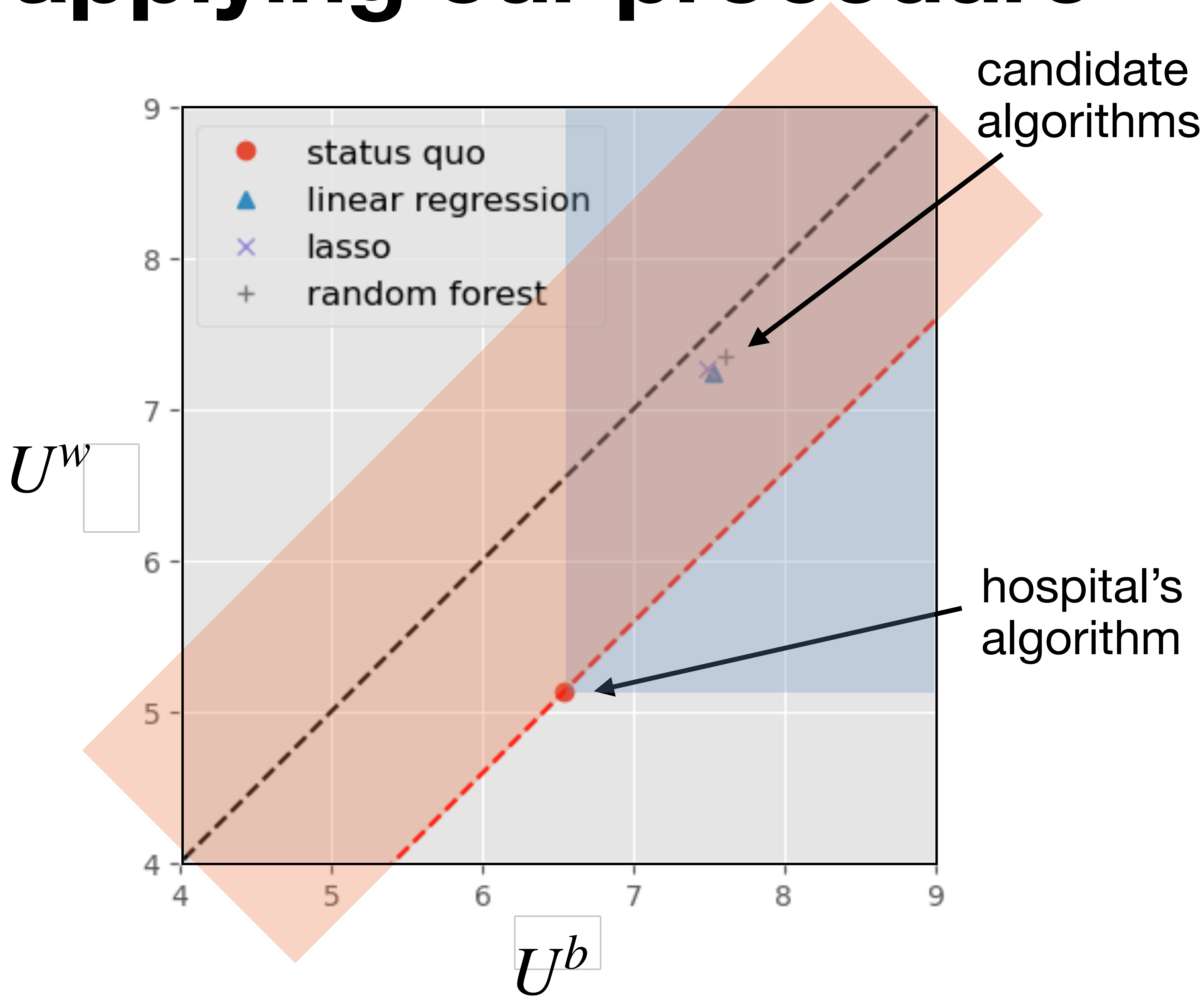


applying our procedure



- we try three selection rules
  - train the model to predict the expected number of illnesses
    using covariates without race
  - pick 3% of the population with the highest predicted scores
- for each selection rule, test the existence of more accurate and more fair alternatives

applying our procedure



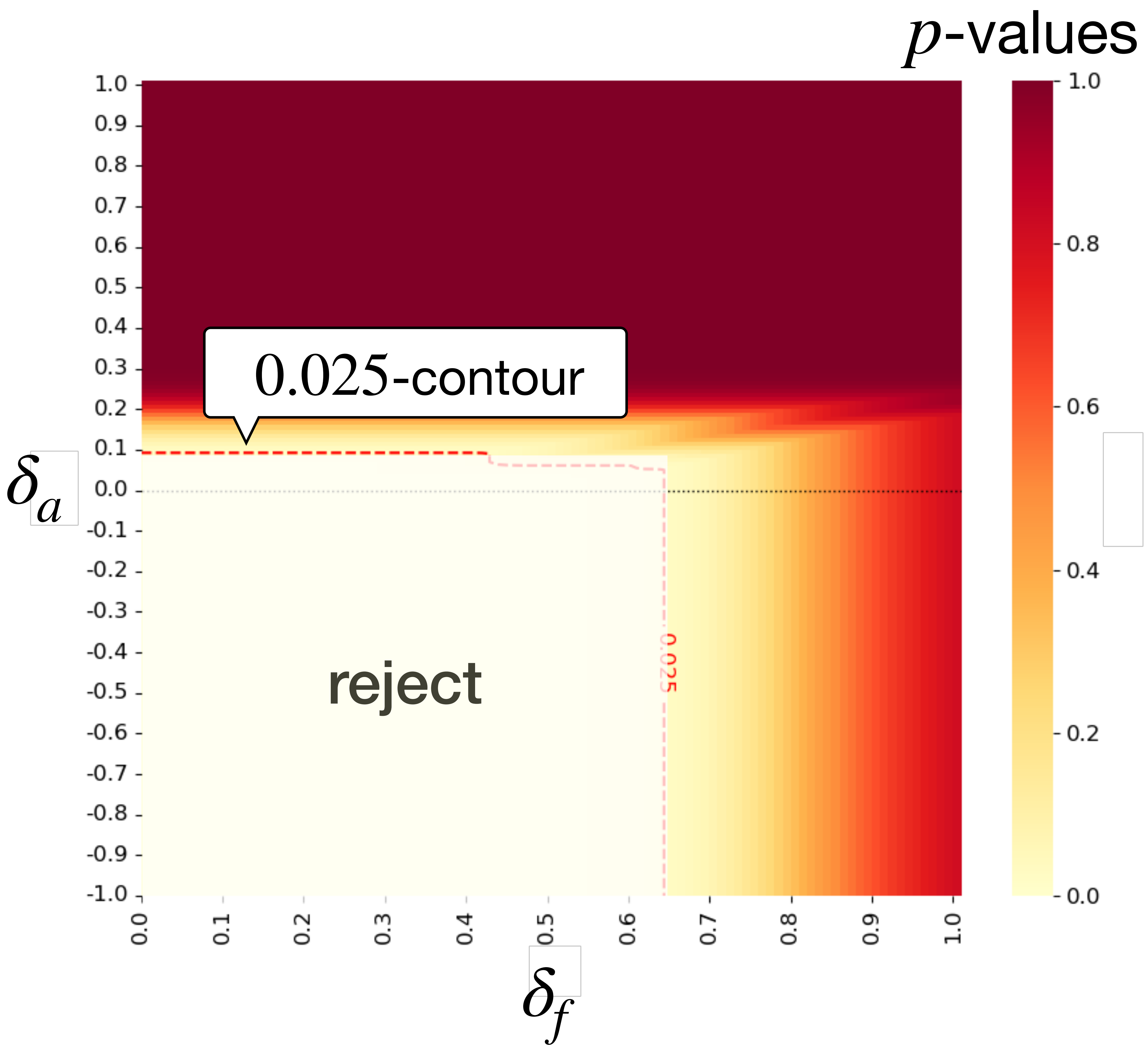
- our test yields $p < 0.001$
- reject the null for $\alpha < 0.01$
- strong statistical evidence that suggests the existence of a more accurate and more fair alternative

go

the size of possible improvements

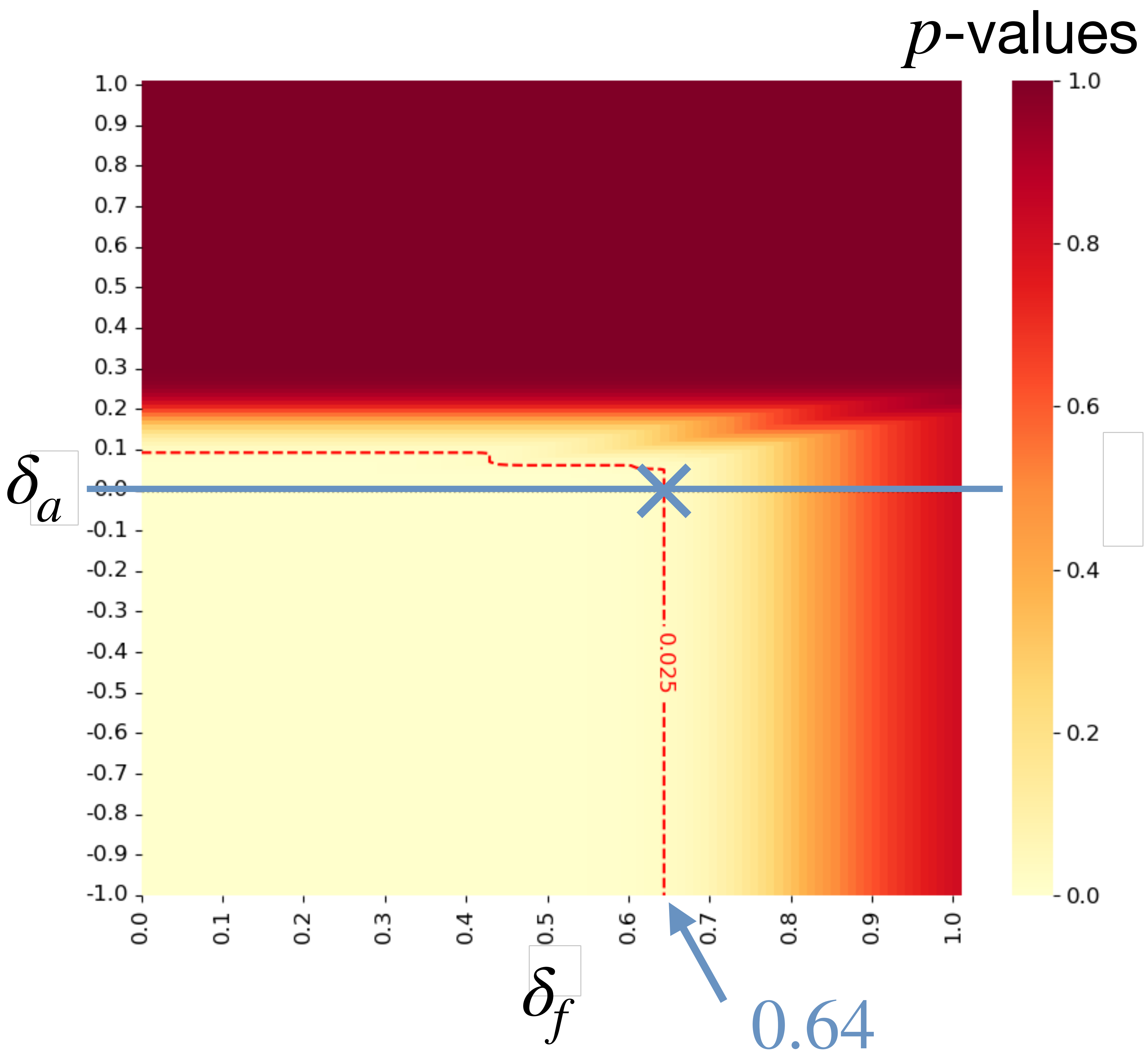
- we further explore the size of possible improvements in accuracy and fairness
- we test $(\delta_a, \delta_a, \delta_f)$-improvability across different values of δ_a and δ_f
  - improve accuracy simultaneously for both groups by at least δ_a percent
  - improve fairness by at least δ_f percent
  - larger δ requires bigger improvements

the size of possible improvements



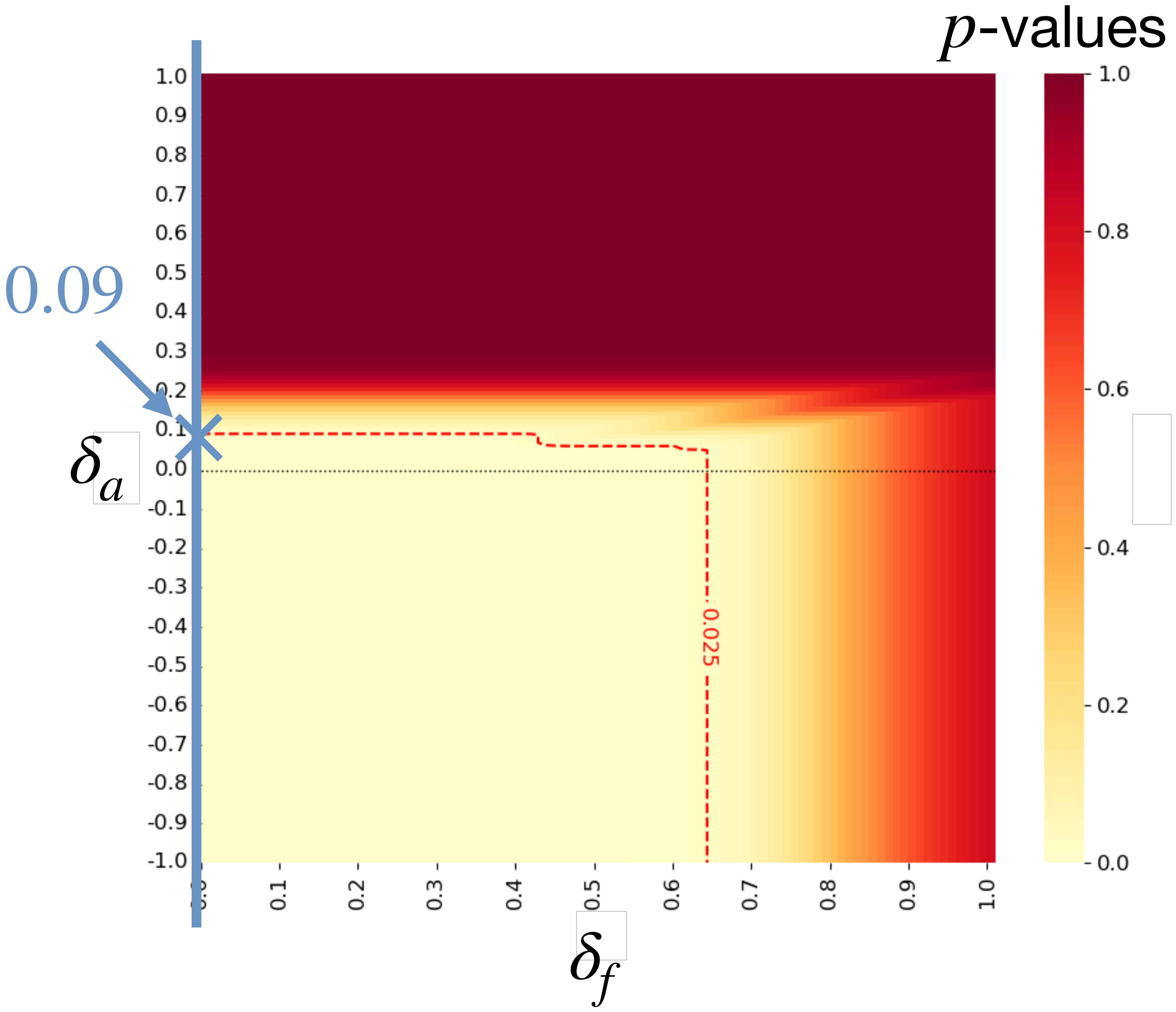
- focus on the random-forest-based selection rule
- compute p-values for different (δ_a, δ_f) pairs
- set 5% significance level ($\alpha/2 = 0.025$)

the size of possible improvements



- focus on the random-forest-based selection rule
- compute p-values for different (δ_a, δ_f) pairs
- set 5% significance level ($\alpha/2 = 0.025$),
  - we can reduce disparate impact by 64%, maintaining accuracy for all groups

the size of possible improvements



- focus on the random-forest-based selection rule
- compute p-values for different (δ_a, δ_f) pairs
- set 5% significance level ($\alpha/2 = 0.025$),
  - we can reduce disparate impact by 64%, maintaining accuracy for all groups
  - we can also reduce accuracy while maintaining fairness, but only by 9%

takeaways

in this application:

- it is possible to simultaneously improve on the accuracy and the fairness of the status quo algorithm
- (statistically) large improvements in fairness are possible without compromising on accuracy, while the reverse is not true

conclusion

- we develop a **statistical framework** and a **test** to determine whether there exist alternatives that outperform the status quo algorithm on multiple criteria
- our test is **practical**:
  - it accommodates most fairness/accuracy metrics proposed in the literature
  - it allows for any exogenous constraints on permissible algorithms
- our test has **several theoretical guarantees**:
  - asymptotically valid, consistent, and (more) robust to manipulation by the analyst
- we illustrated its use on a dataset from Obermeyer et al. (2019)

thank you ☺

questions or comments?

comments on definition

the ideal definition

a_1 improves on a_0 if

$$U_A^r(a_1) \geq U_A^r(a_0) \text{ AND}$$
$$U_A^b(a_1) \geq U_A^b(a_0) \text{ AND}$$
$$|U_F^r(a_1) - U_F^b(a_1)| \leq |U_A^r(a_0) - U_A^b(a_b)| \text{ AND}$$

one of them holds strictly.

our alternative hypothesis

a_1 improves on a_0 if

$$U_A^r(a_1) > U_A^r(a_0) \text{ AND}$$
$$U_A^b(a_1) > U_A^b(a_0) \text{ AND}$$
$$|U_F^r(a_1) - U_F^b(a_1)| < |U_A^r(a_0) - U_A^b(a_b)|$$

- there is a subtle gap between what we want to test and what we can statistically test due to technical issues related to "**testability**"
  - the space for the null hypothesis must be closed; otherwise, we cannot construct a test that is both valid and consistent (distributions on the boundary create challenges)

- however, we expect that this gap does not have a significant impact in practice

test results

	Accuracy (Black)			Accuracy (White)			Unfairness			
	a_1	a_0	p_b	a_1	a_0	p_w	a_1	a_0	p_f	p
Iteration 1	7.44	6.33	0.0000	7.35	5.14	0.0000	0.09	1.19	0.0000	0.0000
Iteration 2	7.50	6.32	0.0001	7.41	5.11	0.0000	0.09	1.20	0.0000	0.0001
Iteration 3	7.55	6.67	0.0001	7.25	5.15	0.0000	0.30	1.52	0.0000	0.0001
Iteration 4	7.46	6.35	0.0000	7.31	5.06	0.0000	0.15	1.28	0.0000	0.0000
Iteration 5	7.76	6.88	0.0009	7.33	5.27	0.0000	0.43	1.61	0.0000	0.0009
Iteration 6	7.86	6.52	0.0000	7.43	5.02	0.0000	0.43	1.51	0.0002	0.0002
Iteration 7	7.66	6.74	0.0005	7.40	5.19	0.0000	0.26	1.55	0.0001	0.0005

TABLE 1. The candidate algorithm a_1 in the table is based on random forests. Reported p-values are computed via bootstrap with 10,000 iterations. The median p-value is 0.0001.