Classification Under Strategic Self-Selection

Guy Horowitz [*1] **Yonatan Sommer** [*1] **Moran Koren** [2] **Nir Rosenfeld** [1]

Abstract

When users stand to gain from certain predictive outcomes, they are prone to act strategically to obtain predictions that are favorable. Most current works consider strategic behavior that manifests as users modifying their features; instead, we study a novel setting in which users decide whether to even participate (or not), this in response to the learned classifier. Considering learning approaches of increasing strategic awareness, we investigate the effects of user self-selection on learning, and the implications of learning on the composition of the self-selected population. Building on this, we propose a differentiable framework for learning under self-selective behavior, which can be optimized effectively. We conclude with experiments on real data and simulated behavior that complement our analysis and demonstrate the utility of our approach.

1. Introduction

Machine learning is increasingly being used for informing decisions regarding humans; some common examples include loan approvals, university admissions, job hiring, welfare benefits, and healthcare programs. In these domains, learned models often serve as 'gatekeepers', used for screening potential candidates in order to determine their qualification (e.g., for a job, loan, or program). This approach is based on the premise that more accurate models should provide better screening—which in turn should enable better decisions regarding additional costly testing (e.g., who to interview or to recruit for a try-out period) and consequent actions (e.g., who to hire). But conventional learning methods optimize for accuracy on the distribution of input data, i.e., the train-time population of candidates; this overlooks the important fact that *who will apply after model deployment*—and who will not—often depends on the screening rule itself.

In this work we study classification of strategic agents that choose whether to apply or not in response to the learned classifier. Strategic candidates apply only if the expected utility from passing screening outweighs associated costs; thus, application choices derive from beliefs regarding classification outcomes. Since these choices in aggregate determine the test-time distribution, learning becomes susceptible to *self-selection*—namely selection that is carried out by the agents which predictions target. Our goal in this paper is to study learning under such self-selective behavior, which we believe is prevalent in many application domains. We seek to: (i) establish the ramifications of self-selection on conventional learning methods; (ii) propose a strategically robust method that is accurate on the self-selective distribution it induces; (iii) study the power of such methods to influence choices and shape the applicant population; and (iv) propose means for regulating and mitigating potential ill effects.

Our setting considers a firm which trains a classifier to be used for screening, where applicants who pass screening then partake in an accurate but costly test (e.g., trial period) that determines final outcomes (e.g., hiring). Candidates would like to pass the test, but also to avoid unnecessary testing costs; the challenge for them is that they do not know a-priori whether or not they will pass screening, making their decisions regarding application inherently uncertain. To cope with this, candidates can make use of relevant statistics regarding their chances of being hired. We imagine these as being made public either by a third party (e.g., auditor, media outlet), or by the firm itself, e.g. due to regulation on transparency (Matthews & Murphy, 2023) or as a service to prospective candidates.[1] The statistics we consider rely on a subset of (categorical) features describing candidates that provide semi-individualized, group-level information, useful to them for making informed application decisions. Since the choice of classifier determines the reported statistics, these become the interface through which learning influences applications. This process is illustrated in Figure 1.

The goal of learning in our setting is to train a classifier that will be accurate on the induced applicant distribution—

[*]Equal contribution [1]Faculty of Computer Science, Technion – Israel Institute of Technology [2]Department of Industrial Engineering and Management, Ben Gurion University. Correspondence to: Nir Rosenfeld <nirr@cs.technion.ac.il>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

[1]This is similar in spirit to e.g. credit calculators, that based on partial information provide an estimated range of likely credit scores.

as determined by the classifier, indirectly through how it shapes self-selection. We study how learning approaches of increasing strategic sophistication affect, and are affected by, the process of self-selection. We begin by showing that whereas learning optimizes for accuracy, candidates benefit from the classifier's *precision*, which governs their decisions regarding application. A classifier's performance on the induced (test-time) distribution therefore depends on how it balances accuracy and precision. This also means that a *strategic* learner can maximize induced accuracy by carefully controlling its precision for different candidates as a means for shaping the population of eventual applicants. Our results show that this, coupled with the firm's informational advantage, provides it with much power: under mild conditions, learning can fully determine for each group in the population whether its members will apply, or not.

To restrict this power, we propose to enforce a certain independence criterion, which draws connections to the literature on fairness. Our main result here is that this ensures that applications adhere to a natural, classifier-independent 'ordering', which relies only on the innate group-level base rate. We show how this can allow a social planner to implement affirmative action policies using targeted subsidies.

We then switch gears and turn to proposing a practical method for learning under strategic self selection. Our method is differentiable and so can be optimized using gradient methods. Our first step is to model self-selection in the objective using per-example weights, where $w_i = 1$ if candidate i applies, and $w_i = 0$ if she does not; importantly, these weights depend on the learned classifier. We then show how weights can be effectively 'smoothed', so that gradients can be passed through application decisions. The challenge is that applications depend on precision, which in turn depends on the predictions of the classifier that is being optimized. For this we propose a differentiable proxy for (conditional) precision, and provide an effective implementation. We conclude with an empirical demonstration of our approach in a semi-synthetic experimental setting that uses real data and simulated self-selective behavior. Code is publicly available at `https://github.com/Ysommer/GKSC-ICML`.

1.1. Related work

Strategic classification. Our work is tightly connected to the growing literature on *strategic classification* (Hardt et al., 2016; Brückner et al., 2012), in which learning must cope with agents that can strategically modify their features (at a cost) in order to obtain preferred predictive outcomes. There is an ongoing effort to extend and generalize beyond the original problem setting; examples include support for richer models of user behavior (Jagadeesan et al., 2021; Sundaram et al., 2021; Levanon & Rosenfeld, 2022; Eilat et al., 2023), relaxing informational assumptions (Ghalme et al., 2021; Bechavod et al., 2022; Barsotti et al., 2022; Lin & Zrnic, 2023; Shao et al., 2023; Lechner et al., 2023; Harris et al., 2023; Rosenfeld & Rosenfeld, 2023), and introducing causal elements (Miller et al., 2020; Chen et al., 2023; Horowitz & Rosenfeld, 2023; Mendler-Dünner et al., 2022). These works, as well as the large majority of others in the field, focus on feature modification as the action that users can take (one notable exception is Krishnaswamy et al. (2021), who allow users to withhold certain features). In contrast, our work extends the literature by considering a drastically different type of action—namely the initial choice of users regarding whether to participate or not.

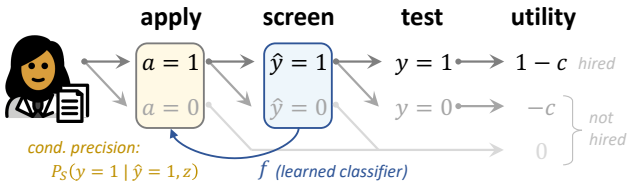


Figure 1: **The application process**. Candidates who apply must first pass a screening classifier; if successful, they advance to take a costly qualifying test. Candidates are strategic, and apply only if it is cost-effective. Since their likelihood of passing screening depends on the classifier (through its conditional precision on past data), learning has the power to shape the composition of the applicant population.

Screening, selection, and self-selection. The study of self-selection has a significant history in economics; some recent works that are relevant to our context include Lagziel & Lehrer (2021) and Lagziel & Lehrer (2019) who analyze signal distributions in filtering mechanisms and identify conditions leading to inefficiencies from excessive filtration steps in selection processes; Carroll (2017) who focuses on screening in principal-agent models; and Courty & Hao (2000) who investigates dynamic pricing as a tool to screen consumers with low willingness to pay. Most related to ours is Koren (2024), which establishes the connection between hiring and self-selection, as mediated by the quality of screening. In machine learning, several recent works study the use of classifiers for screening (Wang et al., 2022), also for strategic agents (Cohen et al., 2023; Beyhaghi et al., 2023), and with connections to fairness (Khalili et al., 2021; Blum et al., 2022; Okati et al., 2023). Other works study learning in settings with self-selection, although in different contexts and with differing goals. Zhang et al. (2021) consider a sequential screening setting where applicants can decide when (and if) to quit, and show how self-selection can be exploited. Ben-Porat et al. (2022) model user attrition in recommendation systems as a bandit problem with 'departing' arms. Cherapanamjeri et al. (2023) give algorithms for endogenous self-selection that controls realized labels (rather than participation). These

works have a strong game-theoretical emphasis; in contrast, our focus is primarily on learning aspects.

2. Problem setup

Consider a firm interested in training a classifier to be used for screening job applicants. Prospective candidates are represented by features x and a binary label $y \in \{0, 1\}$ indicating whether the candidate is qualified or not. We assume that x includes at least some categorical features, but allow the other features to be of any type or modality (e.g., vectors, images, text). Candidates are assumed to be sampled iid from some unknown joint distribution as $(x, y) \sim p$. Given a sample set $S = \{(x_i, y_i)\}_{i=1}^m \sim p^m$, the firm seeks to train a classifier $\hat{y} = f(x)$ to accurately predict labels y for unseen future candidates x. Typically we will have that $f(x) = \mathbb{1}\{\phi(x) > 0\}$ where ϕ is a learned score function. Once f is obtained, it is used by the firm as a 'gatekeeper' for screening: any candidate predicted to be qualified (i.e., has $\hat{y} = 1$) is invited to partake in an accurate (but costly) qualification test or trial period which reveals her true y.[2] The firm then hires any candidate deemed qualified, i.e., has $y = 1$.

Strategic application. Candidates would like to be hired, but also to avoid incurring the possibly unnecessary costs of potential testing. Assuming w.l.o.g. that candidates gain unit utility from being hired (which occurs iff $y = 1$), let $c \in [0, 1]$ be the cost candidates incur when taking the test. We assume candidates are *strategic*, and hence make informed decisions regarding whether to apply, denoted $a \in \{0, 1\}$. These are made on the basis of information regarding the screening process, as it depends on the learned f. Since testing takes place only if a candidate applies *and* passes screening (i.e., obtains $\hat{y} = 1$), utility is given by:

$$u(a) = a \cdot \hat{y} \cdot (y - c) \tag{1}$$

Candidates would like to apply only if this admits positive utility, i.e., if $u(1) \geq 0$ (note $u(0) = 0$). The firm, however, does not provide pre-application access to individual predictions $\hat{y}$—i.e., candidates cannot know with certainty whether they will pass screening or not. This, coupled with candidates not knowing their true y, means that $u(a)$ cannot be computed (nor optimized) exactly. To cope with this uncertainty, we model candidates as rational decision-makers, who choose to apply iff this maximizes their expected utility:

$$a^* = \text{argmax}_{a \in \{0,1\}} \mathbb{E}_{\tilde{p}(y, \hat{y} | x)}[u(a)] \tag{2}$$

where $\tilde{p}(y, \hat{y} \mid x)$ encodes their beliefs regarding the joint uncertainty in y and $\hat{y}$, conditional on x (which they know). We refer to candidates who select to apply via $a^* = 1$ as *applicants*. We next discuss what constitutes these beliefs.

[2]This is similar to the screen-then-test setup of Blum et al. (2022).

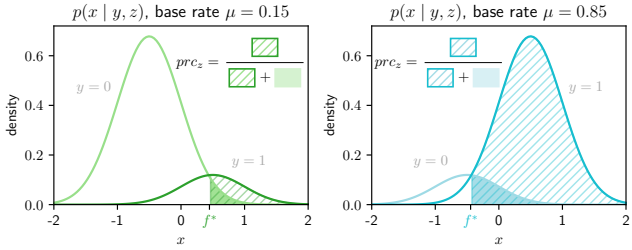


Figure 2: $\texttt{prc}_z$ under optimal f^* for high and low base rates.

Decision-making under uncertainty. To facilitate informed decision-making, we assume that the firm publishes coarse aggregate statistics concerning y and $\hat{y}$, which candidates then use to form beliefs $\tilde{p}$. In particular, let $z \subset x$ be a subset of (categorical) features of size k, and denote by K the number of distinct values z can take.[3] Then we assume the system makes public the *conditional precision* metrics:

$$\texttt{prc}_z = P_S(y = 1 \mid \hat{y} = 1, z) \quad \forall z \tag{3}$$

where P_S is the empirical distribution over the training set S. Note that $\texttt{prc}_z$ depends on the classifier f through the conditioning on (positive) predictions $\hat{y} = f(x)$.

Precision provides candidates a rough estimate of their likelihood of being hired, given that they pass the screening phase. By partitioning all candidates into K 'groups', as determined by z, candidates can obtain partially-individualized group-level information by querying $\texttt{prc}_z$.[4] Interestingly, precision turns out to be sufficient for decision-making.

Proposition 1. *Given a classifier f, the utility-maximizing application rule in Eq. (2) admits the following simple form:*

$$a^* = \mathbb{1}\{\texttt{prc}_z \geq c\} \tag{4}$$

We defer all proofs to Appendix A. Eq. (4) holds under the mild condition that $P_S(\hat{y} = 1 \mid z) > 0$, i.e., as long as in each z, not all candidates are classified as negative. Importantly, once beliefs are shaped by $\texttt{prc}_z$, who applies—and who does not—becomes dependent on the learned f.[5] This idea is illustrated in Fig. 2. When needed, we will use a_z^* to denote applications under z. Note Eq. (4) implies that a^* does not depend on screening outcomes.

Learning under self-selection. Since the goal of screening is to reduce the load on testing, screening needs to be

[3]Group variables z can correspond to sensitive or protected variables, but are not necessarily such—rather, we think of them simply as the set of variables for which the firm chooses to (or must, e.g. due to regulation) report conditional statistics.

[4]One reason for having variables $x \setminus z$ that are not conditioned on is that they materialize only after application: for example, in the academic job market, whether a submitted paper is accepted or not, or the contents of a recommendation letter.

[5]Note that precision accounts for a strict subset of the information in $\tilde{p}(y, \hat{y} \mid z)$, which is generally required for a^* in Eq. (2).

accurate on the pool of *applicants*—not on the entire population of candidates. For a given classifier f, denote by p^f its induced distribution over applicants, defined as:

$$p^f(x,y) = \begin{cases} \frac{1}{A}p(x,y) & \text{if } a^* = 1 \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

where $A = \int a^* p(x,y)\,\mathrm{d}x\,\mathrm{d}y$ is the normalizing constant. In other words, the probability of sampling a candidate from p^f remains proportional to p (with respect to all applicants) if the applicant applies, and zero otherwise. We refer to p^f as the *self-selective distribution* induced by f (Fig. 3). Given this, the goal in learning is to minimize predictive error on this induced applicant distribution:

$$\operatorname{argmin}_{f \in F} \; \mathbb{E}_{p^f(x,y)}[\mathbb{1}\{y \neq f(x)\}] \tag{6}$$

where F is some chosen model class (e.g., linear classifiers or neural networks). Note that in Eq. (6), both the loss *and* the distribution in the expectation depend on the optimized f, making it an instance of learning under *decision-dependent distribution shift* (Drusvyatskiy & Xiao, 2022). Our goals will be to devise a method for optimizing Eq. (6) effectively, and to study the effects of different learning approaches on application outcomes.

2.1. Preliminaries

Before turning to our main results, we begin with some basic analysis which sheds light on important aspects of our setup.

The role of precision. Since the decisions of candidates are based on (conditional) precision, a key question is whether higher precision is beneficial for them. Generally, the answer is yes—this is since increased precision can enable $a^* = 1$, which implies that the utility gained is positive (vs. $u = 0$ for $a^* = 0$). However, this connection is more nuanced, and is made precise by the following result:

Proposition 2. *For any group z, its expected utility is monotonically increasing in conditional precision $\texttt{prc}_z$, as long as its positive prediction rate $P_S(\hat{y} = 1 \mid z)$ is kept fixed.*

Nonetheless, and perhaps surprisingly, higher precision does not always entail better outcomes for applicants:

Proposition 3. *There exist classifiers f_1, f_2 where f_1 has higher precision, but f_2 entails higher utility for applicants.*

The proof is constructive, and relies on a simple contingency table. Thus, the choice of classifier not only determines who applies, but also the potential benefit of applying.

Incentive alignment. Since learning optimizes for accuracy, but users generally seek higher precision, it is natural to ask how these two incentives relate. While not precisely aligned, our next result shows that they are tightly connected.

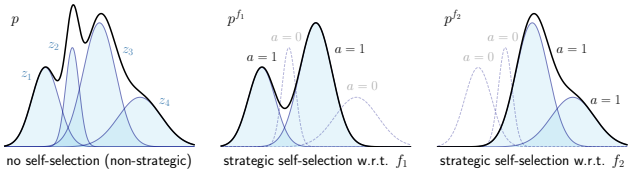


Figure 3: **Self-selective distributions**. Absent strategic behavior, all groups in the population are assumed to participate (left). But when applications depend on the learned classifier (here, f_1 vs. f_2), self-selection shapes the target distribution that the classifier will face (center vs. right).

For a group z, let $\mu_z = P_S(y = 1 \mid z)$ denote its *base rate* (which does not depend on f), and $\texttt{acc}_z = P_S(y = \hat{y} \mid z)$ denote its empirical accuracy (which does).

Proposition 4. *Fix c, and consider some group z.*

- *If $\texttt{acc}_z < 1 - \mu_z \max\{1, \frac{1}{c} - 1\}$, then necessarily $a^*_z = 0$.*
- *If $\texttt{acc}_z \geq 1 - \mu_z \min\{1, \frac{1}{c} - 1\}$, then necessarily $a^*_z = 1$.*

Prop. 4 shows that excessively low accuracy can impede application, and that high enough accuracy can enable it (see Fig. 7 in Appx. A for an illustration of these relations). Thus, disparity across group accuracies $\texttt{acc}_z$ can translate to disparity in applications a^*_z. One worrying implication is that the firm, by controlling accuracy, can indirectly control for which groups applying is cost-effective. Fortunately, there are regimes in which learning is devoid such power.

Corollary 1. *Fix $\texttt{acc}_z$. If neither conditions from Prop. 4 hold, then a^*_z is unconstrained: there exist f_1, f_2 that both attain $\texttt{acc}_z$, but $a^*_z = 0$ under f_1, and $a^*_z = 1$ under f_2.*

Prop. 4 also reveals the roles of μ_z and c as mediating factors. On the one hand, for fixed c, if some groups have low base rates μ_z, then this becomes an innate obstacle to equitable application. On the other hand, if costs can be reduced for those groups, then this disparatiy can be corrected—motivating our discussion on targeted subsidies in Sec. 3.4.

3. Analysis

In this section we consider the implications and possible outcomes of learning in the face of model-induced self-selection. We begin with a simple form of learning, and then proceed to consider increasingly more informed approaches.

3.1. Naïve non-strategic learning

The constructive relation between accuracy and precision discussed in Sec. 2.1 suggests that perhaps it may suffice to employ a conventional learning approach that naïvely maximizes predictive accuracy on the original distribution p, despite this being oblivious to self-selection. This may also seem to favor users: notice that if we discard a, then user util-

ity, given by $\mathbb{E}_p[\hat{y}(y-c)]$, correlates with accuracy, which can be written as $\mathbb{E}_p[(2\hat{y}-1)(2y-1))]$, since one is obtained from the other via linear transformations of y and $\hat{y}$.

The crux, however, is that once a *is* accounted for, this global correlation pattern breaks, as does the relation between per-group accuracy and precision. The result is a disconnect between the accuracy the model is assumed to obtain (w.r.t. p), and its actual accuracy at test time (i.e., on the induced p^f). This is demonstrated through a constructive example in Appx. C.1: let $K=2$, and for each $z \in \{1,2\}$, define $p(x,y|z)$ to be composed of two per-class Gaussians over x, each centered at their corresponding y. Then by varying the base rate μ_2 (and keeping μ_1 fixed), where by varying a single μ_z, we are able to generate arbitrary outcomes, where assumed (i.e., standard) accuracy is either: worse than induced accuracy; appears to be better; or remains the same.

3.2. Semi-strategic learning: varying the threshold

Since precision determines applications, a slightly more sophisticated "semi-stretegic" approach is to take a pre-trained 'naïve' model and then tune its precision strategically to maximize induced accuracy. For threshold classifiers $f_{\phi,\tau}(x) = \mathbb{1}\{\phi(x) > \tau\}$ with score function ϕ and threshold τ, here we build on the common practice of first training f for accuracy, and then varying τ (with ϕ fixed) to control precision.

Monotonicity. Generally, precision is expected to increase with τ, but this does not occur monotonically.[6] To simplify our analysis, we present a condition which guarantees monotonicity; this will ensure that when τ is increased, applications can transition from $a=0$ to $a=1$ at most once.

Definition 1 (Calibrated score function)**.** Let ϕ be a score function such that $p(\phi(x), y)$ is a well-defined density, and for which ϕ has full support on $[\alpha, \beta]$. We say ϕ is a *calibrated score function* w.r.t. p if for all $\tau \in [\alpha, \beta)$:

$$P(y=1 \mid \phi(x) > \tau) \geq P(y=1 \mid \phi(x) = \tau)$$

Intuitively, ϕ is calibrated if it captures the 'direction' in which $P(y=1|x)$ increases.[7] And while optimizing f for accuracy does not guarantee calibration, it is a desirable property which we can hope will emerge (perhaps approximately). Note score function calibration is a weaker condition (and is implied by) standard calibration (see Appx. B.1). Our next result links calibration and monotonicity.

Lemma 1. *Let ϕ be a score function. Then ϕ is calibrated if and only if the precision of its corresponding $f_{\phi,\tau}$, namely $P(y=1 \mid f_{\phi,\tau}(x)=1)$, is monotonically increasing in τ.*

[6] Intuitively, the reason is that τ, which determines $\hat{y}$, appears in the denominator: $\texttt{prc} = \frac{P(y=1,\hat{y}=1)}{P(\hat{y}=1)}$. Recall, however, is monotone.

[7] A similar definition appears in Okati et al. (2023) in the context of fairness, but is stronger (Appx. B.2) and used for different ends.

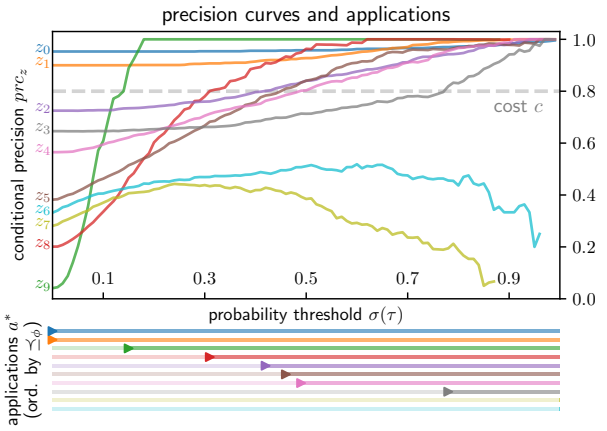


Figure 4: **Precision and application.** For a fixed score function ϕ and varying threshold τ, different groups ($K=10$, colored lines) exhibit different precision curves. Though ϕ is not mutually calibrated, most curves are roughly monotone and cross the cost c (dashed line) at most once. This induces an ordering $\preceq_\phi$ over applications a^* (lower plot). A semi-strategic learner can affect a^* only by thresholding on $\preceq_\phi$.

We say a score function is calibrated for group z if it is calibrated w.r.t $p(\cdot|z)$. This ensures that $\texttt{prc}_z$ is monotonic, which in turn determines how varying τ affects applications:

Corollary 2. *Consider some group z. If ϕ is calibrated w.r.t. z, then a^*_z is either a step function in τ, or is constant.*

Thresholds and ordering. We say that ϕ is *mutually calibrated* if it is calibrated for all z.[8] When this holds, Cor. 2 implies that ϕ induces an *ordering* over groups, and τ serves to 'threshold' applications w.r.t. that ordering. Let c_z be the minimal τ s.t. $\texttt{prc}_z \geq c$, and define $z \preceq_\phi z'$ iff $c_z \leq c_{z'}$.

Corollary 3. *Let ϕ be mutually calibrated, and consider some z, z' for which $z \preceq_\phi z'$. Then for any τ under which candidates from z' apply, candidates from z apply as well.*

This idea is exemplified in Fig. 4, which plots per-group precisions $\texttt{prc}_z$ as a function of τ for a standard logistic regression model trained on simple synthetic data ($K=10$, class-conditional Gaussians $p(x|y,z)$ per group z; details in Appendix C.3). As can be seen, mutual calibration does not hold, but nonetheless most $\texttt{prc}_z$ are generally increasing, and cross c at most once, which enables the ordering $\preceq_\phi$.

This points to an inherent limitation of semi-strategic learning: once ϕ is fixed, applications that result from tuning τ must comply with the constraints imposed by $\preceq_\phi$. On the one hand, this provides an innate restraint on the firm's power to determine who applies and who does not. On the

[8] Works discussing standard calibration across groups includes in Pleiss et al. (2017) in the context of fairness and in Wald et al. (2021) in the context of out-of-distribution generalization.

other hand, it impedes the firm's ability to obtain its predictive goals: Appendix C.2 shows an example with $K = 2$ where any reasonable classifier induces an ordering $z_1 \preceq z_2$, but optimality w.r.t. induced accuracy necessitates $z_1 \succ z_2$.

Base rates and applications. Notice that for the minimal τ, it holds that $\texttt{prc}_z = \mu_z$; this is since *all* predictions are positive, and therefore conditioning on $\hat{y} = 1$ is vacuous. Coupled with calibration, this has concrete implications:

Corollary 4. *Let z with $\mu_z \geq c$. If ϕ is calibrated w.r.t. z, then z always applies, i.e., has $a^*_z = 1$ for any value of τ.*

One consequence is that a semi-strategic firm wishing to favor 'quality' groups (i.e., having high μ_z) can easily do so by revoking screening altogether (via $\tau = -\infty$), so that $a^* = \mathbb{1}\{\mu_z \geq c\}$: this creates an appearance of equal opportunity, but in effect discriminates against low-μ_z candidates by exploiting their need to overcome a larger cost gap.

Nonetheless, for general τ, the restraining effect of μ_z can weaken. Note how in Fig. 4 curves differ in their intercepts (equal to μ_z), but also in their 'slopes'. Slopes are given by the part of $\texttt{prc}_z$ that does not depend on μ_z—using Bayes, we can write:[9]

$$\texttt{prc}_z = \frac{P(\hat{y} = 1 \mid y = 1, z)}{P(\hat{y} = 1 \mid z)} \mu_z \tag{7}$$

This shows how low-μ_z groups can apply 'before' (i.e., under lower τ) other groups with higher base rates. For example, in Fig. 4, note how z_9 has low base rate, but its precision curve rises quickly, and it applies early in $\preceq_\phi$. Another example are z_5 and z_6 who have similar base rates and therefore begin similarly, but end up very differently in the ordering.

3.3. Strategic learning: anticipating self-selection

If the firm is aware of self-selection, then it should benefit from encoding this directly into the learning objective. A natural objective for a strategic firm is therefore:

$$\operatorname*{argmin}_{f \in F} \frac{1}{m^f} \sum\nolimits_i a^*_i \ell(y_i, f(x_i)) \tag{8}$$

which is the empirical analog of Eq. (6). Here, ℓ is some proxy loss (e.g., log-loss or hinge loss), a^*_i is the application decision of candidate i (recalling this depends on f), and $m^f = \sum_i a^*_i$ is the total number of applicants under f.[10]

Strategic power. Once learning accounts for self-selection, it can utilize its informational advantage over candidates: the learner knows y and can compute $\hat{y}$ for any $x \in S$, whereas candidates only have access to $\texttt{prc}_z$. A key question is: what are the reaches of this power? Our next result shows that the firm's control can be quite extensive.

Proposition 5. *Let F be any class of functions with group-specific 'offset' terms v_z, i.e., are of the form $f(x) = g(x) + v^\top z$. If all groups have $\mu_z < c$, then the optimal f for Eq. (8) is such that only a single group applies. Otherwise, f is such that of all groups with $\mu_z < c$, only at most one group applies. In both cases, this group is that which attains the highest accuracy had it been trained on separately.*

The proof is constructive, giving an algorithm that could theoretically obtain the optimal f. Intuitively, Prop. 5 holds since linear models $w^\top x + b$ have coefficients w_z associated with each z (encoded as 1-hot); thus, f can be constructed to ensure that only a specific target group z applies by setting $w_z = 0$; setting $w_{z'} = -\infty$ for all other $z' \neq z$; and training the remaining coefficents $w_{x \setminus z}$ and b to predict well on examples from z in S, subject to $\texttt{prc}_z \geq c$.

Prop. 5 asserts that even with simple models, strategic learning has the capacity to designate a single group—that which enables the highest accuracy—and block all others candidates from applying. But this power extends further: if the firm has in mind goals other than accuracy, it can essentially shape applications, through self-selection, as it sees fit.

Corollary 5. *Let F include functions with group-specific offsets v_z as before, and consider some $f \in F$. Then for any set of groups $A = \{z : \mu_z < c\} \subseteq [K]$, it is possible to construct an f' which agrees with f on all x with $z \notin A$, but prevents application from all candidates in any $z \in A$.*

Taming strategic learning. The power to determine application outcomes derives from the ability to influence each group individually. To restrict this power, one idea is to simply remove z from the set of features. The caveat in this approach is that even if z is not used explicitly, if the remaining features $x \setminus z$ are informative of z, then z could be exploited implicitly. To ensure that learning is entirely agnostic to group memberships, we propose to augment the objective in Eq. (6) with a constraint enforcing independence:

$$\operatorname{argmin}_{f \in F} \mathbb{E}_{p^f}[\mathbb{1}\{y \neq f(x)\}] \quad \text{s.t.} \quad f(x) \perp z \tag{9}$$

In the fairness literature, this constraint is known as *statistical parity* (Dwork et al., 2012). Note that independence must be enforced on the training distribution p, rather than on the induced p^f (i.e., after the fact); otherwise, a strategic learner could 'choose' to discard any z for which satisfying Eq. (9) is either difficult or hurts predictive performance.

Our next result shows that statistical parity limits the power of strategic learning in a particular way—by inducing an ordering over group applications. Let $\preceq_\mu$ be an ordering over groups by decreasing base rate, namely $z \preceq_\mu z'$ if $\mu_z \geq \mu_{z'}$.

[9] This slope term is sometimes referred to as 'normalized' recall.

[10] Technically, Eq. (8) is ill-defined if $a^*_i = 0$ for all i. To circumvent this outcome, we can define $\frac{0}{0} = \infty$; in practice, we implement this using an additive penalty term (see end of Sec. 4).

Proposition 6. *Let f, and assume $f(x) \perp z$. For any z, z' with $z \preceq_\mu z'$, if z' applies under f, then z also applies.*

Thus, if f satisfies statistical parity, then applications must comply with $\preceq_\mu$; in other words, f can still determine who applies, but now only by thresholding on μ. Note this ordering is independent of f (c.f. $\preceq_\phi$ from Cor. 3, which is). This follows from precision now admitting the following form:

$$\texttt{prc}_z = P_S(y=1 \mid \hat{y}=1)\frac{P_S(y=1 \mid z)}{P_S(y=1)} = \texttt{prc}\frac{\mu_z}{\mu} \tag{10}$$

where $\texttt{prc}$ and μ are the global precision and base rate, respectively (see proof in Appendix A). Applications can therefore be rewritten as $a^* = \mathbb{1}\{\texttt{prc} \geq c \cdot \frac{\mu}{\mu_z}\}$, and since f only affects $\texttt{prc}$, and does so globally, high μ_z can be interpreted as reducing 'effective' per-group application costs.

3.4. Affirmative action

Consider a social planner who wishes to promote the application of some group z. If statistical parity holds, then the social planner can grant *targeted subsidies*, $s_z \geq 0$, to reduce costs for candidates in z. The decision rule becomes:

$$a^*_z = \mathbb{1}\{\texttt{prc}_z \geq c_z\} \quad \text{where} \quad c_z = c - s_z \tag{11}$$

Here, per-group costs c_z correct for low μ_z (in units of μ) as:

$$\texttt{prc}_z(s_z) \coloneqq \texttt{prc}\frac{\mu_z(s_z)}{\mu}, \qquad \mu_z(s_z) \coloneqq \mu_z + s_z\mu \tag{12}$$

Subsidies can be used to 'bump up' the target group in the ordering $\preceq_\mu$ (Prop. 6). Nonetheless, a promoted group will apply only if its inclusion does not degrade potential accuracy, given its new position. With calibration, z can be guaranteed to apply if s_z is sufficiently large so that $\mu_z > c_z$ (Cor. 4).

4. Method

We now turn to describing an effective method for optimizing the strategic learning objective in Eq. (8). This objective differs from the standard ERM objective in that the classifier f determines for each example not only its prediction, but also, through a^*, whether it should be 'turned on' or not. The challenge is therefore to account for the dependence of application decisions a^* on the classifier f being optimized. This is further complicated by the fact that a^* is discrete.

Our solution, which jointly addresses both problems, is to replace $a^*_i \in \{0,1\}$ with a continuous surrogate $\tilde{a}_i \in [0,1]$ that is differentiable in the parameters of f. These, together with the normalizing factor, are used to define differentiable per-example weights, $w^f_i = \tilde{a}_i/\tilde{m}^f$, where $\tilde{m}_f = \sum_i \tilde{a}_i$, so that $\sum_i w^f_i = 1$ always. Our proposed objective is:

$$\operatorname{argmin}_{f\in F} \sum\nolimits_i w^f_i \ell(y_i, f(x_i)) + \lambda R(f) \tag{13}$$

where R is an (optional) regularization or penalty term. We now describe how to effectively implement each component.

Differentiable applications. Recall that for a candidate i with $z_i = z$, her application decision is $a^*_i = \mathbb{1}\{\texttt{prc}_z \geq c\}$ (see Eq. (4)). A natural first step is to replace the indicator function with a smooth sigmoidal function ξ, so that:

$$\tilde{a}_i = \xi(\texttt{prc}_z - c) \tag{14}$$

Note however that the standard sigmoid is inappropriate, since its domain is the real line, whereas $\texttt{prc}_z$ and c are both in $[0,1]$. A proper alternative should satisfy the following properties: (i) have $\xi(-c) = 0$ for $\texttt{prc}_z = 0$, and $\xi(1-c) = 1$ for $\texttt{prc}_z = 1$; (ii) be indifferent at $\texttt{prc}_z = c$, i.e., $\xi(0) = 0.5$; (iii) include a temperature parameter τ s.t. $\lim_{\tau\to\infty} \xi_\tau = \mathbb{1}$. We propose the following sigmoid:

$$\xi_\tau(r; c) = \left(1 + \left(\frac{(r+c)(1-c)}{c(1-(r+c))}\right)^{-\tau}\right)^{-1} \tag{15}$$

where $\tau \geq 1$. Eq. (15) is differentiable and satisfies all three criteria, as illustrated in Appx. B.3. Note how the domain is $[-c, 1-c]$, which shifts with c. In practice we add tolerance as $c + \varepsilon$ to safeguard against statistical discrepancies between train and test.

Differentiable precision proxy. For $\tilde{a}_i$ in Eq. (14) to be differentiable, $\texttt{prc}_z$ must also be differentiable. Note that:

$$\texttt{prc}_z = P_S(y=1 \mid \hat{y}=1, z) = \frac{\sum_{j:z_j=z} y_j\,\hat{y}_j}{\sum_{j:z_j=z} \hat{y}_j} \tag{16}$$

Thus, it suffices to replace 'hard' predictions $\hat{y}$ with 'soft' probabilistic predictions $\tilde{y} \in [0,1]$. For the common case where $\hat{y} = \mathbb{1}\{\phi(x) > 0\}$, we can replace the indicator with the standard sigmoid σ, e.g., $\sigma(r) = (1+e^{-r})^{-1}$, giving:

$$\widetilde{\texttt{prc}}_z = \frac{\sum_{j:z_j=z} y_j\,\tilde{y}_j}{\sum_{j:z_j=z} \tilde{y}_j}, \qquad \tilde{y} = \sigma(\phi(x)) \tag{17}$$

While sound, this approach suffers from a subtle form of bias, which makes it ineffective for our purposes. To see this, consider that the nominator sums only over positive examples (i.e., with $y_j = 1$). For a reasonably accurate f (which implies $y, \hat{y}$ are correlated), since $\tilde{y}_j \leq \hat{y}_j$, we can expect the nominator of $\widetilde{\texttt{prc}}$ to be consistently *smaller* than that of $\texttt{prc}$; if the denominator is only mildly affected (which is reasonable to assume), then $\widetilde{\texttt{prc}}$ as a proxy becomes negatively biased. More worrying is the fact that this bias becomes *worse* as accuracy improves—a phenomena we've observed repeatedly in our empirical investigations. Thus, it becomes harder, through this proxy, to enable application for high-accuracy groups—which goes precisely against our learning goals. To remedy this, we propose to apply a corrective term:

$$\widetilde{\texttt{prc}}_z = \frac{\sum_{j:z_j=z} y_j\,\tilde{y}_j}{\sum_{j:z_j=z} \tilde{y}_j - B}, \quad B = \frac{1}{c}\sum\nolimits_i (y_i - c)(\hat{y}_i - \tilde{y}_i)$$

Table 1: **Experimental results.** For representative $c \in \{0.7, 0.8\}$ and averaged over 10 random splits, results show: induced accuracy ($\pm$stderr), number of applying groups, and the r^2 between the ideal $\preceq_\mu$ and the actual ranking based on $\texttt{prc}_z$.

	adult ($c = 0.7$)			**adult** ($c = 0.8$)			**bank** ($c = 0.7$)			**bank** ($c = 0.8$)		
	ind. acc.	apply	rank r^2	ind. acc.	apply	rank r^2	ind. acc.	apply	rank r^2	ind. acc.	apply	rank r^2
naïve	85.2 $\pm$0.3	3.0/4	0.219	(87.8)	0.1/4	(0.219)	(80.9)	5.4/10	(0.066)	-	0/10	-
semi	87.4 $\pm$0.6	2.1/4	0.135	90.1 $\pm$0.5	1.2/4	0.300	**90.0** $\pm$0.4	1.5/10	0.068	86.4 $\pm$0.6	2.4/10	0.100
$\texttt{strat}_x$	**91.1** $\pm$0.5	1.1/4	0.076	**90.5** $\pm$0.5	1.0/4	0.003	**90.1** $\pm$0.4	1.8/10	0.177	**88.7** $\pm$0.4	1.1/10	0.238
$\texttt{strat}_{x\setminus z}$	86.0 $\pm$0.4	2.5/4	0.244	89.0 $\pm$0.5	1.0/4	0.029	87.9 $\pm$0.4	6.7/10	0.170	87.0 $\pm$1.1	1.5/10	0.121
$\texttt{strat}_{\hat{y}\perp z}$	86.5 $\pm$0	0.6/4	**0.85**	88.5 $\pm$0	0.4/4	**0.537**	87.3 $\pm$0.6	0.9/10	**0.343**	88.2 $\pm$0.5	1.3/10	**0.361**

Appendix B.4 shows how B serves to de-bias $\widetilde{\texttt{prc}}$. In practice, we clip $\widetilde{\texttt{prc}}$ to be in $[0, 1]$ so that it will be well-defined.

The downside is that the corrected $\widetilde{\texttt{prc}}$ is no longer differentiable (since it now includes hard predictions $\hat{y}$). Our solution is to fix at each epoch the values of $\hat{y}_i$ from the previous iteration, and update them after each gradient step. Because in each step (and especially in later stages of training) only a few examples are likely to flip predictions, and since B sums over all examples, we expect this to only marginally affect the resulting gradient computations.

Ensuring application and well-defined precision. One issue with precision is that it is undefined if $\hat{y} = 0$ for all examples. Although this does not affect our continuous proxy, we would still like to ensure that at test time the true precision is well-behaved. Similarly, it is important that at least one group applies. Our solution is regularize via:

$$R_{\texttt{app}}(\phi; S) = -\frac{1}{K}\sum\nolimits_z \log \max_{i \in z} \tilde{y}_i - \log \max_z \tilde{a}_z \quad (18)$$

As we will see, one drawback of non-strategic approaches is that in some cases learning results in no applications at all.

Implementing independence. Enforcing statistical parity as independence constraints is generally hard, but there are many approximate methods (e.g., Agarwal et al. (2018)). In line with our general differentiable framework, we opt for a simple approach and add to the objective the penalty:

$$R_\perp(\phi; S) = \frac{1}{K}\sum\nolimits_z \left(\mathbb{E}[\tilde{y} = 1 \mid z] - \mathbb{E}[\tilde{y} = 1]\right)^2 \quad (19)$$

5. Experiments

We now turn to our experimental analysis based on real data and simulated self-selective behavior. We use two public datasets: (i) `adult` and (ii) `bank`, both of which are publicly available, commonly used for evaluation in the fairness literature (Le Quy et al., 2022), and appropriate for our setting. As group variables, we use 'race' for `adult` ($K = 4$) and 'job' for `bank` ($K = 10$). We experiment with $c \in [0.65, 0.85]$, since this range includes all significant variation in application outcome. Data is split 70-30 into train and test sets. Results are averaged over 10 random splits and include standard errors. Appendix D provides further details on data and preparation, methods, and optimization.

Methods. We compare between three general approaches, as discussed in Sec. 3: (i) `naïve`, which trains a classifier conventionally and is agnostic to self-selective behavior; (ii) `semi`, which implements the semi-strategic approach of first training a naïve classifier and then tuning its threshold strategically; and (iii) `strat`, which trains using our strategically-aware objective (Sec. 4). For the latter, we consider three variants: (iii.a) $\texttt{strat}_x$, which uses all information in x; (iii.b) $\texttt{strat}_{x\setminus z}$, which discards group features z; and (iii.c) $\texttt{strat}_{\hat{y}\perp z}$, which encourages statistical parity via regularization. All methods are based on linear classifiers.

Optimization. For a clean comparison, all methods are based on the same core implementation (pytorch) and optimized in a similar fashion.[11] We used vanilla gradient descent with learning rate 0.1 and trained for a predetermined and fixed number of epochs. Coefficients for $R_{\texttt{app}}$ and $R_\perp$ (when used) were chosen to be small yet still ensure feasibility and (approximate) independence, respectively, but overall performance was not very sensitive to chosen values.

Main results. Table 2 shows induced accuracies, application rates, and correlations between ranks based on $\texttt{prc}_z$ and $\preceq_\mu$ for representative $c = 0.7, 0.8$ (full results in Appx. E). Note all accuracies are relatively high and in a narrow range; this is since both datasets have inherently low base rates ($\mu_{\texttt{adult}} = 0.24$, $\mu_{\texttt{bank}} = 0.16$) and so even a few percentage points in accuracy are significant. The `naïve` approach is clearly suboptimal; moreover, in some cases it leads to no groups applying at all (parentheses/hyphen indicate no applications in some/all splits). Meanwhile, across all settings, $\texttt{strat}_x$ obtains the highest induced accuracy. Notice how, as suggested by Prop. 5, this follows from the number of applying groups being to be close to 1. Interestingly, `semi` does rather well, albeit inconsistently; more importantly, it does not support constraining for statistical parity, which is

[11] We also compared our implementation of `naïve` to a vanilla sk-learn implementation, ensuring that they performed similarly.

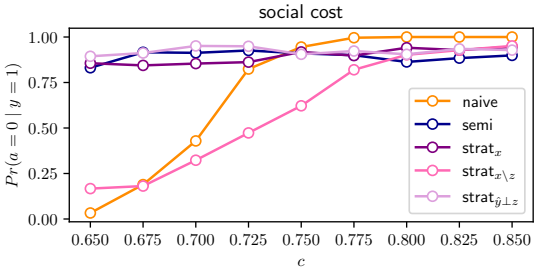


Figure 5: **Social cost** per cost c for different methods.

necessary for enabling affirmative action. As expected, discarding features in $\texttt{strat}_{x\setminus z}$ and further adding constraints in $\texttt{strat}_{\hat{y}\perp z}$ reduces their performance. However, in line with Prop. 6, the large r^2 values suggest that in most cases $\texttt{strat}_{\hat{y}\perp z}$ is able to approximately enforce the ordering $\preceq_\mu$, this effectively limiting the learner's power (results for bank were highly variate for larger c). In contrast, note how the excessive power of $\texttt{strat}_x$ pushes rankings away from $\preceq_\mu$.

Detailed analysis. Fig. 6 shows accuracy and precision curves for varying c on a typical bank instance. Results show how strat is able to consistently perform well by ensuring that only two quality groups apply: z_1 (orange), which provides high accuracy, and z_9 (cyan), which has lower accuracy but maintains its high precision. Since z_1 includes significantly more data points than z_9, it becomes the dominant component of the overall induced accuracy (black line). Note how strat increases $\texttt{prc}_1, \texttt{prc}_9$ to be above the application threshold as c increases; all other $\texttt{prc}_z$ are substantially lower (see scatterplot). In contrast, since naïve does not account for c, the order in which groups apply remains fixed to the order induced by $\texttt{prc}_z$; as c increases, less groups apply, and accuracy varies arbitrarily. Meanwhile, $\texttt{strat}_{x\setminus z}$ is able to push $\texttt{prc}_9$ higher, but struggles for the more important $\texttt{prc}_1$—succeeding for $c \leq 0.725$, but failing above. In general, the unavailability of z as a feature greatly limits its capacity to control individual $\texttt{prc}_z$.

Fig. 5 shows *social costs*, defined as the ratio of qualified applicants ($y = 1$) that did not apply ($a = 0$) over all groups, for increasing costs c and average over splits. As expected, for $\texttt{strat}_x$ the social cost is high since the number of applications is very low. And whereas semi and $\texttt{strat}_{\hat{y}\perp z}$ display similar trends, it is $\texttt{strat}_{x\setminus z}$ that shows favorable costs for low c. These are even lower that naïve where costs are obtained simply by thresholding on the non-adjusting $\texttt{prc}_z$ values, and are therefore zero for the lowest $c = 0.65$. Together, results suggest that accuracy for the firm comes at a social cost, which is not mitigated by statistical parity. This calls for other means for ensuring that qualified candidates apply across all groups—which should benefit both the firm and the candidate population.

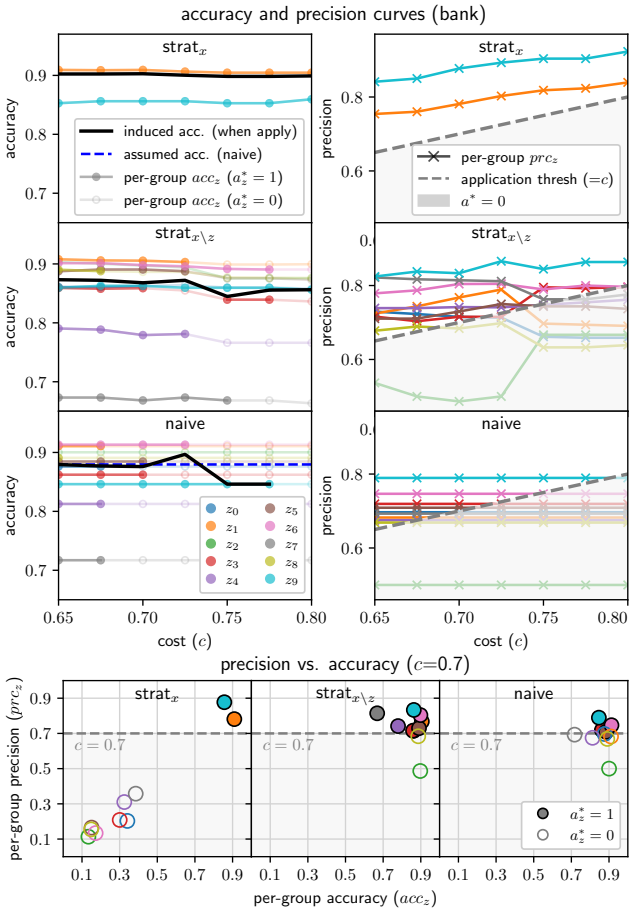


Figure 6: **(Top:)** Accuracy and precision curves on an instance of bank. **(Bottom):** Precision vs. accuracy per z.

6. Discussion

This work studies classification under strategic self-selection, a setting in which humans—as the subjects of prediction—choose if to participate or not, this in response to the learned classifier used for screening. Self-selection is a well-studied, well-documented phenomena that is highly prevalent across many social domains; given its significant role in determining the eventual composition of the applying sub-population in job hiring, school admissions, welfare programs, and other domains, we believe it is important to understand how self-selection affects, and is affected, by deployed predictive models. Our work focuses on classifiers used in a particular screening setting, but learning can influence self-selection in broader manners as well. One avenue for future work is to enable more fine-grained individualized decisions, e.g., based on private information or beliefs. Another path is to consider self-selective dynamics (e.g., in a performative prediction setting). Yet another path is to determine how to best choose which statistics to reveal. We leave these for future investigation.

Acknowledgements

This work is supported by the Israel Science Foundation grant no. 278/22.

Impact Statement

A basic assumption in conventional machine learning is that data is drawn from a fixed underlying distribution. Our work challenges this assumption by positing that in social settings, the distribution in practice will be shaped by which users choose to participate, this in response to the learned classifier. We believe this applies widely in settings where learning is used to support downstream decisions regarding humans, such as in hiring, admissions, or loan approval. Our work suggests that overlooking how the learned classifier influences participation can result in unexpected or even undesired outcomes, and propose means for learning in a way that anticipates and accounts for strategic self-selection. Our hope is that this will enable firms, governments, and public institutions to learn in ways which balance their goals with those of their potential users.

One implication of our work relates to algorithmic bias. Amounting evidence suggests that learning systems have the capacity, and often the tendency, to perpetuate social biases that exist in the data. This is often seen as an undesired artifact of blindly pursuing learning objectives (e.g., maximizing accuracy) without any social considerations. And whereas much effort has been devoted to developing methods that reduce such data-driven biases (e.g., by enforcing fairness constraints), our work's perspective provides a possible explanation for *how such biases came to be*, i.e., why the data we observe is biased to begin with. For example, note it is possible to construct a classifier that is entirely 'fair' (under any reasonable notion) when measured on the *induced* test distribution; but where this is so only because the classifier prevented certain groups from applying—groups which otherwise would have made it difficult (or impossible) to achieve fairness. This alludes to a more subtle form of 'implicit' inequity in which classifiers exploit the fact that users make decisions under uncertainty to create an illusion of fairness and thus circumvent accountability.

A second implication of our work considers its scope. Note that many of our conclusions relate to the possible discrepancies between learning in a way that is aware of self-selective behavior, compared to ways that are not. It is however important to emphasize that these conclusions hold under the simplified setting we consider, and in particular, under rational strategic behavior. Thus, while we believe our work carries practical implications for learning and policy-making, these must be considered with much care and deliberation as to the feasibility of our assumptions in reality. We are also hopeful that future work will extend beyond our focal setting, whether by considering more general behavioral models, other learning settings, or broader economic environments. As for the latter, note our work applies mostly to markets where there is scarcity in supply (e.g., jobs) and a relative abundance of demand (e.g., qualified workers). This justifies why firms in our setting need not worry about 'missing' potential candidates—only about hiring good ones. And while many markets are such (e.g., academic positions, executive managers, academic publishing, and in some cases loans), clearly there are markets (or times) in which demand is scarce and supply is abundant (e.g., many jobs but few qualified candidates). This distinction has implications not only in utilitarian terms, but also in terms of welfare and the role of learning in determining such outcomes.

A final implication regards transparency and the need for responsible usage of learning in social settings. Our results suggests that, under the setting we consider, learning has the capacity to determine the eventual user population. This grants learning much power that should not be taken lightly. As we show, when learning simply pursues the conventional goal of optimizing accuracy, this can come at the expense of preventing most groups from applying (indirectly, by reducing the cost-effectiveness of application). Since such undesirable outcomes arise even unintentionally—simply as a result of deploying a classifier—any firm that seeks to employ learning for improving decision-making should do so responsibly and transparently. The latter is crucial given the key role information plays in our setting, in the forms of what statistics are made public by the firm and therefore shared with potential candidates (e.g., conditional precision metrics). Note that what information is revealed is a *choice*—one that has major implications on outcomes. Whereas our work considers this choice as a given, in practice, we believe that it offers a valuable entry point for a social planner or regulator to implement social policy—although how to do this precisely and effectively requires further research efforts.

Overall, we hope our work aids in raising awareness to self-selection in social settings, its likely prevalence, and the potential capacity of learning to affect it.

References

Agarwal, A., Beygelzimer, A., Dudik, M., Langford, J., and Wallach, H. A reductions approach to fair classification. In Dy, J. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 60–69. PMLR, 10–15 Jul 2018.

Alvarez, S. A. An exact analytical relation among recall, precision, and classification accuracy in information retrieval. *Boston College, Boston, Technical Report BCCS-02-01*, pp. 1–22, 2002.

Barsotti, F., Koçer, R. G., and Santos, F. P. Transparency, detection and imitation in strategic classification. In *Proceedings of the 31st International Joint Conference on Artificial Intelligence, IJCAI 2022*, 2022.

Bechavod, Y., Podimata, C., Wu, S., and Ziani, J. Information discrepancy in strategic learning. In *International Conference on Machine Learning*, pp. 1691–1715. PMLR, 2022.

Ben-Porat, O., Cohen, L., Leqi, L., Lipton, Z. C., and Mansour, Y. Modeling attrition in recommender systems with departing bandits. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 6072–6079, 2022.

Beyhaghi, H., Camara, M. K., Hartline, J., Johnsen, A., and Long, S. Screening with disadvantaged agents. In *4th Symposium on Foundations of Responsible Computing*, 2023.

Blum, A., Stangl, K., and Vakilian, A. Multi stage screening: Enforcing fairness and maximizing efficiency in a pre-existing pipeline. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, pp. 1178–1193, 2022.

Brückner, M., Kanzow, C., and Scheffer, T. Static prediction games for adversarial learning problems. *The Journal of Machine Learning Research*, 13(1):2617–2654, 2012.

Carroll, G. Robustness and separation in multidimensional screening. *Econometrica*, 85(2):453–488, 2017. doi: 10.3982/ECTA14165.

Chen, Y., Wang, J., and Liu, Y. Linear classifiers that encourage constructive adaptation. *Transactions on Machine Learning Research*, 2023.

Cherapanamjeri, Y., Daskalakis, C., Ilyas, A., and Zampetakis, M. What makes a good fisherman? linear regression under self-selection bias. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, pp. 1699–1712, 2023.

Cohen, L., Sharifi-Malvajerdi, S., Stangl, K., Vakilian, A., and Ziani, J. Sequential strategic screening. In *International Conference on Machine Learning*, pp. 6279–6295. PMLR, 2023.

Courty, P. and Hao, L. Sequential Screening. *The Review of Economic Studies*, 67(4):697–717, October 2000. ISSN 0034-6527. doi: 10.1111/1467-937X.00150.

Drusvyatskiy, D. and Xiao, L. Stochastic optimization with decision-dependent distributions. *Mathematics of Operations Research*, 2022.

Dwork, C., Hardt, M., Pitassi, T., Reingold, O., and Zemel, R. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*, pp. 214–226, 2012.

Eilat, I., Finkelshtein, B., Baskin, C., and Rosenfeld, N. Strategic classification with graph neural networks. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023.

Ghalme, G., Nair, V., Eilat, I., Talgam-Cohen, I., and Rosenfeld, N. Strategic classification in the dark. In *International Conference on Machine Learning*, pp. 3672–3681. PMLR, 2021.

Hardt, M., Megiddo, N., Papadimitriou, C., and Wootters, M. Strategic classification. In *Proceedings of the 2016 ACM conference on innovations in theoretical computer science*, pp. 111–122, 2016.

Harris, K., Podimata, C., and Wu, Z. S. Strategic apple tasting. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Horowitz, G. and Rosenfeld, N. Causal strategic classifiaction. In *International Conference on Machine Learning*. PMLR, 2023.

Jagadeesan, M., Mendler-Dünner, C., and Hardt, M. Alternative microfoundations for strategic classification. In *International Conference on Machine Learning*, pp. 4687–4697. PMLR, 2021.

Khalili, M. M., Zhang, X., and Abroshan, M. Fair sequential selection using supervised learning models. *Advances in Neural Information Processing Systems*, 34:28144–28155, 2021.

Koren, M. The Gatekeeper Effect: The Implications of Pre-Screening, Self-Selection, and Bias for Hiring Processes. *Management Science*, forthcoming, May 2024. ISSN 0025-1909. doi: 10.1287/mnsc.2021.03918.

Krishnaswamy, A. K., Li, H., Rein, D., Zhang, H., and Conitzer, V. Classification with strategically withheld data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 5514–5522, 2021.

Lagziel, D. and Lehrer, E. A Bias of Screening. *American Economic Review: Insights*, 1(3):343–56, September 2019. doi: 10.1257/AERI.20180578.

Lagziel, D. and Lehrer, E. Screening Dominance: A Comparison of Noisy Signals. *American Economic Journal: Microeconomics*, 2021. ISSN 1945-7669. doi: 10.1257/MIC.20200284.

Le Quy, T., Roy, A., Iosifidis, V., Zhang, W., and Ntoutsi, E. A survey on datasets for fairness-aware machine learning. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 12(3):e1452, 2022.

Lechner, T., Urner, R., and Ben-David, S. Strategic classification with unknown user manipulations. In *International Conference on Machine Learning*. PMLR, 2023.

Levanon, S. and Rosenfeld, N. Generalized strategic classification and the case of aligned incentives. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, 2022.

Lin, L. and Zrnic, T. Plug-in performative optimization. *arXiv preprint arXiv:2305.18728*, 2023.

Matthews, V. and Murphy, M. Addressing bias in artificial intelligence: The current regulatory landscape (white paper). Technical report, Duke Law, 2023. URL https://www.thomsonreuters.com/en-us/posts/wp-content/uploads/sites/20/2023/08/Addressing-Bias-in-AI-Report.pdf?ref=blog.salesforceairesearch.com.

Mendler-Dünner, C., Ding, F., and Wang, Y. Anticipating performativity by predicting from predictions. *Advances in Neural Information Processing Systems*, 35:31171–31185, 2022.

Miller, J., Milli, S., and Hardt, M. Strategic classification is causal modeling in disguise. In *International Conference on Machine Learning*, pp. 6917–6926. PMLR, 2020.

Okati, N., Tsirtsis, S., and Gomez Rodriguez, M. On the within-group fairness of screening classifiers. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 26495–26516, 2023.

Pleiss, G., Raghavan, M., Wu, F., Kleinberg, J., and Weinberger, K. Q. On fairness and calibration. *Advances in neural information processing systems*, 30, 2017.

Rosenfeld, E. and Rosenfeld, N. One-shot strategic classification under unknown costs. *arXiv preprint arXiv:2311.02761*, 2023.

Shao, H., Blum, A., and Montasser, O. Strategic classification under unknown personalized manipulation. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Sundaram, R., Vullikanti, A., Xu, H., and Yao, F. PAC-learning for strategic classification. In *International Conference on Machine Learning*, pp. 9978–9988. PMLR, 2021.

Wald, Y., Feder, A., Greenfeld, D., and Shalit, U. On calibration and out-of-domain generalization. *Advances in neural information processing systems*, 34:2215–2227, 2021.

Wang, L., Joachims, T., and Rodriguez, M. G. Improving screening processes via calibrated subset selection. In *International Conference on Machine Learning*, pp. 22702–22726. PMLR, 2022.

Zhang, H., Cheng, Y., and Conitzer, V. Classification with few tests through self-selection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 5805–5812, 2021.

A. Proofs

Proposition 1:

Proof. If candidates base their beliefs $\tilde{p}(y, \hat{y} \mid x)$ on valid information, which in this case amounts to the aggregate per-group statistics published by the firm, namely $P_S(y, \hat{y} \mid z)$,[12] then we can rewrite the decision rule as:

$$\begin{aligned} a^* &= \text{argmax}_{a\in\{0,1\}} \mathbb{E}_{\tilde{p}(y,\hat{y}|x)}[u(a)] \\ &= \text{argmax}_{a\in\{0,1\}} \mathbb{E}_{p(y,\hat{y}|z)}[u(a)] \\ &= \mathbb{1}\{\mathbb{E}_{p(y,\hat{y}|z)}[u(1)] \geq 0\} \\ &= \mathbb{1}\{\mathbb{E}_{p(y,\hat{y}|z)}[\hat{y}\cdot(y-c)] \geq 0\} \end{aligned} \tag{20}$$

Computing the expectation directly, we get:

$$\begin{aligned} \mathbb{E}_{p(y,\hat{y}|z)}[\hat{y}\cdot(y-c)] &= \sum_{y,\hat{y}\in\{0,1\}} p(y,\hat{y} \mid z)\cdot\hat{y}\cdot(y-c) \\ &= \sum_{y\in\{0,1\}} p(y,\hat{y}=0 \mid z)\cdot 0\cdot(y-c) + p(y,\hat{y}=1 \mid z)\cdot 1\cdot(y-c) \\ &= \sum_{y\in\{0,1\}} p(y,\hat{y}=1 \mid z)\cdot(y-c) \\ &= p(y=0,\hat{y}=1 \mid z)\cdot(0-c) + p(y=1,\hat{y}=1 \mid z)\cdot(1-c) \\ &= p(y=1,\hat{y}=1 \mid z) - c\cdot(p(y=0,\hat{y}=1 \mid z) + p(y=1,\hat{y}=1 \mid z)) \\ &= p(y=1,\hat{y}=1 \mid z) - c\cdot p(\hat{y}=1 \mid z) \\ &= p(\hat{y}=1 \mid z)\cdot(y=1|\hat{y}=1,z) - c\cdot p(\hat{y}=1 \mid z) \\ &= p(\hat{y}=1 \mid z)\,(p(y=1|\hat{y}=1,z) - c) \\ &= p(\hat{y}=1 \mid z)\,(\texttt{prc}_z - c) \end{aligned} \tag{21}$$

Plugging this back into Eq. (20), we get:

$$\begin{aligned} a^* &= \mathbb{1}\{p(\hat{y}=1 \mid z)\,(\texttt{prc}_z - c) \geq 0\} \\ &= \begin{cases} \mathbb{1}\{\texttt{prc}_z \geq c\} & \text{if } p(\hat{y}=1 \mid z) > 0 \\ 0 & \text{o.w.} \end{cases} \end{aligned} \tag{22}$$

which under the assumption that $p(\hat{y}=1 \mid z) > 0$ always simplifies to $a^* = \mathbb{1}\{\texttt{prc}_z \geq c\}$. □

Proposition 2:

Proof. Immediate from Eq. (21) □

Proposition 3:

Proof. Proof by construction. Let $c = 0.5$, fix $m = 15$, and consider the following contingency tables:

f_1	$y=0$	$y=1$
$\hat{y}=0$	2	3
$\hat{y}=1$	3	7

f_2	$y=0$	$y=1$
$\hat{y}=0$	0	0
$\hat{y}=1$	5	10

Precision values are $7/(3+7) = 0.7$ for f_1 and $10/(5+10) = 2/3 < 0.7$ for f_2, so higher for f_1. Recall $u(a) = a\hat{y}(y-c)$, then utilities are:

$$u_1(1) = -\frac{2}{15}\cdot 0\cdot\frac{1}{2} - \frac{3}{15}\cdot 1\cdot\frac{1}{2} + \frac{3}{15}\cdot 0\cdot\frac{1}{2} + \frac{7}{15}\cdot 1\cdot\frac{1}{2} = 0.133$$

$$u_2(1) = -\frac{0}{15}\cdot 0\cdot\frac{1}{2} - \frac{5}{15}\cdot 1\cdot\frac{1}{2} + \frac{0}{15}\cdot 0\cdot\frac{1}{2} + \frac{10}{15}\cdot 1\cdot\frac{1}{2} = 0.166$$

[12] In particular, we assume that candidates do not hold any private information, other than x itself, that is further informative of $p(y, \hat{y} \mid z)$.

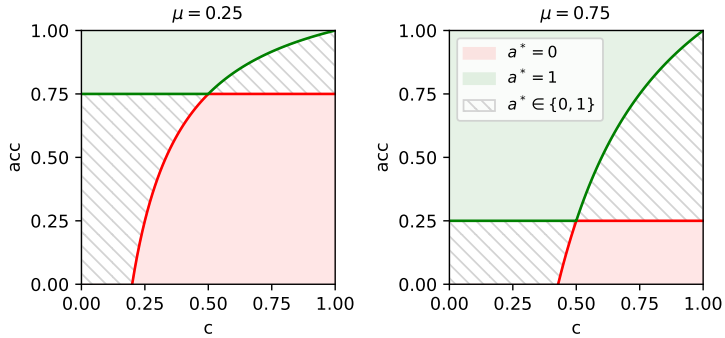


Figure 7: **Regions of application**. Plots show for different base rates μ the relations between a classifier's accuracy $\texttt{acc}_z$ on group z and that group's application a^*. In red regions $a^* = 0$, in green regions $a^* = 1$, and elsewhere a^* is unconstrained.

and so higher for f_2. □

Proposition 4 and Corollary 1:

Proof. Our proof relies on the main Theorem from Alvarez (2002) that makes precise the relation between base rate, accuracy, precision, and recall (denoted $\texttt{rcl}$) of any classifier. The result states that:

$$\mu \cdot \texttt{rcl} + (\mu - \texttt{err})\texttt{prc} = 2 \cdot \mu \cdot \texttt{rcl} \cdot \texttt{prc} \tag{23}$$

In what follows, for clarity we will omit the subscript z. Extracting $\texttt{prc}$ gives:

$$\texttt{prc} = \frac{\mu \cdot \texttt{rcl}}{\mu(2\texttt{rcl} - 1) + \texttt{err}}$$

For the first statement, bounding the RHS from above by c will guarantee $\texttt{prc} \leq c$ and therefore $a^* = 0$. Rearranging gives:

$$\texttt{acc} \leq 1 - \mu(\texttt{rcl}(\frac{1}{c} - 2) + 1)$$

The RHS is linear in $\texttt{rcl}$, is increasing if $c > 1/2$, and decreasing if $c < 1/2$ (and constant otherwise). It can therefore be bounded from above by plugging in $\texttt{rcl} = 0$ and $\texttt{rcl} = 1$, and taking the minimum. This gives:

$$\texttt{acc} \leq \min\{1 - \mu, 1 - \mu\frac{1-c}{c}\} = 1 - \mu \max\{1, \mu(\frac{1}{c} - 1)\} \quad \Rightarrow \quad a^* = 0$$

For the second statement, we can bound the RHS in 23 from below by c; this will guarantee $\texttt{prc} < c$, and therefore $a^* = 1$. Similarly, we can take the minimum from $\texttt{rcl} = 0$ and $\texttt{rcl} = 1$, which gives:

$$\texttt{acc} > \max\{1 - \mu, 1 - \mu\frac{1-c}{c}\} = 1 - \mu \min\{1, \mu(\frac{1}{c} - 1)\} \quad \Rightarrow \quad a^* = 1 \tag{24}$$

For the corollary, note that the above effectively makes use of $\texttt{rcl}$ as the only degree of freedom; thus, for any $\texttt{acc}$ satisfying:

$$1 - \mu \max\{1, \mu(\frac{1}{c} - 1)\} < \texttt{acc} \leq 1 - \mu \min\{1, \mu(\frac{1}{c} - 1)\}$$

then as long as F is sufficiently expressive, a^* remains unconstrained. □

Lemma 1:

Proof. Let $f_{\phi,\tau}(x) = \mathbb{1}\{\phi(x) > \tau\}$ where $\phi : \mathcal{X} \to [\alpha, \beta]$. Denote the random variable $\varphi = \phi(x)$ and assume that $p(\varphi, y)$ is a well defined density function, so that the marginal and conditional density functions p_φ and $p_{\varphi|y}$ are also well-defined, and further assume that $\text{supp}(p_\varphi) = [\alpha, \beta]$ (this to ensure that φ also induces a well-defined density). Denote by $\texttt{prc}(\tau)$ the precision of $f_{\phi,\tau}(x)$ as a function of the threshold $\tau \in [\alpha, \beta)$, i.e.:

$$\texttt{prc}(\tau) = P(y = 1|\phi(x) > \tau) = P(y = 1|\varphi > \tau)$$

Note that $\texttt{prc}(\tau)$ is well defined for all $\tau < \beta$, because for these values of τ, $P(\varphi > \tau) > 0$ since $\text{supp}(p_\varphi) = [\alpha, \beta]$. With Bayes theorem we can write:

$$\texttt{prc}(\tau) = P(y = 1) \cdot \frac{P(\varphi > \tau | y = 1)}{P(\varphi > \tau)}$$

If $P(y = 1) = 0$, then $\texttt{prc}(\tau) = 0$ for any τ, therefore $\texttt{prc}(\tau)$ is (weakly) monotonically increasing.

Now, assume that $P(y = 1) > 0$. Since $p(\varphi, y)$ is a well-defined density function, $P(\varphi > \tau | y = 1)$ and $P(\varphi > \tau)$ are differentiable, therefore $\texttt{prc}(\tau)$ is differentiable. Therefore, $\texttt{prc}(\tau)$ is monotonically increasing $\Leftrightarrow \texttt{prc}'(\tau) \geq 0$. Since $P(y = 1)$ is a constant, the derivative is:

$$\texttt{prc}'(\tau) = P(y = 1) \cdot \left(\frac{P(\varphi > \tau | y = 1)}{P(\varphi > \tau)} \right)'$$

Since $P(y = 1) > 0$, $\texttt{prc}'(\tau) \geq 0 \Leftrightarrow \left(\frac{P(\varphi > \tau | y = 1)}{P(\varphi > \tau)} \right)' \geq 0$. Using the quotient rule, we get that

$$\left(\frac{P(\varphi > \tau | y = 1)}{P(\varphi > \tau)} \right)' = \frac{(P(\varphi > \tau | y = 1))' \cdot P(\varphi > \tau) - P(\varphi > \tau | y = 1) \cdot (P(\varphi > \tau))'}{(P(\varphi > \tau))^2}$$

Since $P(\varphi > \tau) > 0$, the denominator is positive, therefore

$$\left(\frac{P(\varphi > \tau | y = 1)}{P(\varphi > \tau)} \right)' \geq 0 \Leftrightarrow (P(\varphi > \tau | y = 1))' \cdot P(\varphi > \tau) - P(\varphi > \tau | y = 1) \cdot (P(\varphi > \tau))' \geq 0 \tag{25}$$

Using the connection between CDF and PDF, we get

$$(P(\varphi > \tau))' = (1 - P(\varphi \leq \tau))' = -p_\varphi(\tau)$$

and

$$(P(\varphi > \tau | y = 1))' = (1 - P(\varphi \leq \tau | y = 1))' = -p_{\varphi | y=1}(\tau)$$

Plugging these in Eq. (25):

$$\texttt{prc}'(\tau) \geq 0 \qquad \Leftrightarrow -p_{\varphi|y=1}(\tau) \cdot P(\varphi > \tau) + p_\varphi(\tau) \cdot P(\varphi > \tau | y = 1) \geq 0$$

With Bayes theorem applied to $P(\varphi > \tau | y = 1)$, we get that this holds iff:

$$\begin{aligned} &- p_{\varphi|y=1}(\tau) \cdot P(\varphi > \tau) + p_\varphi(\tau) \cdot P(\varphi > \tau) \cdot \frac{P(y = 1 | \varphi > \tau)}{P(y = 1)} \geq 0 \\ &\Leftrightarrow P(\varphi > \tau) \cdot \left(-p_{\varphi|y=1}(\tau) + p_\varphi(\tau) \cdot \frac{P(y = 1 | \varphi > \tau)}{P(y = 1)} \right) \geq 0 \end{aligned}$$

Since $P(\varphi > \tau) > 0$, this holds iff:

$$- p_{\varphi|y=1}(\tau) + p_\varphi(\tau) \cdot \frac{P(y = 1 | \varphi > \tau)}{P(y = 1)} \geq 0$$

With Bayes theorem applied to $p_{\varphi|y=1}(\tau)$, we get that this holds iff:

$$\begin{aligned} &- p_\varphi(\tau) \cdot \frac{P(y = 1 | \varphi = \tau)}{P(y = 1)} + p_\varphi(\tau) \cdot \frac{P(y = 1 | \varphi > \tau)}{P(y = 1)} \geq 0 \\ &\Leftrightarrow \frac{p_\varphi(\tau)}{P(y = 1)} \cdot \Big(- P(y = 1 | \varphi = \tau) + P(y = 1 | \varphi > \tau) \Big) \geq 0 \end{aligned}$$

and since $p_\varphi(\tau) > 0$ and $P(y = 1) > 0$, we further get that this holds iff:

$$\begin{aligned} &- Pr(y = 1 | \varphi = \tau) + P(y = 1 | \varphi > \tau) \geq 0 \\ &\Leftrightarrow P(y = 1 | \varphi > \tau) \geq P(y = 1 | \varphi = \tau) \\ &\Leftrightarrow P(y = 1 | \phi(x) > \tau) \geq P(y = 1 | \phi(x) = \tau) \end{aligned}$$

Overall, we get that $\texttt{prc}'(\tau) \geq 0 \Leftrightarrow P(y = 1 | \phi(x) > \tau) \geq P(y = 1 | \phi(x) = \tau)$, i.e. $\texttt{prc}$ is monotonically increasing if and only if ϕ is a calibrated score function.

□

Corollary 2:

Proof. Let z be some group, and let ϕ be a score function with range $[\alpha, \beta]$ that is calibrated w.r.t. z. Denote $a_z^*(\tau) = \mathbb{1}\{\texttt{prc}_z(\tau) \geq c\}$ as the optimal application function of z w.r.t. the classifier $f_{\phi,\tau}(x)$. From Lemma 1 we get that $\texttt{prc}_z(\tau)$ is monotonically increasing in τ. Assume that $a_z^*(\tau)$ is neither a constant or a step function. Therefore, there exists $\alpha \leq \tau_1 < \tau_2 < \tau_3 < \beta$, such that $a_z^*(\tau_1) \neq a_z^*(\tau_2)$ and $a_z^*(\tau_2) \neq a_z^*(\tau_3)$.

- If $a_z^*(\tau_2) = 1$, then $a_z^*(\tau_3) = 0$. Therefore, $\texttt{prc}_z(\tau_2) \geq c$ and $\texttt{prc}_z(\tau_3) < c$, hence $\texttt{prc}_z(\tau_2) > \texttt{prc}_z(\tau_3)$, and since $\tau_3 > \tau_2$ this is a contradiction of the monotonicity of $\texttt{prc}_z(\tau)$.
- If $a_z^*(\tau_2) = 0$, then $a_z^*(\tau_1) = 1$. Therefore, $\texttt{prc}_z(\tau_1) \geq c$ and $\texttt{prc}_z(\tau_2) < c$, hence $\texttt{prc}_z(\tau_1) > \texttt{prc}_z(\tau_2)$, and since $\tau_2 > \tau_1$ this is a contradiction of the monotonicity of $\texttt{prc}_z(\tau)$.

Therefore, $a_z^*(\tau)$ is either a constant or a step function. □

Corollary 3:

Proof. Let ϕ be a mutually calibrated score function with range $[\alpha, \beta]$, and let z, z' such that $z \preceq_\phi z'$. From the definition of $\preceq_\phi$, this implies that $c_z \leq c_{z'}$. Let $\tau \in [\alpha, \beta]$ be such that candidates from z' apply, i.e. $a_{z'}^*(\tau) = 1$. Therefore, $\texttt{prc}_{z'}(\tau) \geq c$, and from the definition of $c_{z'}$ we get $\tau \geq c_{z'} \geq c_z$. Since ϕ is mutually calibrated, it is calibrated w.r.t. z, and from Lemma 1, $\texttt{prc}_z(\tau)$ is monotonically increasing, therefore $\texttt{prc}_z(\tau) \geq \texttt{prc}_z(c_z)$. From the definition of c_z, $\texttt{prc}_z(c_z) \geq c$, therefore $\texttt{prc}_z(\tau) \geq c$. Therefore $a_z^*(\tau) = 1$, i.e. candidates from z apply. □

Proposition 5:

Proof. Consider first the case where all z are such that $\mu_z < c$. As we have shown, if for some group z we have $\hat{y} = 1$ for all x in that group, then $\texttt{prc}_z = \mu_z$, and as a result, $a_z^* = 0$. To obtain the optimal classifier: (i) go over all groups, and for each group z, train f^z on the subset of data from z alone; (ii) find z^* such that f^{z^*} has the highest accuracy; and (iii) construct the final classifier $f(x) = g(x) + v^\top z$ as $g = f^{z^*}$, $v_{z^*} = 0$, and $v_z = -\infty$ for all $z \neq z^*$. This ensures that only z^* applies, and f obtains the same accuracy (on this group, and hence on all applicants) as f^{z^*}. Note that any other f' for which other groups apply can have accuracy at most that of f^{z^*}, since this would include averaging over additional groups. Hence, the constructed f is optimal. In the more general case where some groups may have $\mu_z > c$, it is impossible to guarantee that only a specific group applies (in particular when monotonicity does not hold); however, for all groups that do have $\mu_z < c$, the same reasoning as before still applies. Note that in this case the proof is not constructive, since the task of inferring the optimal subset of groups becomes combinatorially challenging. □

Corollary 5:

Proof. Let f. As before, if for some group z with $\mu_z < c$ we set $v_z = -\infty$, then this group will not apply. Furthermore, the behavior of f on any x from other $z' \neq z$ will remain the same. □

Proposition 6:

Proof. Let f, and assume $f(x) \perp z$, therefore $\hat{y} \perp z$. Let z, z' such that $z \preceq_\mu z'$, i.e. $\mu_z \geq \mu_{z'}$. Assume that z' applies under f, i.e. $\texttt{prc}_{z'} \geq c$. For clarity we will write P to mean P_S. With Bayes theorem can express $\texttt{prc}_z$ as:

$$\begin{aligned}\texttt{prc}_z &= P(y = 1 \mid \hat{y} = 1, z) = P(\hat{y} = 1 \mid y = 1, z)\frac{P(y = 1 \mid z)}{P(\hat{y} = 1 \mid z)} \\ &= P(\hat{y} = 1 \mid y = 1, z)\frac{\mu_z}{P(\hat{y} = 1 \mid z)}\end{aligned}$$

Since $\hat{y} \perp z$, $P(\hat{y} = 1 \mid z) = P(\hat{y} = 1)$ and $P(\hat{y} = 1 \mid y = 1, z) = P(\hat{y} = 1 \mid y = 1)$, so we get:

$$\texttt{prc}_z = P(\hat{y} = 1 \mid y = 1)\frac{\mu_z}{P(\hat{y} = 1)}$$

Using Bayes theorem again on $P(\hat{y} = 1 \mid y = 1)$, we get:

$$\begin{aligned}\texttt{prc}_z &= P(y = 1 \mid \hat{y} = 1)\frac{P(\hat{y} = 1) \cdot \mu_z}{P(y = 1) \cdot P(\hat{y} = 1)} \\ &= P(y = 1 \mid \hat{y} = 1)\frac{\mu_z}{P(y = 1)} \\ &= \texttt{prc}\frac{\mu_z}{\mu}\end{aligned}$$

where $\texttt{prc}$ and μ are the global precision and base rate, respectively. In the same way, $\texttt{prc}_{z'} = \texttt{prc}\frac{\mu_{z'}}{\mu}$. Therefore,

$$\texttt{prc}_z = \texttt{prc}\frac{\mu_z}{\mu} \geq \texttt{prc}\frac{\mu_{z'}}{\mu} = \texttt{prc}_{z'}$$

And since $\texttt{prc}_{z'} \geq c$, we get that $\texttt{prc}_z \geq c$, i.e. z applies. □

B. Additional results and illustrations.

B.1. Calibrated score functions vs. calibrated classifiers

Our next result connects score function calibration (Definition 1) to the standard notion of calibration for probabilistic classifiers. In particular, we show that under the assumption that $p(\phi(x), y)$ is a well-defined density, score function calibration is a weaker requirement, and is therefore implied by standard calibration.

Lemma 2. *Let $g(x) = \hat{p}(y = 1|x)$ be a calibrated probabilistic classifier, i.e., satisfies $P(y = 1 \mid \hat{p} = \tau) = \tau$. Then $\phi = g$ is a calibrated score function (with range $[0, 1]$).*

Proof. Let $\varphi = \phi(x)$ be a random variable depicting scores, and assume that $p(\varphi, y)$ is a well defined density function, so that the marginal density function p_φ is also well-defined. If we think of ϕ as a probabilistic classifier h, then from the definition of classifier calibration we get $P(y = 1|\varphi = \tau) = \tau$, where $\tau \in [0, 1]$. Therefore, by the definition of conditional probability we get:

$$p_{y,\varphi}(y = 1, \varphi = \tau) = \tau \cdot p_\varphi(\tau) \tag{26}$$

Using the definition of conditional probability again, we get:

$$P(y = 1|\varphi > \tau) = \frac{Pr(y = 1, \varphi > \tau)}{P(\varphi > \tau)}$$

Using the law of total probability, we get:

$$\frac{P(y = 1, \varphi > \tau)}{P(\varphi > \tau)} = \frac{\int_{t=\tau}^{1} p_{y,\varphi}(y = 1, \varphi = t)\,\mathrm{d}t}{P(\varphi > \tau)}$$

Plugging in Eq. (26), we get:

$$\frac{\int_{t=\tau}^{1} p_{y,\varphi}(y = 1, \varphi = t)\,\mathrm{d}t}{P(\varphi > \tau)} = \frac{\int_{t=\tau}^{1} t \cdot p_\varphi(t)\,\mathrm{d}t}{P(\varphi > \tau)} \geq \frac{\int_{t=\tau}^{1} \tau \cdot p_\varphi(t)\,\mathrm{d}t}{P(\varphi > \tau)} = \frac{\tau \cdot P(\varphi > \tau)}{P(\varphi > \tau)} = \tau$$

Finally, this gives:

$$P(y = 1|\phi(x) > \tau) \geq \tau = P(y = 1|\phi(x) = \tau)$$

which means that ϕ is a calibrated score function over $[0, 1]$. □

B.2. Calibrated score functions vs. within group monotone classifiers

Our next result connects score function calibration (Definition 1) to the definition of *within-group monotonicity* from Okati et al. (2023). Using our notations, their definition can be stated as:

Definition 2 (Within-group monotonicity)**.** Let ϕ be a score function with range $[\alpha, \beta]$. Then ϕ is within-group monotone if for any z and for any $\alpha \leq \tau < \tau' < \beta$ such that $p(z \mid \phi(x) = \tau) > 0$ and $p(z \mid \phi(x) = \tau') > 0$, it holds that :

$$P(y = 1 \mid \phi(x) = \tau,\, z) \leq P(y = 1 \mid \phi(x) = \tau',\, z)$$

In some sense, this definition is more general than our definition of mutually calibrated score functions, because it does not require a well-defined density $p(\phi(x), y)$. However, we show that within our setting (in which we do assume that $p(\phi(x), y)$ is a well-defined density), our notion of score function calibration turns out to be a *strictly weaker* requirement than within-group monotone, and is implied by it. For simplicity, and w.l.o.g., we will prove this for a version within-group monotonicity that considers a general distribution p, i.e., absent the conditioning on z: $P(y = 1 \mid \phi(x) = \tau) \leq P(y = 1 \mid \phi(x) = \tau')$.[13]

We begin by showing that monotonicity implies calibration, and then providing an example of a score function that is calibrated but is not monotone.

Lemma 3. *Let $\phi(x)$ be a score function with range $[\alpha, \beta]$, such that $p(\phi(x), y)$ is a well-defined density, and ϕ has full support on $[\alpha, \beta]$ under p. Then if ϕ is within-group monotone, it is also a calibrated score function.*

Proof. Let $\varphi = \phi(x)$ be a random variable depicting scores, and denote its density function by p_φ. Let $\tau \in [\alpha, \beta)$. Using Bayes theorem we can write:

$$\begin{aligned} P(y = 1 \mid \varphi > \tau) &= \frac{P(y = 1)}{P(\varphi > \tau)} P(\varphi > \tau \mid y = 1) \\ &= \frac{P(y = 1)}{P(\varphi > \tau)} \int_{t=\tau}^{\beta} p_{\varphi \mid y=1}(t)\, \mathrm{d}t \end{aligned}$$

Using Bayes theorem again on $p_{\varphi \mid y=1,z}(\tau)$ we get:

$$\begin{aligned} P(y = 1 \mid \varphi > \tau) &= \frac{P(y = 1)}{P(\varphi > \tau)} \int_{t=\tau}^{\beta} P(y = 1 \mid \varphi = t) \frac{p_\varphi(t)}{P(y = 1)}\, \mathrm{d}t \\ &= \frac{1}{P(\varphi > \tau)} \int_{t=\tau}^{\beta} P(y = 1 \mid \varphi = t) \cdot p_\varphi(t)\, \mathrm{d}t \end{aligned}$$

Since ϕ is within-group monotone, for all $t > \tau$ it holds that $P(y = 1 \mid \varphi = t) \geq P(y = 1 \mid \varphi = \tau)$, therefore:

$$\begin{aligned} P(y = 1 \mid \varphi > \tau) &\geq \frac{1}{P(\varphi > \tau)} \int_{t=\tau}^{\beta} P(y = 1 \mid \varphi = \tau) \cdot p_\varphi(t)\, \mathrm{d}t \\ &= \frac{P(y = 1 \mid \varphi = \tau)}{P(\varphi > \tau)} \int_{t=\tau}^{\beta} p_\varphi(t)\, \mathrm{d}t \\ &= \frac{P(y = 1 \mid \varphi = \tau)}{P(\varphi > \tau)} P(\varphi > \tau) \\ &= P(y = 1 \mid \varphi = \tau) \end{aligned}$$

Therefore ϕ is a calibrated score function. □

Lemma 4. *If $p(\phi(x), y)$ is well-defined, then within-group monotonicity is strictly stronger than score function calibration.*

The proof is based on a constructive example of a score function that is calibrated, but is not within-group monotone. Let $x \in \mathbb{R}$, such that $x \sim U(0, 1)$, and let $P(y = 1, x) = (x - \frac{1}{3})^2 + \frac{1}{3}$ for $x \in [0, 1]$. Let $\phi(x) = \varphi = x$ be the identity score function. We get that for $\tau \in [0, 1)$,

$$P(y = 1 \mid \varphi = \tau) = \frac{P(y = 1, x = \tau)}{p_x(\tau)} = \frac{\left(\tau - \frac{1}{3}\right)^2 + \frac{1}{3}}{1} = \left(\tau - \frac{1}{3}\right)^2 + \frac{1}{3}$$

[13] Given this proof, the proof for Lemma 2 can be simplified, since a calibrated probabilistic classifier is a special case of a within-group monotone classifier, thus it implies score function calibration.

For $\tau = 0$ and $\tau' = \frac{1}{3}$, we get that $\tau < \tau'$, but

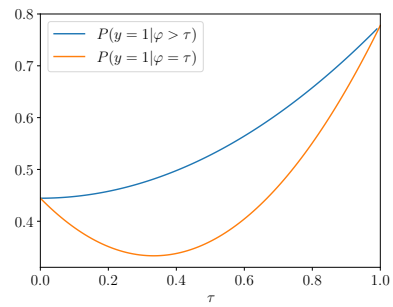
$$P(y = 1 \mid \varphi = \tau) = \frac{1}{3} + \frac{1}{9} > \frac{1}{3} = P(y = 1 \mid \varphi = \tau')$$

therefore ϕ is not within-group monotone. However, ϕ is a calibrated score function:

$$P(y = 1 \mid \varphi > \tau) = \frac{P(y = 1, x > \tau)}{P(x > \tau)} = \frac{\int_{x=\tau}^{1} \left(x - \frac{1}{3}\right)^2 + \frac{1}{3}\,\mathrm{d}x}{1 - \tau} = \frac{-3\tau^3 + 3\tau^2 - 4\tau + 4}{9(1 - \tau)}$$

As can be seen in Figure 8 below, $P(y = 1 \mid \varphi > \tau) \geq P(y = 1 \mid \varphi = \tau)$ for all $\tau \in [0, 1)$.

Figure 8: **A calibrated score function**. Plot shows data distribution with a score function $\phi(x) = \varphi = x$, where $P(y = 1 \mid \varphi = \tau) = \left(\tau - \frac{1}{3}\right)^2 + \frac{1}{3}$, and $P(y = 1 \mid \varphi > \tau) = \frac{-3\tau^3+3\tau^2-4\tau+4}{9(1-\tau)}$. Under this distribution, ϕ is a calibrated score function: $P(y = 1 \mid \varphi > \tau) \geq P(y = 1 \mid \varphi = \tau)$ for all $\tau \in [0, 1)$. However, ϕ is not within-group monotone: as can be seen, $P(y = 1 \mid \varphi = \tau)$ is decreasing for $\tau \in [0, \frac{1}{3}]$.



B.3. A sigmoid for smoothed applications

Fig. 9 illustrates our proposed sigmoid ξ in Eq. (15) from Sec. 4, intended to serve as a differentiable proxy for applications. The figure shows how the sigmoid's shape is affected by the cost parameter c and the temperature parameter τ. The top row illustrates the role of c (for fixed τ). Note how the domain is $-c, 1 - c$, and so shifts with c, but the indifference point (for which the output is 0.5) remains at input 0. The bottom row shows how, for fixed c, increasing τ increases the sigmoid's slope, this enabling ξ to better approximate a step function, but making it harder to optimize.

B.4. A corrective term for the smoothed precision proxy

First, note we can write precision as:

$$a^* = \mathbb{1}\{\texttt{prc} > c\} = \mathbb{1}\{\frac{\sum_i y_i \hat{y}_i}{\sum_i \hat{y}_i} > c\} = \mathbb{1}\{\sum_i y_i \hat{y}_i > c \sum_i \hat{y}_i\}$$

and therefore the smoothed precision proxy as:

$$\tilde{a} = \mathbb{1}\{\widetilde{\texttt{prc}} > c\} = \mathbb{1}\{\sum_i y_i \tilde{y}_i > c \sum_i \tilde{y}_i\}$$

Using the definition of the corrective term:

$$B = \frac{1}{c} \sum_i (y_i - c)(\hat{y}_i - \tilde{y}_i)$$

rearranging the expression in the indicator for hard precision gives:

$$\sum_i y_i \hat{y}_i - c \sum_i \hat{y}_i = \sum_i y_i (\hat{y}_i + \tilde{y}_i - \tilde{y}_i) - c \sum_i (\hat{y}_i + \tilde{y}_i - \tilde{y}_i) = \sum_i y_i \tilde{y}_i - c(\sum_i \tilde{y}_i + B)$$

From this, we can derive the corrected soft prediction decision rule:

$$\tilde{a} = \mathbb{1}\{\sum_i y_i \tilde{y}_i > c(\sum_i \tilde{y}_i - B)\} = \mathbb{1}\{\frac{\sum_i y_i \tilde{y}_i}{\sum_i \tilde{y}_i - B} > c\}$$

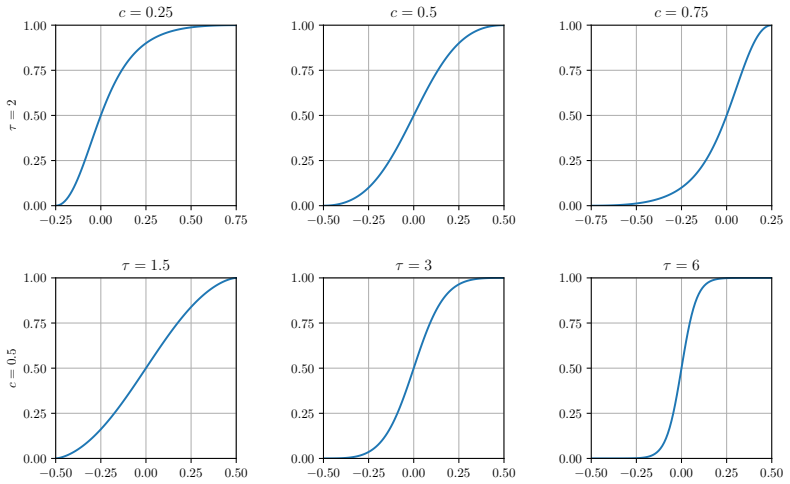


Figure 9: **Proposed sigmoid function ξ for smoothed applications.**

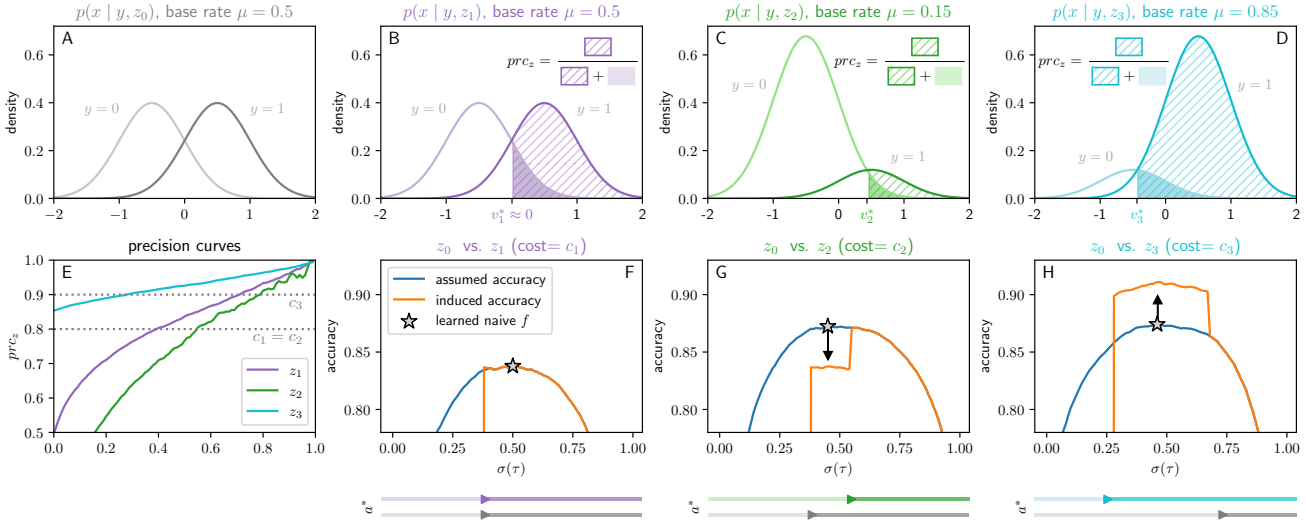


Figure 10: **Assumed vs. induced accuracy for a naive classifier.** Each experimental condition includes two groups: z_0, and one of $\{z_1, z_2, z_3\}$. **(A)-(D):** Per-group class-conditional distributions, varying only by base rate μ. Shaded/hatched areas show how precision is computed. **(E):** Precision curves for each group. **(F)-(G)**: Assumed accuracy vs. actual induced accuracy (on the applicant population) in each experimental condition. Curves show accuracies for varying thresholds τ. Also shown are the actual learned thresholds (star), showing how the assumed accuracy of a naïve classifier can be either correct (F), overly optimistic (and wrong) (G), or overly pessimistic (but correct) (H). For further details see the text in Appendix C.1.

C. Synthetic experimental results

C.1. Naive learning – assumed vs. induced accuracy

The goal of this experiment is to demonstrate how a naïve learner that is oblivious to strategic self-selection can end up with a classifier who's actual performance on the induced self-selective distribution is essentially arbitrary: it can be as expected, lower than expected, or better than expected.

Denoting by $\bar{x} = x \setminus z$ the non-group features, here we use $\bar{x} \in \mathbb{R}$. We consider four groups: z_1, z_2, z_3, z_4, and in each instance of the experiment include two groups: z_0, and an additional z_i, where $i \in \{1, 2, 3\}$. Each group z is associated with distribution $p(\bar{x}, y \mid z) = p(\bar{x} \mid y, z)p(y \mid z)$. We define the class-conditional distributions to be Gaussian with $\bar{x} \mid y = 0 \sim \mathcal{N}(-0.5,\, 0.5)$ and with $\bar{x} \mid y = 1 \sim \mathcal{N}(0.5,\, 0.5)$, which are fixed across groups, and let $\mu_i = p(y = 1 \mid z_i)$ be group-specific (i.e., do not depend on z), and where μ_i is the base rate of group z_i, and $p(y = 0 \mid z_i) = 1 - \mu_i$. In particular,

we set $\mu_1 = \mu_2 = 0.5, \mu_3 = 0.15$, and $\mu_4 = 0.85$; thus, compared to z_1, z_2 has the same base rate, z_3 has lower base rate, and z_4 has higher base rate. For the first two experiments we use $c_1 = c_2 = 0.8$, and for the last we use $c_3 = 0.9$ (we explain why below).

In each instance we sample $m = 10,000$ examples and 'naïvely' train a linear model using logistic regression on the full dataset including all groups. We then measure the learned classifier's *assumed accuracy* (i.e., made under the assumption that there is no self-selection), and its actual *induced accuracy* on the applicant population. We also vary the decision threshold τ and report both assumed accuracy and induced accuracy on the entire (probabilistic) range $\sigma(\tau) \in [0, 1]$. Note that because $\bar{x}$ is uni-dimensional, a linear model $w^\top x + b$ can be rewritten as $a \cdot \bar{x} + v^\top z + b$ where $a, b \in \mathbb{R}$ and $v \in \mathbb{R}^2$. Hard predictions $\hat{y} = f(x)$ (on which precision and accuracy depend) therefore rely only on the per-group offsets v_i, where w.l.o.g. we can assume $b = 0$. Thus, learning f amounts to learning how to "shift" each group's conditional distribution by its corresponding v^i so that a global threshold $\tau = b = 0$ performs well. We denote the learned group-specific offset terms by v_i^*.

Results are shown in Figure 10. Subplots (A)-(D) show the data distributions of the different groups, overlaid with an illustration of how precision is computed under the learned model, i.e., for each v_i^*. As can be seen, a lower base rate (as for z_2 in (C)) causes v_2^* to increase (i.e., shift right). For a fixed base rate, increasing the threshold should *increase* precision; however, the smaller base rate causes precision to generally *decrease*, and this effect is stronger. The result is a lower precision curve (compared to z_1), which is shown in (E). In contrast, an increased base rate (as for z_3 in (D)) *increases* precision—this time leading to a higher precision curve (also shown in (E)).

Sobplots (F)-(H) show assumed and induced accuracy for each experiment and for a range of (global) thresholds, and below each plot are shown the points along the threshold axis in which each group applies. For z_1 (F), whose distribution matches that of z_0 (since $\mu_1 = \mu_0$), we see that the assumed accuracy matches induces accuracy—this is since for the learned f both groups apply, and so the naïve perspective turned out to be correct. For z_2 (G), assumed accuracy is *higher* than the actual induced accuracy, since at this point only one group applies (here, z_0), which is precisely the result of the lowered precision curve (due to the lower μ_2). For z_3 (H), for which we used a larger cost ($c_3 = 0.9$), assumed accuracy is *lower* than the actual induced accuracy. Here again this is only since one group applies (this time z_3), though now due to the higher precision curve, which both 'kicks in' earlier, and provides higher accuracy at that point. Note for z_3, though the assumed accuracy differs from the actual induced accuracy, the learned classifier is nonetheless optimal also for the induced distribution. In contrast, for z_2, the naïve classifier is suboptimal, since the optimal classifier requires a larger threshold which ensures that both groups apply.

C.2. Semi-strategic learning – ordering constraints

The goal of this experiment is to demonstrate the limitations of semi-strategic learning, namely first training a score function ϕ using standard methods (i.e., naïvely), and then strategically tuning the threshold τ. In particular, we show how the fact that semi-strategic learning induces an ordering $\preceq_\phi$ on applications, derived from the learned ϕ, prevents this approach from obtaining the optimal classifier, whose applications do not comply with $\preceq_\phi$.

Again denoting by $\bar{x} = x \setminus z$ the non-group features, here we use $\bar{x} \in \mathbb{R}^2$. We define $K = 2$ groups, where each group z is associated with a data distribution $p(\bar{x}, y|z) = (\bar{x}_1 \mid y, z)p(\bar{x}_2 \mid y, z)p(y \mid z)$. Note this means that $\bar{x}_1 \perp \bar{x}_2$. We set:

$$
\begin{aligned}
&\bar{x}_1 \mid y = 0, z_1 \sim \mathcal{N}(-0.3,\, 0.1), \quad \bar{x}_1 \mid y = 1, z_1 \sim \mathcal{N}(0.2,\, 0.1),\\
&\bar{x}_2 \mid y = 0, z_1 \sim \mathcal{N}(-0.35,\, 0.3), \quad \bar{x}_2 \mid y = 1, z_1 \sim \mathcal{N}(0.25,\, 0.3),\\
&\bar{x}_1 \mid y = 0, z_2 \sim \mathcal{N}(0,\, 0.2), \quad \bar{x}_1 \mid y = 1, z_2 \sim \mathcal{N}(0,\, 0.2),\\
&\bar{x}_2 \mid y = 0, z_2 \sim \mathcal{N}(-0.35,\, 0.3), \quad \bar{x}_2 \mid y = 1, z_2 \sim \mathcal{N}(0.25,\, 0.3),\\
&\mu_1 = 0.3, \quad \mu_2 = 0.7
\end{aligned}
$$

Fig. 11 (A-D) visualize these distributions. We also set $p(z = z_1) = 0.1, p(z = z_2) = 0.9$, and sampled 10,000 examples.

The idea underlying this construction is the following: In terms of features, x_2 is generally informative of y, albeit noisy (see how the class-conditional distributions overlap in Fig. 11 (B) and (D)). This holds for both $z_1$and z_2. In contrast, x_2 is highly informative of y but *only for* z_1 (see (A), whereas for z_2 it is completely uninformative (note how the distributions in (C) fully overlap). Because a naïve learner is unaware of application, it learns a score function that relies primarily on the informative feature—namely x_2. This, coupled with the higher base rate of z_2, results in z_2 applying before z_1 once the

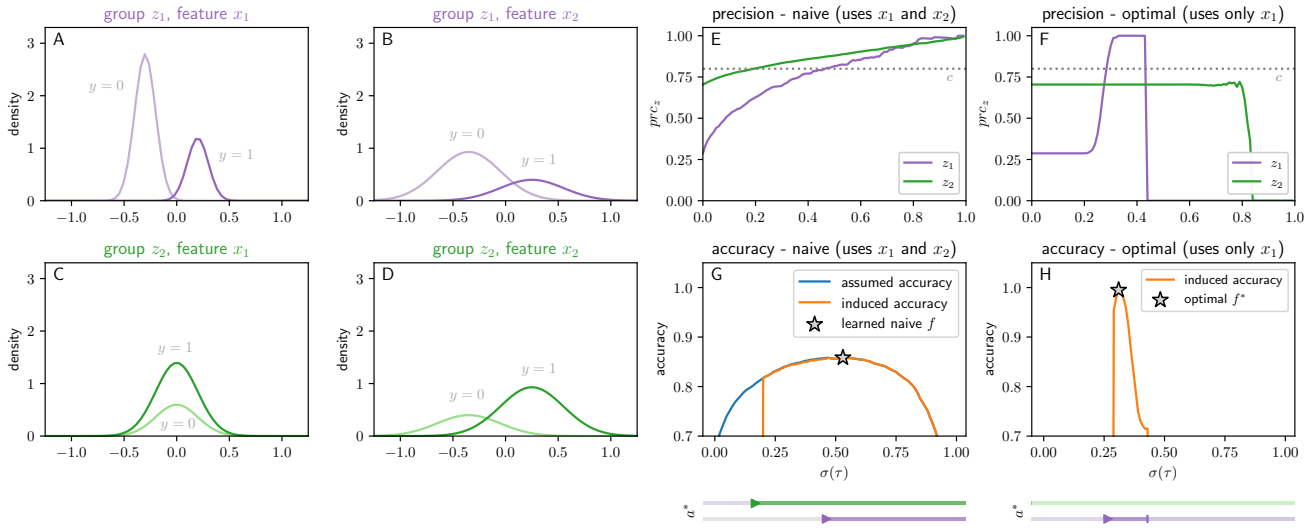


Figure 11: **Semi-strategic suboptimallity due to application order constraints.** This example includes two groups z_1, z_2 and two features $\bar{x}_1, \bar{x}_2$, constructed so that a naïvely-trained classifier f would rely mostly on $\bar{x}_2$ **(B,D)**, whereas the optimal strategic classifier f^* is to use only $\bar{x}_1$ **(A,C)**. This is because a varying the threshold on f induces an ordering $z_2 \preceq_\phi z_1$ **(E)**, enabling a maximal accuracy of ≈ 0.86 **(G)**. In contrast, f^* induces $a_1^* = 1$ but $a_2^* = 0$ **(F)**, which is impossible under $\preceq_\phi$, and enables an accuracy of ≈ 0.99. This is since only $\bar{x}_1$ is used and only for z_1 **(A)**, while avoiding its ineffectiveness for z_2 **(C)**, and not needing to further rely on the less informative $\bar{x}_2$ **(B)**. For further details see the text in Appendix C.2.

threshold is increased, i.e., $z_2 \preceq_\phi z_1$ (see (E)). As a result, the only feasible application assignments are:

$$a_1^* = 0, a_2^* = 0, \qquad a_1^* = 0, a_2^* = 1, \qquad a_1^* = 1, a_2^* = 1 \tag{27}$$

Under the optimal semi-strategic model (trained with logistic regression, threshold then optimized on the induced distribution), applications turned out to be $a_1^* = 1, a_2^* = 1$ (see (G)). However, *neither* of the applications assignments in Eq. (27) are optimal. This is because the optimal solution is to completely discard x_2, and use only x_1, but make sure that only z_1 applies (see (F)). By learning $f(\bar{x}_1)$ and varying τ, we ensure that the only information used to predict y derives from $p(\bar{x}_1 \mid z_1)$ (see A), and dodges the uninformative $p(\bar{x}_1 \mid z_2)$ (see C). This classifier induces an application profile of $a_1^* = 1, a_2^* = 0$ (impossible under $\preceq_\phi$), which in turns provides it with almost perfect induced accuracy (see (H)).

Note that the semi-strategic classifier was 'correct' in its assumed accuracy (see G).[14] Hence, its failure does not derive from a disparity between train and test distribution, but rather, from its inability to anticipate and account for strategic self-selective behavior during training.

C.3. Details for illustration of precision curves under semi-strategic learning (Fig. 4)

The following describes the experimental setup of the synthetic example illustrated in Sec. 3.2. Denoting by $\bar{x} = x \setminus z$ the non-group features, here we use $\bar{x} \in \mathbb{R}$. In this experiment, we created data for $K = 10$ groups. Each group z_i is associated with a data distribution $p(\bar{x}, y \mid z_i) = p(\bar{x} \mid y, z_i)p(y \mid z_i)$, with Gaussian class-conditional distributions: $\bar{x} \mid y = 0 \sim \mathcal{N}(a_i^-, \sigma^{2^-}_i)$ and $\bar{x} \mid y = 1 \sim \mathcal{N}(a_i^+, \sigma^{2^+}_i)$, and a base rate $\mu_i = p(y = 1 \mid z_i)$, with $p(y = 0 \mid z_i) = 1 - \mu_i$. These per-group distributions were created by randomly sampling their parameters: $a_i^- \sim \mathcal{U}(0, 0.5)$, $\sigma^{2^-}_i \sim \mathcal{U}(0.2, 0.5)$, $a_i^+ \sim \mathcal{U}(0.5, 1)$, $\sigma^{2^+}_i \sim \mathcal{U}(0.2, 0.5)$, $\mu_i \sim \mathcal{U}(0, 1)$. We sampled a total of 10,000 examples, with 1,000 examples per group, and 'naïvely' trained a linear model using logistic regression on the full dataset including all groups. Then we fixed the cost $c = 0.8$, and varied the decision threshold τ between $[0, 1]$ (applied to the probabilistic scores of the learned classifier), measuring $\texttt{prc}_i$ and a_i^* of the groups for each value of τ.

[14] Technically this is since $a_1^* = 1, a_2^* = 1$, which aligns with the naïve assumption, but note that assumed accuracy almost precisely matches induced accuracy also for lower thresholds where $a_1^* = 0, a_2^* = 1$, and so this holds more generally.

D. Experimental details

D.1. Data and preprocessing

D.1.1. ADULT

Data description. This dataset contains features based on census data from the 1994 census database that describe demographic and financial data. There are 13 features, 6 of which are categorical and the others numerical. The binary label is whether a person's income exceeds $50k. The dataset includes a total of 48,842 entries, 76% of which are labeled as negative. The data is publicly available at https://archive.ics.uci.edu/dataset/2/adult.

Preprocessing and features. We used the following numerical features: age, final_weight, education_num, capitol_gain, capitol_loss, hours_per_week. All such features were normalized to be in $[0, 1]$. We used the following categorical features: work_class, marital_status, relationship, race, sex and occupation. All of these were transformed into one-hot binary features representations. The occupation feature was reduced from 15 values to 5 based on similarity. We chose race as the group variable z. The dataset includes 5 race categories, but since Amer-Indian-Eskimo category has very few entries, it was combined with the "Other" category. We did not use two features: education, since it correlates perfectly to education_num (which we do use); and native_country, since it takes on many possible values and is uninformative of the label. The final number of features used for this setting is $d =$.

Sub-sampling. In the original data, the category "white" (coded as z_3) consists of 86% of all examples, which is highly imbalanced. To create a more balanced experimental setting, we removed at random 75% of the examples in that group. After this step, the number of examples per groups were: $z_0 : 1303;\ z_1 : 4228;\ z_2 : 788;$ and $z_3 : 9726$.

D.1.2. BANK

Data description. This dataset describes users and results of direct marketing campaigns of a Portuguese banking institution. There are 16 features, 6 of which are categorical and the others numerical. The binary label is whether a person subscribes to a proposed term deposit. The dataset includes a total of 45,211 entries, 88% of which are labeled as negative. The data is publicly available at https://www.kaggle.com/datasets/prakharrathi25/banking-dataset-marketing-targets.

Preprocessing and features. We used the following numerical features: default, balance, housing, loan, contact, day, month, duration, campaign, pdays and previous. All such features were normalized to be in $[0, 1]$. We used the following categorical features: job, marital_status, education, contact, and poutcome, and transformed them into one-hot binary features representations. Features day and month where not used given that they have no meaningful relation to the label. We chose job to determine groups features z, and removed the categories unknown and housemaid (which had few entries) to remain with 10 groups. The final number of features used for this setting is $d =$.

Sub-sampling. The original data is reasonably balanced across groups, where the number of examples per group are: $z_0 : 5,175;\ z_1 : 9,732;\ z_2 : 1,487;\ z_3 : 9,458;\ z_4 : 2,264;\ z_5 : 1,579;\ z_6 : 4,154;\ z_7 : 9,38;\ z_8 : 7,597;\ z_9 : 1,303$. However, labels are highly-imbalanced, with only 11.78% positive. As a result, we observed that learning in general sometimes converges to a solution that predicts $\hat{y} = 0$ always. To circumvent this and make for a more interesting experimental settings, we globally down-sampled 30% of all negative examples. This increased the base rate to 16.02%—which is still relatively low, but enabled more meaningful learning solutions. Results in the main body are for this setting, but for completeness, we include full results for both the original and down-sampled datasets in Appendix E.

D.2. Splits and repetitions

All experiments use a 70-30 split for partitioning the data into train and test sets. For adult, this amounts to ∼11,000 train examples and ∼4,800 test examples. For bank, this amounts to ∼30,000 train examples and ∼13,000 test examples for the original dataset, and ∼22,000 train examples and ∼9,600 test examples for the down-sampled variant. Overall we did not see evidence of overfitting, and hence had no need for a validation set. We experimented with 10 random splits and report results averaged over these splits, including standard errors.

D.3. Methods

Our experiments compares three main learning approaches, corresponding to those discussed in Sec. 3. In our experiments we use linear classifiers as the underlying hypothesis class, although generally any differentiable class could be used.

- `naïve`: A naive learner that does not account for strategic behaviour and simply optimizes for accuracy using a conventional learning approach (see Sec. 3.1). In particular, we optimize the log-loss using gradient descent.
- `semi`: A semi-strategic approach in which we first train a classifier using the `naïve` method, and then strategically choose the threshold that maximizes induced accuracy on the training set (see Sec. 3.2). Because the strategic aspect reduces to solving a uni-dimensional problem, this is implemented by line search over the feasible range of thresholds. Because this does not require taking gradient steps, we use the 'hard' induced accuracy metric (i.e., 0-1 accuracy on hard predictions $\hat{y}$) as the tresholding criterion. In this sense, `semi` has an advantage over `strat`.
- `strat`: Optimizes our proposed strategically-aware learning objective in (Eq. (13)). The objective is designed to be a differential proxy for induced accuracy, and is optimized using gradient descent (for details see Sec. 4). We consider three variants of this approach, as discussed in Sec. 3.3:
  - $\texttt{strat}_x$: Makes use of all available features in x, i.e., has the form $f(x) = \langle w, x\rangle + b$ where $w \in \mathbb{R}^d$ and $b \in \mathbb{R}$. Note that in particular, since $z \subset x$, and since we represent z as a one-hot vector of size K, this means that the model has individual per-group offset terms w_z (note this makes b redundant). This allows it to effectively set group-specific thresholds—from which it derives much of its power to influence applications.
  - $\texttt{strat}_{x\setminus z}$: Uses only non-group features, i.e., $\bar{x} = x \setminus z$. This serves as a simple heuristic for approximating independence $\hat{y} \perp z$ (see Eq. (9))—although of course this provides no guarantees, since the remaining features $\bar{x}$ can still be informative of z. Nonetheless, this approach is still much less expressive: this is since the model is now $f(x) = \langle w, \bar{x}\rangle + b$, where $w \in R^{d-K}$. In particular, this means that f does *not* have group-specific offsets w_z, and can influence applications only globally by varying the global offset b.
  - $\texttt{strat}_{\hat{y}\perp z}$: Uses only non-group features, but additionally penalizes for violation of the statistical parity independence constraints in Eq. (9). Technically, this is achieved by adding to the objective the regularization term $R_\perp$ from Eq. (19) as a 'soft' constraint which encourages $\hat{y} \perp z$.

D.4. Training, tuning, and optimization

Implementation. All code was implemented in python, and the learning framework was implemented using Pytorch. For proper comparison, all methods were optimized using the same underlying implementation framework, with `semi` and `strat` implemented as subclasses of `naïve` and using the same code base. To ensure validity we also made sure that the performance of our implementation of `naïve` matches that of a standard sklearn implementation on a subset of the experimental settings.

Optimization. Training for all methods and settings was done using vanilla gradient descent (full batch). We used a learning rate of 0.1, chosen manually to provide fast convergence while maintaining a smooth learning curve (we observes that values > 0.5 result in sporadic instability). Lower values resulted in similar results, but converged slower. We ran for a fixed number of 30,000 epochs since in most instances this was sufficient for the objective to converge sufficiently and for other metrics (train accuracy and precision) to be relatively stable. We also experimented with shorter and longer runs, but this did not seem to effect results generally.

Initialization. All models were initialized with Gaussian noise. For `strat`, which is non-convex, we observed that some initializations converged to highly suboptimal solutions—this occurred when learning 'committed' to advancing the precision of a low-quality group at the onset, and was unable to correct for this. To compensate for this, we ran with different initialization (5 for `adult`, 10 for `bank`) and chose model obtaining the highest induced accuracy on the training set. We validated that `naïve` (and therefore `semi`), which is convex, did not benefit from multiple initializations, and therefore used a single initialization.

Hyperparameters. We used the following hyperparameters:

- Temperature τ_{app} for the application sigmoid ξ in Eq. (15): 5
- Temperature τ_{prec} for the precision proxy $\widetilde{\texttt{prc}}$ in Eq. (17): 5

- Temperature τ for standard sigmoid σ: 2
- Cost tolerance ε: 0.02 for `adult`, 0.05 for `bank`.
- Regularization coefficient λ_{app} for R_{app} in Eq. (18): 1/6 for `strat` and $\texttt{strat}_{\hat{y}\perp z}$ for `adult`, 1/6 for `strat` and 1/64 for $\texttt{strat}_{\hat{y}\perp z}$ for `bank`.
- Regularization coefficient $\lambda_{\perp}$ for $R_{\perp}$ in Eq. (19) (used only for $\texttt{strat}_{\hat{y}\perp z}$): For `adult`, we set 8 for the lowest cost $c = 0.65$, and increasing linearly up to 16 up to $c = 0.85$. For `bank` we used 100.[15]

Temperature parameters where chosen mostly to provide fast convergence while ensuring that gradients do not explode. This choice is not overly sensitive, although we did observe that excessively low values resulted in premature convergence. Cost tolerance was chosen to be slightly higher for `bank` since here we did observe mild overfitting in application outcomes—which is precisely the reason for our use of a tolerance term. Regularization for applications was chosen to be as small as possible yet ensure the feasibility of applications and precisions. Note that $\texttt{strat}_{\hat{y}\perp z}$ requires a different cofficients since it must balance application reagularization with the independence penalty term. The latter was chosen to ensure that the mean squared distance is sufficiently low so that independence is reasonably-well approximated.

Compute and runtime. The main experiment was run on a CPU cluster of AMD EPYC 7713 machines (1.6 Ghz, 256M, 128 cores). A typical epoch was timed at roughly 0.05 seconds per epoch for `adult`, and 0.07 for `bank`. Thus, a single experimental instance (i.e., for a single method, cost, split, and initialization) completes in approximately 20-30 minutes for `adult` and 30-45 minutes for `bank`.

E. Additional experimental results

[15] In our implementation, to avoid numerical instability, we multiplied the expectation terms inside the mean squares operator by 10, and used a coefficient of 10, which is equivalent to a coefficient of 100 without scaling.

Table 2: **Extended experimental results.** Results show: induced accuracy ($\pm$stderr), number of applying groups, and the r^2 between the ideal $\preceq_\mu$ and the actual ranking based on $\texttt{prc}_z$. Parentheses/dashes mark settings in which there were no applications in some/all splits (out of 10).

		adult			**bank** (30% negs.)			**bank** (original)		
		ind. acc.	apply	rank r^2	ind. acc.	apply	rank r^2	ind. acc.	apply	rank r^2
$c = 0.65$	naïve	$85.5_{\pm 0.1}$	4.0/4	0.219	$87.4_{\pm 0.1}$	9.1/10	0.066	$89.4_{\pm 0.2}$	5.4/10	0.066
	semi	$86.3_{\pm 0.5}$	2.8/4	0.168	$89.4_{\pm 0.4}$	2.1/10	0.054	$92.7_{\pm 0.1}$	1.0/10	0.078
	$\texttt{strat}_x$	$90.8_{\pm 0.3}$	1.6/4	0.388	$90.4_{\pm 0.5}$	1.6/10	0.183	$92.2_{\pm 0.2}$	1.3/10	0.216
	$\texttt{strat}_{x\setminus z}$	$85.4_{\pm 0.2}$	3.1/4	0.186	$87.5_{\pm 0.3}$	7.8/10	0.243	$90.6_{\pm 0.4}$	3.6/10	0.089
	$\texttt{strat}_{\hat{y}\perp z}$	$81.7_{\pm 0.5}$	0.8/4	0.708	$88.1_{\pm 0.4}$	1.5/10	0.442	$91.7_{\pm 0.4}$	1.4/10	0.373
$c = 0.675$	naïve	$85.5_{\pm 0.1}$	3.9/4	0.219	$83.5_{\pm 0.4}$	7.8/10	0.066	$84.8_{\pm 1}$	3.3/10	0.066
	semi	$86.7_{\pm 0.6}$	2.5/4	0.154	$89.5_{\pm 0.3}$	1.4/10	0.125	$92.7_{\pm 0.1}$	1.1/10	0.067
	$\texttt{strat}_x$	$90.8_{\pm 0.5}$	1.3/4	0.149	$90.4_{\pm 0.5}$	1.7/10	0.167	$92.2_{\pm 0.2}$	1.1/10	0.304
	$\texttt{strat}_{x\setminus z}$	$85.4_{\pm 0.3}$	3.0/4	0.262	$87.5_{\pm 0.3}$	7.7/10	0.172	$90.3_{\pm 0.6}$	2.9/10	0.098
	$\texttt{strat}_{\hat{y}\perp z}$	$86.4_{\pm 0.6}$	0.2/4	0.851	$87.5_{\pm 0.5}$	1.3/10	0.429	$92.2_{\pm 0.2}$	1.0/10	0.330
$c = 0.7$	naïve	$85.2_{\pm 0.3}$	3.0/4	0.219	$80.9_{\pm 0.8}$	5.4/10	0.066	$82.7_{\pm 1.4}$	1.7/10	0.066
	semi	$87.4_{\pm 0.6}$	2.1/4	0.135	$90.0_{\pm 0.4}$	1.5/10	0.068	$92.3_{\pm 0.2}$	1.2/10	0.045
	$\texttt{strat}_x$	$91.1_{\pm 0.5}$	1.1/4	0.076	$90.1_{\pm 0.4}$	1.8/10	0.177	$92.1_{\pm 0.2}$	1.1/10	0.230
	$\texttt{strat}_{x\setminus z}$	$86.0_{\pm 0.4}$	2.5/4	0.244	$87.9_{\pm 0.6}$	6.7/10	0.170	$91.1_{\pm 0.5}$	2.6/10	0.057
	$\texttt{strat}_{\hat{y}\perp z}$	$86.5_{\pm 0.1}$	0.6/4	0.85	$87.3_{\pm 0.4}$	0.9/10	0.343	$92.0_{\pm 0.3}$	1.1/10	0.267
$c = 0.725$	naïve	$84.5_{\pm 0.7}$	1.8/4	0.219	$76.7_{\pm 1.6}$	2.3/10	0.066	(77.9)	0.5/10	(0.066)
	semi	$87.9_{\pm 0.9}$	1.7/4	0.065	$90.5_{\pm 0.3}$	1.1/10	0.107	$91.4_{\pm 0.8}$	1.1/10	0.048
	$\texttt{strat}_x$	$90.4_{\pm 0.4}$	1.0/4	0.003	$90.1_{\pm 0.4}$	1.8/10	0.193	$92.1_{\pm 0.2}$	1.2/10	0.296
	$\texttt{strat}_{x\setminus z}$	$87.0_{\pm 0.8}$	1.9/4	0.389	$87.4_{\pm 0.7}$	5.9/10	0.160	$91.4_{\pm 0.4}$	2.0/10	0.086
	$\texttt{strat}_{\hat{y}\perp z}$	$86.7_{\pm 0.7}$	0.4/4	0.659	$87.4_{\pm 0.3}$	1.0/10	0.379	$92.4_{\pm 0.1}$	1.2/10	0.311
$c = 0.75$	naïve	(90.0)	0.4/4	(0.219)	$78.7_{\pm 2.8}$	1.0/10	0.066	(77.0)	0.1/10	(0.066)
	semi	$88.0_{\pm 1}$	1.7/4	0.125	$90.0_{\pm 0.3}$	1.5/10	0.080	$90.4_{\pm 0.6}$	1.7/10	0.062
	$\texttt{strat}_x$	$91.0_{\pm 0.6}$	1.0/4	0.005	$89.8_{\pm 0.3}$	1.3/10	0.275	$91.8_{\pm 0}$	1.0/10	0.327
	$\texttt{strat}_{x\setminus z}$	$88.9_{\pm 0.6}$	1.2/4	0.094	$87.2_{\pm 1.4}$	4.5/10	0.091	$91.6_{\pm 0.3}$	1.7/10	0.056
	$\texttt{strat}_{\hat{y}\perp z}$	$87.4_{\pm 0.6}$	0.6/4	0.675	$88.2_{\pm 0.4}$	1.4/10	0.424	$92.4_{\pm 0}$	1.1/10	0.244
$c = 0.775$	naïve	(87.8)	0.1/4	(0.219)	(86.7)	0.1/10	(0.066)	(77.0)	0.1/10	(0.066)
	semi	$88.7_{\pm 0.8}$	1.5/4	0.327	$88.6_{\pm 0.7}$	1.7/10	0.111	$90.7_{\pm 0.8}$	1.3/10	0.096
	$\texttt{strat}_x$	$91.1_{\pm 0.6}$	1.0/4	0.005	$89.7_{\pm 0.3}$	1.3/10	0.191	$91.8_{\pm 0}$	1.0/10	0.352
	$\texttt{strat}_{x\setminus z}$	$89.0_{\pm 0.6}$	1.0/4	0.030	$87.5_{\pm 1.5}$	2.3/10	0.095	$91.3_{\pm 0.8}$	1.2/10	0.094
	$\texttt{strat}_{\hat{y}\perp z}$	$87.4_{\pm 0.8}$	0.4/4	0.505	$88.1_{\pm 0.4}$	1.2/10	0.462	$92.2_{\pm 0}$	1.1/10	0.200
$c = 0.8$	naïve	(87.8)	0.1/4	(0.219)	-	0/10	-	-	0/10	-
	semi	$90.1_{\pm 0.5}$	1.2/4	0.300	$86.4_{\pm 0.6}$	2.4/10	0.100	$90.4_{\pm 0.8}$	1.5/10	0.114
	$\texttt{strat}_x$	$90.5_{\pm 0.5}$	1.0/4	0.003	$88.7_{\pm 0.4}$	1.1/10	0.238	$92.1_{\pm 0.2}$	1.0/10	0.321
	$\texttt{strat}_{x\setminus z}$	$89.0_{\pm 0.5}$	1.0/4	0.029	$87.0_{\pm 1.1}$	1.5/10	0.121	$91.2_{\pm 0.8}$	1.3/10	0.061
	$\texttt{strat}_{\hat{y}\perp z}$	$88.4_{\pm 1.2}$	0.4/4	0.537	$88.2_{\pm 0.5}$	1.3/10	0.361	$91.9_{\pm 0.3}$	1.2/10	0.150
$c = 0.825$	naïve	-	0/10	-	-	0/10	-	-	0/10	-
	semi	$90.3_{\pm 0.3}$	1.1/4	0.411	$85.6_{\pm 0.4}$	2.3/10	0.107	$90.2_{\pm 0.8}$	1.2/10	0.081
	$\texttt{strat}_x$	$90.7_{\pm 0.5}$	1.0/4	0.003	$89.0_{\pm 0.3}$	1.1/10	0.292	$92.2_{\pm 0.2}$	1.0/10	0.285
	$\texttt{strat}_{x\setminus z}$	$88.9_{\pm 0.5}$	1.0/4	0.180	$87.0_{\pm 0.7}$	1.6/10	0.150	$90.8_{\pm 0.8}$	1.3/10	0.075
	$\texttt{strat}_{\hat{y}\perp z}$	$88.7_{\pm 1.1}$	0.6/4	0.354	$88.3_{\pm 0.4}$	1.1/10	0.336	$92.0_{\pm 0.3}$	1.1/10	0.281
$c = 0.85$	naïve	-	0/10	-	-	0/10	-	-	0/10	-
	semi	$90.1_{\pm 0.2}$	1.2/4	0.560	$86.9_{\pm 0.6}$	2.1/10	0.067	$90.2_{\pm 0.5}$	1.0/10	0.121
	$\texttt{strat}_x$	$91.0_{\pm 0.6}$	1.0/4	0.005	$89.0_{\pm 0.2}$	1.0/10	0.344	$91.8_{\pm 0.3}$	1.0/10	0.314
	$\texttt{strat}_{x\setminus z}$	$89.2_{\pm 0.6}$	1.0/4	0.219	$86.4_{\pm 0.8}$	1.0/10	0.102	$91.1_{\pm 0.8}$	1.2/10	0.066
	$\texttt{strat}_{\hat{y}\perp z}$	$89.1_{\pm 0.8}$	1/4	0.198	$87.6_{\pm 0.7}$	1.3/10	0.341	$92.1_{\pm 0.2}$	1.0/10	0.183