Capacity Constraints Make Admissions Processes Less Predictable

Evan Dong[1], **Nikhil Garg**[1], **Sarah Dean**[1]

[1] Cornell University

edong@cs.cornell.edu, ngarg@cornell.edu, sdean@cornell.edu

Abstract

Machine learning models are often used to make predictions about the outcomes of applications to selective programs. Many prospective school or college applicants turn to machine learning models to predict whether they will be admitted to a program, and employers may use algorithmic tools to filter out resumes predicted to have a low probability of being hired when offering interviews for a job opening. However, such decision processes differ substantially from the conventional machine learning setting: decisions are not independent across applicants. Whether a student is admitted depends on the other applicants who apply because admissions decisions are capacity-constrained. We formalize how the nature of admission decisions results in a data-generating process which is incompatible with traditional machine learning assumptions. We characterize how selection functions properties affect the difficulty of generalization to applicant pool distribution shifts, introducing two concepts: stability, which measures how many existing decisions can change when a single new applicant is introduced; and variability, which measures the number of unique students whose decisions can change. We demonstrate our theory on admissions data from the New York City high school matching system, showing that machine learning performance degrades as the applicant pool increasingly differs from the training data. Furthermore, there are larger performance drops for schools using decision rules that are less stable and more variable. Our work raises questions about the reliability of predicting individual admissions probabilities.

1 Introduction

It is common to use statistical models based on historical data to make predictions about the outcomes of applications to selective programs. Such models are often used by decision makers, as in the case of recruiters using resume filters or college admissions officers ranking candidates for review. They are also used by applicants to understand their own admissions chances, influencing application decisions. The use of machine learning (ML) in these settings is subject to substantial scrutiny, stemming from a combination of ethical concerns and from the limitations of ML, such as regarding bias, data limitations, and lack of transparency. Admissions decisions are high stakes, so errors have significant consequences; on the other hand, computational tools may be useful for both applicants and decision makers.

In this paper, we identify and analyze a new source of trouble for machine learning as classically used in the context of admissions. Admissions are inherently capacity constrained—there are only so many seats in class. As a result, the decision to admit one applicant inherently affects the chances of others in the applicant pool. This phenomenon of *cohort-dependence* clashes with the independence assumptions and formulation of machine learning. We show that the extent of this dissonance relates to properties of the decision process, and may be worse for programs that seek to admit diverse or balanced cohorts of applicants. In short, historical application decisions – that compose the training set – were functions of historical applicant pools, in ways that are not explicitly modeled in traditional machine learning. As we show, then, even small applicant pool changes can change true outcomes – applicants who would have truly been accepted historically may be rejected in the future, and vice versa. Our contributions are as follows.

(1) We introduce formal properties of admission decisions which affect the difficulty of generalization (§4.1); **(2)** we present a characterization which bridges the theory of choice functions with machine learning (§4.2); **(3)** We illustrate our theory using admissions data from the New York City high school matching system (§5). Before diving in, we begin by reviewing related work (§2) and introducing the problem setting, notation, and background (§3). We conclude with a discussion of implications in §6.

2 Related Work

Machine Learning for Admissions Many works apply statistical models and machine learning to undergraduate admissions (Bruggink and Gambhir 1996; Lux et al. 2016; Neda and Gago-Masague 2022; Kiaghadi and Hoseinpour 2023; Lee et al. 2023), and similar lines of work for graduate admissions (Moore 1998; Waters and Miikkulainen 2014; Gupta, Sawhney, and Roth 2016; Staudaher, Lee, and Soleimani 2020) and medical residency (Rees and Ryder 2023). For example, see Lee et al. (2024) and the references therein. Many of these works primarily demonstrate the possibility of modeling admissions outcomes with machine learning. Another motivation is auditing for bias, where statistical models of admission probability have played an important role, for example in the Supreme Court case *Students for Fair Admissions v. Harvard* (Gray et al. 2022).

In other cases, statistical models are geared towards deployment, though generally not to automate the entire decision process. Lee, Kizilcec, and Joachims (2023) describe a highly-selective college in the United States that uses a predictive model to provide coarse estimates that are used to group and rank the order of students' applications for human review. Models are also deployed to give guidance to the applicants, for example, to universities (Sirolly, Kanoria, and Ma 2024; Rauls 2021).

In the context of our empirical illustration, the New York City Department of Education deployed a predictive model in the 2024-2025 application year to assist students in assessing their odds of admission to high schools (Shen-Berro 2024). It is important to note that the model – which displays coarse outcome predictions (low, medium, high) – does *not* use machine learning, but rather directly simulates the (known) choice functions in use by schools. Importantly, this approach does *not* face the challenges we characterize in this work, since simulating the choice functions does take into account other applicants, for various assumptions on what the applicant pool would be. Such an approach is one path forward for prediction in admissions settings as opposed to standard machine learning approaches directly, though may not be feasible in settings where the choice functions are not directly known and can be simulated – for example, in this context, some schools use covariates unavailable to researchers, and so some machine learning component may be necessary. Our work evaluates the challenges of such a potential approach.

Pitfalls of Social Prediction Prediction of social outcomes, of which admissions is one example, can be fraught. In the context of education, Perdomo et al. (2025) argue that personalized predictions of high school graduation in Wisconsin do not outperform high school-level predictions and interventions. Liu et al. (2023) discuss the mismatch between supervised machine learning problem formulation and actual educator needs. Raji et al. (2022) argue that AI models are often erroneously assumed to be effective, when in reality they simply do not work as advertised. Wang et al. (2024) provide a list of reasons why predictive machine learning modeling may fail within their normative argument against 'predictive optimization' – for example, distribution shifts induce fundamental limits to prediction. To this literature, we contribute a novel *explanatory mechanism* driving poor predictive performance in admissions settings: cohort-dependence induced distribution shifts.

A large literature benchmarks (poor) performance of machine learning algorithms under distribution shift, and aims to develop distributionally robust algorithms (Gardner, Popovic, and Schmidt 2023; Koh et al. 2021; Yao et al. 2022). Such approaches often (implicitly or explicitly) specify a model of how distributions can shift at test time (Kaur, Kiciman, and Sharma 2023). To our knowledge, these approaches stay within the ML paradigm of modeling decisions as independent; our results further suggest that developing robust predictors in ML settings requires modeling how decisions depend on entire applicant pools.

Other recent works also study independence in decision-making in machine learning model. Most related, Dong et al. (2025) challenges independent decision-making – in the context of discretizing continuous scores for demographic imputation (e.g., race prediction); both works suggest that decisions for an individual data point should depend on the entire cohort (sample), but the rationales differ: in this work, the *true* decisions for an applicant depend on inference-time application pools; in Dong et al. (2025), the true labels (e.g., the demographics of an individual) are fixed, but independent decision-making biases the imputed label distribution and downstream tasks.

Finally, we note that a recent line of work studies the pitfalls (and potential benefits) of algorithmic monoculture, in which multiple machine learning models make correlated decisions *for the same data point* (Kleinberg and Raghavan 2021; Bommasani et al. 2022; Creel and Hellman 2022; Toups et al. 2023; Jain et al. 2024; Peng and Garg 2024a,b; Jain, Creel, and Wilson 2024; Kim et al. 2025); in contrast, this work studies the dependence of true labels *across* inference-time datapoints.

Formal Models of Choice and Admission The formalization of admissions processes ties into a long and broad economic literature on choice functions, preferences, and behavior (Arrow 1959; Sen 1971; Moulin 1985; Kalai, Rubinstein, and Spiegler 2002). The basic structure of choice functions have formed the basis for voting theory, utility theory, matching markets, and more. Much of the seminal choice function literature focuses on narrowing the vast space of possible choice functions by suggesting desirable properties and proving relations between them.

A related literature on affirmative action studies choice functions subject to diversity constraints—often assuming a given ordering of preferences over students (Echenique and Yenmez 2015; Celebi 2023; Arnosti, Bonet, and Sethuraman 2024). Much of this literature focuses on improving the definition of constraints or designing better mechanisms for implementing preferences subject to the constraints. In comparison, our work treats the choice function as a given part of the data generating process, and we explore questions of representation via machine learning. We relate representation ability to classic properties in economic models, such as substitutability.

3 Model and Setting

3.1 Machine Learning

Data from admissions processes are often used for supervised machine learning as follows: each applicant is a data point and admissions decisions are labels. Applicants are represented by the features $x \in \mathcal{X}$ on their application, and labels $y \in \{0, 1\}$ are binary (accept or reject). Then the relevant task is binary classification, i.e., generating a model $f : \mathcal{X} \to \{0, 1\}$ using data $X = \{x_1, x_2, ..., x_n\}$ and $Y = \{y_1, y_2, ..., y_n\}$ from past admissions processes.

Training a machine learning model means selecting a f from a set of possible models, i.e., a hypothesis class. Generally, this is done by finding a model which achieves low error on available data (a train set). It is typical to use continuous optimization algorithms to first train a continuous *score* function $s : \mathcal{X} \to [0, 1]$ by minimizing the error. Then the classification model is defined as

$$f(x) = \mathbf{1}\{s(x) \geq t\} \quad (1)$$

based on a fixed threshold value t, often $t = 0.5$. Notice that models of the form (1) predict outcomes independently for each applicant. At test time inference, they do not consider the overall applicant pool – they implicitly model the training time pool, through its effect on the training labels.

3.2 Choice Functions

Many admissions processes have capacity constraints (either perceived or actual). This fact gives rise to what we call *cohort-dependence*: an applicant's acceptance decision depends on not only their own characteristics, but also the number and characteristics of other applicants.

Choice functions formalize admission decision processes and allow modeling such dependence. A choice function describes the behavior of "choosing" elements from a set of available options, and is commonly used in economic models. Formally, a choice function over the universe $\mathcal{X}$ is a function $\mathcal{C} : 2^{\mathcal{X}} \to 2^{\mathcal{X}}$ that maps all subsets X of $\mathcal{X}$ to the accepted set $\mathcal{C}(X)$, where $\mathcal{C}(X) \subseteq X$. As our motivation focuses on real applicant datasets, we consider finite subsets.

Recalling admissions data, the binary admissions label y for student x is 1 if $x \in \mathcal{C}(X)$ and 0 otherwise. To aid in our discussion, we will slightly abuse notation to relate pairs (x_i, y_i) to the choice function $\mathcal{C}(X)$ where $x_i \in X$. Define $\mathcal{C}(X)_i$ as 1 if $x_i \in \mathcal{C}(X)$ and 0 otherwise; this allows us to write the label $y_i = \mathcal{C}(X)_i$. Notice that the dependence of labels on the applicant pool introduces a particular form of distribution shift called "concept drift," where the conditional distribution $y|x$ changes over time.[1] This occurs even when the choice function itself is held constant.

A basic property of choice functions we consider is that they are capacity constrained to accept at most q applicants.

Definition 1 (q-Acceptance)**.** *Choice function $\mathcal{C}$ is q-acceptant if $|\mathcal{C}(X)| = \min\{q, |X|\}$ for all input sets X.*

q-acceptant choice functions accept as many applicants as possible, up to q elements. Some (but not all) choice functions can be characterized as a total order.

Definition 2 (Total order)**.** *A* total order *is a transitive, asymmetric, and complete binary relation $\succ$ over elements of $\mathcal{X}$; that is, $x_1 \succ x_2$ implies $x_2 \not\succ x_1$, $x_1 \succ x_2$ and $x_2 \succ x_3$ implies $x_1 \succ x_3$, and all x_1, x_2 are comparable.*

Total orders are frequently used in modeling choice mechanisms; they are often called *preference orderings* or *priority orders* in economic models. Intuitively, total orders induce a ranking over candidates and could result from, e.g., ordering applicants based on their GPA. We will say that a q-acceptant choice function $\mathcal{C}$ is *characterized by a total order* if there exists a total order $\succ$ such that for any input X, the accepted applicants are the top-q elements of X ordered by $\succ$. In other words, $x_1 \succ x_2$ for all $x_1 \in \mathcal{C}(X), x_2 \notin \mathcal{C}(X)$. We will refer to the process of sorting elements according to a total order $\succ$ and selecting the top q as a *queue*. Not all choice functions are characterized by total orders; consider a program which admits the $\frac{q}{2}$ applicants with the highest English scores and $\frac{q}{2}$ of the rest with the highest Math scores.

[1] This contrasts with "covariate shift", where new datasets contain shift x distributions, but maintain the same relationship $y|x$.

4 Theoretical Analysis

We now present a theoretical framework for understanding why admission decisions are hard to predict with ML models, due to capacity constraint-induced dependence. We define two choice function properties: *instability* (the true labels of multiple data points can change when a new data point is added) and *variability* (*which* labels change can depend on the new data point), and characterize choice functions according to these properties. Proofs are in §A.

4.1 Instability and Variability

To characterize the instability of a choice function, we will measure the number of decisions which change as the applicant pool expands. To that end, we first define a distance between sets of applicants induced by the choice function.

Definition 3 (Choice Distance)**.** *For a choice function $\mathcal{C}$ and sets $X_1 \subseteq X_2$, we define the choice distance as:*

$$r_{\mathcal{C}}(X_1, X_2) := |X_1 \cap \mathcal{C}(X_2) \backslash \mathcal{C}(X_1)| + |\mathcal{C}(X_1) \backslash \mathcal{C}(X_2)| \quad (2)$$

This distance[2] measures the number of decisions that *change* when expanding the candidate pool from X_1 to X_2. The first term counts rejections that become acceptances; the second counts acceptances that become rejections.

Definition 4 (d-Instability)**.** *Choice function $\mathcal{C}$ is d-unstable if for all X, and $X' = X \cup \{x'\}$, we have $r_{\mathcal{C}}(X, X') \leq d$ for any nonnegative integer d. $\mathcal{C}$ is called tightly d-unstable if it is d-unstable but not $(d-1)$-unstable.*

Intuitively similar to a Lipschitz bound, note that because $r_{\mathcal{C}}$ obeys the triangle inequality (Theorem A.1), d-instability also guarantees at most dk changes when adding k elements. This definition focuses on changes only to *existing applicants'* decisions. Unlike in conventional statistical learning, this quantity does not relate to performance on new data points, but rather, isolates the effect of new data points on *existing* data points—an effect not traditionally considered.

While instability characterizes how many chosen elements change when perturbing the input set, we also consider *which* elements change. Even for a 1-unstable choice function, there is still substantial diversity in *which* decision might change. For example, in a school that simply selects the q students with the highest GPA, the same "borderline" applicant is displaced regardless of who the new (higher GPA) applicant is. In contrast, a school that selects half of their students based on an English test score and the other half based on Math would reject different students depending whether the new applicant excels at English or at Math.

We formalize[3] this intuition with the following definition.

[2] We use the term "distance" colloquially here; this definition does not fulfill the criteria to be a proper distance metric.

[3] For a more general definition of variability (beyond 1-unstable and q-acceptant), see §A.8.

Definition 5 (Variability)**.** *A* 1*-unstable, q-acceptant choice function* $\mathcal{C}$ *has variability* m *where*

$$m := \max_{X \subset \mathcal{X}} \left| \bigcup_{x' \in \mathcal{X}} \mathcal{C}(X) \setminus \mathcal{C}(X \cup \{x'\}) \right| \tag{3}$$

Variability bounds how many *different* currently accepted candidates could be displaced by adding any single new candidate. As we empirically demonstrate, these notions capture how sensitive decisions are due to applicant pool shifts.

4.2 Main Theoretical Results

We first relate the concepts of instability and variability to the practice of machine learning. We focus on the representation capabilities of ML models: when is it possible for an ML model to faithfully represent an admissions process? Formally, we say that an ML model can represent a choice function $\mathcal{C}$ if there exists a function f such that for all applicant sets X, $f(x_i) = \mathcal{C}(X)_i$. Note that by focusing on representation, our theory is distribution agnostic, and applies to settings where applicant distributions are affected considerations such as strategic behavior and matching algorithms.

Before stating this result, consider a straightforward way to adapt a model (1) to the capacity-constrained setting. Given that the ML model is based on continuous scores, we can adjust the decisions to account for a known capacity constraint if we have available the set of all test-time applications. In particular, we can rank the applicants by their scores and then select the top q. This process results in decisions determined by a cohort dependent threshold $t_q(X)$ which is the score of the qth ranked applicant:

$$f(x; X) = \mathbf{1}\{s(x) \geq t_q(X)\}. \tag{4}$$

Proposition 1 (ML Representation)**.** *A model of the form* (1) *which makes independent predictions can only represent 0-unstable choice functions. A model of the form* (4) *which ranks applicants can represent a 1-variable, 1-unstable choice function. No such model can represent a choice function with variability or instability greater than one.*

This result shows that standard ML practice coherently applies to admissions data only in limited settings. The intuition for this result hinges on the fact that all 1-unstable and 1-variable choice functions correspond to total preference orderings over applicants (formalized in Theorem 2). Then the proposition follows from noticing that total preference orderings correspond to embedding applicants in $\mathbb{R}$ (by their scores). How restrictive are the settings where naive ML applies? We next characterize the instability of choice functions. First, we introduce a choice function property which plays an important role in the school admission setting.

Definition 6 (Substitutability)**.** *Choice function* $\mathcal{C}$ *is substitutable if* $X_1 \subseteq X_2$ *implies* $X_1 \cap \mathcal{C}(X_2) \subseteq \mathcal{C}(X_1)$.

Substitutability means that removing other applicants from the pool cannot hurt an accepted applicant. Substitutability is well-established property in the choice literature (Chernoff 1954; Moulin 1985; Deng, Panigrahi, and Waggoner 2017). It is a necessary condition for the existence of a unstable matching (Roth 1984). Many common optimization objectives are substitutable; selecting applicants by a linear ranking, linear assignment problems, and selection problems optimally solved by greedy algorithms (Yokoi 2019) can all be represented as substitutable choice functions.

Theorem 1. *A* q*-acceptant choice function*

1. *cannot be* 0*-unstable,*
2. *is exactly* 1*-unstable if and only if it is substitutable,*
3. *can be tightly* d*-unstable for every* $1 \leq d \leq 2q$.

First, capacity constraints directly imply nonzero instability; a school will become more selective as the applicant pool widens. As a result, recalling Proposition 1, naive ML of the form (1) cannot represent such decision processes. Capacity constraints force decisions to be cohort-dependent, violating traditional independence assumptions.

Skipping to the third point, we show that there exist choice functions which are much more unstable. One intuitive class of highly unstable (large d) mechanisms is team formation. For example, a music department may want a balanced makeup of instrumentalists accepted to a program; the music director prefers equal numbers of violinists and cellists, but otherwise chooses the most talented musicians. Due to a lack of cellists, after admitting all auditioning cellists, there remain several violinists rejected, and additional slots are filled with, e.g., bassists. Adding an additional cellist to the applicant pool, however, leads to rejecting two bassists in favor of a previously rejected violinist and the new cellist—a 3-unstable function. If admissions instead prioritized string trios (a violin, viola, and cello), there might be *three* bassists rejected to form a trio (5-unstable), while forming a *quartet*, (a viola, cello, and two violins), is 7-unstable. We show in Lemma A.5 that complementary groups of size n can define functions that are tightly $2n - 1$ unstable. We also construct examples of instability $2n$ when additional applicants alter decisions without being accepted themselves in §A.6.

Going back to the second point of Theorem 1, we show that substitutability is equivalent to 1-instability under q-acceptance—violations of substitutability *must* lead to larger changes in admissions decisions, and vice versa. In other words, a broad class of functions are as unstable as possible, given that they are subject to capacity constraints.

Whether ML models of the form (4) – which rank continuous scores – can represent 1-unstable choice functions depends on their variability. Before turning to this characterization, we return to the idea of orderings and queues. Not all choice functions are characterized by a total order. However, a larger class of substitutable and q-acceptant choice functions can be defined as a combination of *multiple* queues. The simplest way of constructing this is by sequentially composing several choice functions together.

Definition 7 (Sequential Composition)**.** *A choice function* $\mathcal{C}$ *is a sequential composition of functions* $\mathcal{C}_n, \ldots, \mathcal{C}_1$ *if:*

$$\mathcal{C}(X) = \mathcal{C}_1(X) \cup \mathcal{C}_2(X_2) \cup \cdots \cup \mathcal{C}_n(X_n)$$

where $X_{i+1} = X_i \setminus \mathcal{C}_i(X_i)$ *and* $X_1 = X$.

For example, a school may first accept $q/2$ students with the highest English scores, and then $q/2$ students with the highest Math scores (who were not already admitted).

Theorem 2. *Consider a q-acceptant, 1-unstable choice function $\mathcal{C}$ which can be represented as the sequential composition of n choice functions, each characterized by a total order. Then $\mathcal{C}$ has a variability m, where $1 \leq m \leq n$. Furthermore, $\mathcal{C}$ has variability 1 if and only if it can be characterized by a single total order, $n = 1$.*

In the appendix, we discuss when this upper bound is tight. We also extend this result to choice functions which are not explicitly defined as a fixed sequential composition, such as linear assignment problems. Theorem A.11 addresses the context-dependent sequential queues which result from such optimization based admissions. Importantly, Theorem 2 means that a choice function can be faithfully represented by a machine learning model if and *only if* it is characterized by a total ordering over $\mathcal{X}$.

5 Application: NYC High School Admissions

We illustrate our theory on data from the NYC high school matching system. Using admissions and applicant data from the 2021-2022 and 2022-2023 admissions cycles,[4] we extract applicant pools and admissions for each individual program. As we cannot release the data, we further generate synthetic data to validate our results, and release our code. The code is available at https://github.com/evan-dong/admissions-prediction, and the replication results with synthetic data are in the appendix.

The motivation for this empirical demonstration is twofold. First, we seek to understand the challenges that may arise when designing interventions to help applicants make decisions; for example, New York City now provides (coarse) personalized admission likelihoods for each program to applicants (Shen-Berro 2024). As noted above, their approach avoids the challenges discussed here by directly modeling the choice functions. However, such a simulation approach may not be feasible in settings where the choice function is not precisely known, such as in holistic college admissions or hiring, or for schools that use features other than numeric ones, such as essays or auditions. Thus, second, this demonstration serves as an illustration of the challenges of prediction in such settings broadly – in a setting where, due to the ability to simulate outcomes, we can control for other reasons that such prediction may be difficult.

In this context, each student can apply to multiple programs and rank them in order of preference; in our data, they were limited to listing twelve programs. They are then matched to a single program according to the deferred acceptance algorithm. Programs admit students according to various criteria; for example, some schools admit students according to a musical audition or essays. The vast majority of programs use explicit and publicly available decision rules, for which we can construct counterfactual decisions according to alternative applicant pools;[5] we use a "simulator" developed by Peng et al. (2025), which implements the choice funtions used by a large set of programs. As in their work, we restrict our analysis to the most common categories of program admissions choice functions that we can replicate accurately based on the data.

In this section, we first apply our theoretical framework to choice functions used by NYC programs in practice, characterizing their instability and variability. Then, we show that these characteristics correspond to machine learning performance in predicting admissions outcomes over time.

5.1 Applying the Theoretical Framework

We first describe the most common categories of admissions functions used by NYC High School programs and then characterize them based on our instability and variability definitions. We analyze three types of programs: "Ed. Opt", "Screened," and "Open." Programs may also participate in the Diversity In Admissions (DIA) initiative. In the language of our model, "Screened" and "Open" programs use the same choice function class. Thus, we have four possible categories of functions: "Ed. Opt", "Screened/Open," "Ed. Opt with DIA", "Screened/Open with DIA."

At a high level, each function is a sequential composition of queues. Within each queue, applicants may be ranked according to a queue-specific score function, Ed. Opt. category, DIA-qualifying status, borough (neighborhood) of middle school attendance and residence, (discretized) grade tier, continuing student status, and a lottery (tiebreaker) number. Each queue also has a capacity (which we take to be fixed). Thus each queue corresponds to a choice function that is characterized by a total ordering. The queues themselves are ordered, and so admissions follows a sequential composition of these choice functions (Definition 7).

In a **Screened/Open** program, there is a single queue (ranking), with students ordered based on a score[6] determined by a combination of their grades, neighborhood and continuing-student priorities, and the lottery number. In a **Screened/Open with DIA** program, there are two queues; in one, all students with DIA-qualifying status are ranked above the remaining students (and then by the score function); in the other queue, all students are ranked by the score function. In an **Ed. Opt.** program, there are three queues, respectively corresponding to whether a student had *low*, *medium*, or *high* grades in middle school[7] – students with these grades are ranked at the top in their respective queues (and otherwise by a score determined by their lottery number and neighborhood and continuing-student priorities). Finally, in **Ed. Opt with DIA**, there are six queues, each corresponding to a Ed Opt. grade tier plus DIA-qualifying status.

[4] We received the data from the New York City Board of Education through a research data use agreement process, to study application behavior and behavioral interventions in the application process. This data is available to researchers with a sponsor inside the Department of Education. Due to a non-disclosure agreement and to protect private student data, we cannot release the data. The research was deemed exempt by our university's Institutional Review Board.

[5] The DOE publishes these criteria online (NYC DOE 2022).

[6] Open programs differ from Screened programs by defining a score which does not depend on grades.

[7] The intention is to induce diversity in terms of academic levels.

We can thus apply our theoretical framework as follows.

Proposition 2. *All considered choice functions (Ed. Opt, Screened/Open, Ed. Opt with DIA, and Screened/Open with DIA) are* 1*-unstable. Furthermore, their variability is (tightly) equal to their number of queues:*

- *Screened/Open programs are* 1*-variable*
- *Screened/Open with DIA programs are* 2*-variable*
- *Ed. Opt. programs are* 3*-variable*
- *Ed. Opt with DIA programs are* 6*-variable*

5.2 Empirical Methodology

Our goal is to illustrate that capacity constraints—and shifts in the applicant pools over time—contribute to the difficulty of predicting application outcomes. For this, we require a dataset in which we have outcomes for applicants from the same choice functions but with different applicant pools. It is also important to isolate the effect of shifting applicant compositions from other reasons that application outcomes may be hard to predict. These include changes to the decision process and choice function complexity more broadly (e.g. nonlinearity or high dimensionality). Finally, we want a range of choice functions, including those with different instability and variability. In other words, we require, for each school program: a consistent *choice function* used on different *application pools* over time, with *admissions outcomes for each applicants* given the pool. Then, we can train *machine learning models* on outcomes from one application pool, and evaluate their performance on outcomes from different pools. We note that such data is in general difficult to obtain. We now describe each component.

At a high level, our empirical approach is designed to make prediction as *easy* as possible—to remove all reasons that prediction may be difficult, other than the applicant pool composition effect that is the focus of this work; as we illustrate, prediction is still difficult over time for choice functions with poor instability or variability characteristics.

Programs, applicants, and applicant pools We use data from the General Education match in the 2021 and 2022 New York City High School match process, comprising of 58,500 students in 2021 and 57,331 students in 2022. We consider their applications to 199 unique programs: 99 that are Ed. Opt, 75 that are Screened/Open, 5 that are Ed. Opt with DIA, and 20 that are Screened/Open with DIA.[8]

The matching mechanism uses the "deferred acceptance" algorithm: students submit rankings over programs and programs' choices are determined according to the above defined policies. In short, the algorithm iteratively selects a student who is not tentatively accepted to a program, and that student "applies" to the top program in their list to which they have not yet been rejected. Given its current set (its current tentatively accepted students and the new applicant), the program may tentatively accept or (permanently) reject the new applicant, and may (permanently) reject a current tentatively accepted applicant. This algorithm proceeds until all students are tentatively accepted somewhere or have exhausted their submitted preferences. This algorithm means that a student truly "applied" to and was accepted by the program they were matched to, "applied" to and was rejected by any program they ranked above their match, and have indeterminate status at the remaining programs, since they never actually "applied" in the deferred acceptance sense.

For our analysis, we do not consider the deferred acceptance algorithm. Rather, we consider each program separately, as making admissions decisions according to a choice function and given an applicant pool. For each program, we consider as the potential pool those students who truly applied (*in the deferred acceptance sense*) to that program in either 2021 or 2022.[9] Based on the above process, we have a total of 107,128 and 104,501 applications for 2021 and 2022 respectively from the chosen students to the given programs.

Choice functions and outcomes We now describe how we construct outcomes for each applicant to a program given a pool. We adapt a *simulator* developed by Peng et al. (2025) to generate admissions decisions: for each program in each year, the rule-based simulator implements the admissions function defined by publicly available policies. Then, given an application pool and a fixed capacity (which we infer from actual outcomes), we can calculate outcomes for each student. This simulator is implemented using 2022 decision policies and accurately[10] reflects actual admission outcomes when applied to the application pools and outcomes from both 2022 (98.79% accuracy) and 2021 (91.47% accuracy).

Why don't we simply use the actual admissions outcomes of each applicant to a program in 2021 and 2022, since we use the *real* applicant pool as data and have access to these outcomes? First, as described above, real-world admissions functions may change from year to year, and our interest here is in studying inherent difficulties of learning admissions functions even when they remain the same. Second, we wish to also display the effect of small changes to the applicant pool (such as substituting a few students), instead of the full changes between years. Third, this approach allows evaluating the predictability of new choice functions, not currently used by any programs (including those that are not 1-unstable), under real applicant pool shifts over time. Similarly, we can apply the same choice functions to all programs, thus controlling for features (such as overall admission rates) that correlate with both prediction difficulty and the choice function a program actually used.[11]

[8] We start with $n = 587$ programs that are either Ed. Opt 398, Screened 129, or Open (60) and existed in both 2022 and 2021. From there, we further restrict our dataset to programs that matched with at least one student in each year ($n = 573$) with a nontrivial offer rate (i.e., rejected at least one student in each year) ($n = 199$).

[9] This can roughly be viewed as understanding the decisions within one step of the deferred acceptance algorithm, in which an applicant "proposes" to a school, which either rejects them immediately or tentatively adds them to their acceptance pool, given the current acceptance pool.

[10] Accuracy is not 100% because there are additional criteria that apply to a small number of students and for which we do not have data (e.g., children of teachers or siblings of current students).

[11] For example, if program A originally admitted students based on a screened admissions method, we can simulate the admissions outcomes had program A instead been an Ed. Opt. or Open pro-

In other words, a simulator allows us to generate counterfactual decisions—we can modify the applicant pool or the underlying choice function. It further allows us to control for other factors (aside from our focus on shifts in the applicant pool composition) that may make predicting admissions outcomes difficult, cf. Wang et al. (2024). By generating every combination of admissions method with every program (for a total of six; three admissions methods, with or without 50% of seats being reserved for the DIA initiative), we can examine the effect of different admissions methods. We also isolate the impact of instability- and variability-related differences in methods in §C.8 by simulating admissions without borough or continuing student priority. We always use the real applicant pools to each program in each year. This allows us to capture the effects of real shifts in applicant pool composition on model accuracy.

Synthetic Admissions Methods. Notably, all choice functions from the programs we analyze are 1-unstable. To further test the theory, we design 0- and 5-unstable methods that we implement in our simulator. We create two different 0-unstable admissions methods not subject to capacity constraints; the first simply admits every student with a tiebreaker number below fixed threshold (calibrated so that approximately half the overall applicants will be admitted).The second similarly thresholds tiebreaker number, but with three different values depending on an applicant's Ed. Opt. category. Our 5-unstable admission method is detailed in §C.1.

Machine learning predictions of admissions outcomes The above processing results in a set of programs, real applicant pools in each year for each program, and admissions outcomes under each choice functions and given each applicant pool. We now describe our ML training and evaluation.

Training and test sets. We aim to replicate real-world ML performance and training: when a model is trained on admissions outcomes from one time period, and then used to predict outcomes in a future time period. Crucially, all admissions outcomes in the training data are based on the train-time applicant pool, while the machine learning model will be used to predict outcomes for students given a future applicant pool. For example, we may train a model on 2021 applicant outcomes, to give advice to applicants as part of the 2022 process. In each experiment, the training data labels are always defined by the training set applicant pools, while the test labels are defined by test applicant pools. For robustness, we provide results when flipping the training and test years in §C.6. To control for the exacerbating effects of distribution shift, we show in-distribution, out-of-sample performance using our synthetic data generator in §C.10.

Training and evaluation. We train a separate machine learning model for each program and admissions function using the actual training set applicant pools and outcomes. We tailor the feature space (removing irrelevant features and defining interaction terms) to each admission function to

gram, with or without DIA-reserved seats. This allows us to deconfound choice function predictability from other correlations between the applicant pool and the method—for example, many of the most selective programs are screened programs with DIA.

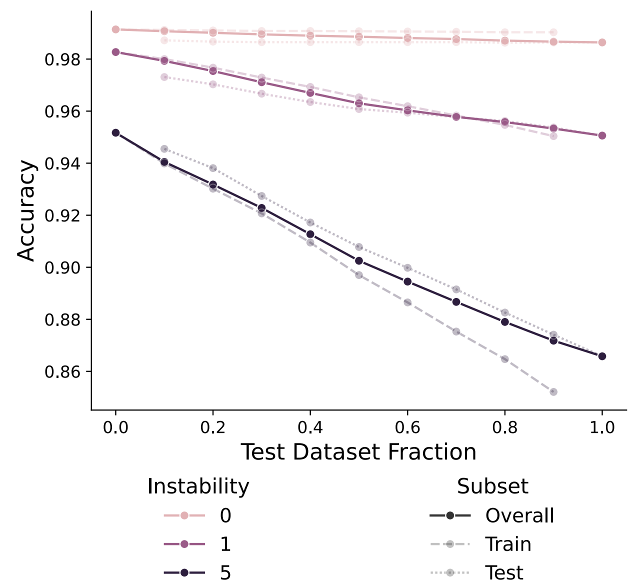

Figure 1: Model accuracy as a function of choice function instability, under increasing levels of distribution shift.

simplify prediction and enable near-perfect learning at train time. We train logistic regression models with L_2 regularization. We define prediction by discretizing the continuous $[0, 1]$ scores. We rank applicants according to their score and predict 1 for the top q, as defined in (4). Empirically, we find that this improves average performance compared with using a fixed threshold as in (1), which is not surprising in light of Proposition 1. Results with fixed thresholds and monotonic gradient boosting are in §C.4 and §C.5.

Experiments Our main experiments involve evaluating the performance of models trained on data from 2021 on a series of slowly shifting applicant pools. We generate mixture pools from the 2021 and 2022 applicant pools for each school where each such pool is made of a randomly sampled γ fraction of the applicants from 2021 and $1-\gamma$ of the applicants from 2022, for $\gamma = \{0, .1, .2, ..., 1\}$. By construction, there are both in-distribution (in-sample) and out-of-distribution applicants in these mixtures. This allows us to evaluate how the addition of out-of-distribution applicants affects even in-sample performance. Due to stochasticity in model performance from sampling the interpolated datasets, we average performance metrics across 10 random trials. We hold admission rate constant across interpolated datasets, adjusting capacity. Results where all datasets are held at an exact fixed size are in §C.7.

5.3 Empirical Results

Instability Figure 1 illustrates model accuracy for each γ-mixture applicant pool, averaged over all programs and choice functions with the same instability value. Model performance decays more steeply with out-of-distribution test data, with more stable choice functions. Furthermore, the decay does not occur just on out-of-distribution points—as

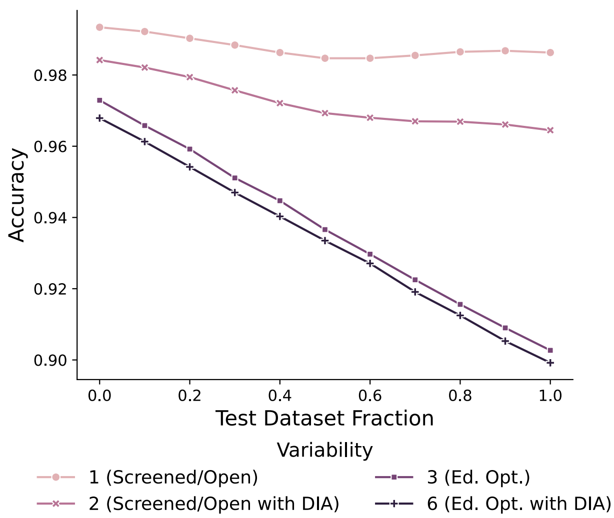

Figure 2: Model accuracy as a function of choice function variability, under increasing levels of distribution shift.

would occur in standard distribution shift settings with independent outcomes. Figure 1 also plots the accuracy on the train and test subsets of the mixture pool separately. In conventional machine learning models of distribution shift, performance on the in- and out-of-distribution subsets remains fixed. Indeed, for 0-unstable functions, performance for both subsets remains relatively constant. In contrast, we observe that *both* in- and out-of-distribution performance degrades for $d = 1$ and $d = 5$, with the change being more drastic for a less unstable function. At the same time, performance in the *testing* subset likewise drops *even as it takes up a larger portion of the overall dataset*—because the model was trained using training set application set outcomes.

Variability We now focus on 1-unstable admissions functions, corresponding to actual decision processes used in NYC high school programs. Figure 2 shows averaged performance for admissions method grouped by variability. Programs with higher variability exhibit larger decays in performance. Notably, the 1-variable programs barely decay, in line with Proposition 1. Larger values of variability have larger performance drops, though to a diminishing extent; the gap between variability 3 and 6 is small. Our definition of variability is "worst case" over applicant sets, so we hypothesize that this observation is due to the fact that actual applicant sets are not adversarial, even as they exhibit distribution shift.

A more detailed breakdown of performance by admissions method can be found in §C.3.

6 Conclusion

We provide one *explanatory mechanism* for why prediction admissions outcomes is hard: outcomes depend on the applicant pool, which shift over time, inducing concept drift—this dependence cannot be represented by the standard ML paradigm. As we empirically show, this dependence degrades machine learning performance, even when other challenges in this context (strategic behavior, data limitations, other forms of distribution shift) are removed.

One natural question for future work is: how do we adapt ML models to be robust to this effect? In contexts where the choice functions are algorithmic and exactly known, one can avoid training ML models and simulate the choice functions directly—in fact, this approach is the one taken for NYC High School admissions predictions, for programs in which the choice function can be simulated under alternative application pools. However, for other contexts where we cannot simulate the choice function – such as hiring and admissions by humans – it may be useful to develop approaches to quantify uncertainty to applicant pool changes or otherwise produce predictions robust to it, rather than naively applying conventional predictive approaches.

Finally, we note that similar concerns – the effects of capacity constraints on predictive model performance – may also be present in other contexts such as healthcare, e.g., hospital admissions.

Acknowledgments

ED is supported by National Science Foundation grant DGE-2139899. NG's work is supported by NSF CAREER IIS-2339427, NASA, the Sloan Foundation, and Cornell Tech Urban Tech Hub, Google, Meta, and Amazon research awards. SD is funded by NSF CCF 2312774, NSF OAC-2311521, NSF IIS-2442137, a gift to the LinkedIn-Cornell Bowers CIS Strategic Partnership, the AI2050 Early Career Fellowship program at Schmidt Sciences, and a PCCW Affinito-Stewart Award.

References

Arnosti, N.; Bonet, C.; and Sethuraman, J. 2024. Explainable affirmative action. In *Proceedings of the 25th ACM Conference on Economics and Computation*, 310–310.

Arrow, K. J. 1959. Rational choice functions and orderings. *Economica*, 26(102): 121–127.

Bommasani, R.; Creel, K. A.; Kumar, A.; Jurafsky, D.; and Liang, P. S. 2022. Picking on the same person: Does algorithmic monoculture lead to outcome homogenization? *Advances in Neural Information Processing Systems*, 35: 3663–3678.

Bruggink, T. H.; and Gambhir, V. 1996. Statistical models for college admission and enrollment: A case study for a selective liberal arts college. *Research in Higher Education*, 37(2): 221–240.

Celebi, O. 2023. Diversity preferences, affirmative action and choice rules. *arXiv preprint arXiv:2310.14442*.

Chernoff, H. 1954. Rational selection of decision functions. *Econometrica: journal of the Econometric Society*, 422–443.

Creel, K.; and Hellman, D. 2022. The algorithmic leviathan: Arbitrariness, fairness, and opportunity in algorithmic decision-making systems. *Canadian Journal of Philosophy*, 52(1): 26–43.

Deng, Y.; Panigrahi, D.; and Waggoner, B. 2017. The complexity of stable matchings under substitutable preferences. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.

Dong, E.; Schein, A.; Wang, Y.; and Garg, N. 2025. Addressing discretization-induced bias in demographic prediction. *PNAS Nexus*, 4(2): pgaf027.

Echenique, F.; and Yenmez, M. B. 2015. How to control controlled school choice. *American Economic Review*, 105(8): 2679–2694.

Gardner, J.; Popovic, Z.; and Schmidt, L. 2023. Benchmarking distribution shift in tabular data with tableshift. *Advances in Neural Information Processing Systems*, 36: 53385–53432.

Gray, C. B.; McCotter, R. T.; Berry, J.; Buschbacher, M.; and Kelson, J. M. 2022. Brief of Economists as Amici Curiae in Support of Petitioner. Students for Fair Admissions, Inc. v. President and Fellows of Harvard College and SFFA v. University of North Carolina (UNC), Supreme Court of the United States, 2023.

Gupta, N.; Sawhney, A.; and Roth, D. 2016. Will I get in? modeling the graduate admission process for American universities. In *2016 IEEE 16th international conference on data mining workshops (ICDMW)*, 631–638. IEEE.

Jain, S.; Creel, K.; and Wilson, A. 2024. Scarce Resource Allocations That Rely On Machine Learning Should Be Randomized. *arXiv preprint arXiv:2404.08592*.

Jain, S.; Suriyakumar, V.; Creel, K.; and Wilson, A. 2024. Algorithmic pluralism: A structural approach to equal opportunity. In *Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency*, 197–206.

Kalai, G.; Rubinstein, A.; and Spiegler, R. 2002. Rationalizing choice functions by multiple rationales. *Econometrica*, 70(6): 2481–2488.

Kaur, J. N.; Kiciman, E.; and Sharma, A. 2023. Modeling the Data-Generating Process is Necessary for Out-of-Distribution Generalization. In *The Eleventh International Conference on Learning Representations*.

Kiaghadi, M.; and Hoseinpour, P. 2023. University admission process: a prescriptive analytics approach. *Artificial Intelligence Review*, 56(1): 233–256.

Kim, E. M.; Garg, A.; Peng, K.; and Garg, N. 2025. Correlated Errors in Large Language Models. In *Forty-second International Conference on Machine Learning*.

Kleinberg, J.; and Raghavan, M. 2021. Algorithmic monoculture and social welfare. *Proceedings of the National Academy of Sciences*, 118(22): e2018340118.

Koh, P. W.; Sagawa, S.; Marklund, H.; Xie, S. M.; Zhang, M.; Balsubramani, A.; Hu, W.; Yasunaga, M.; Phillips, R. L.; Gao, I.; et al. 2021. Wilds: A benchmark of in-the-wild distribution shifts. In *International conference on machine learning*, 5637–5664. PMLR.

Lee, H.; Kizilcec, R. F.; and Joachims, T. 2023. Evaluating a learned admission-prediction model as a replacement for standardized tests in college admissions. In *Proceedings of the Tenth ACM Conference on Learning@ Scale*, 195–203.

Lee, J.; Harvey, E.; Zhou, J.; Garg, N.; Joachims, T.; and Kizilcec, R. F. 2024. Ending affirmative action harms diversity without improving academic merit. In *Proceedings of the 4th ACM Conference on Equity and Access in Algorithms, Mechanisms, and Optimization*, 1–17.

Lee, J.; Thymes, B.; Zhou, J.; Joachims, T.; and Kizilcec, R. F. 2023. Augmenting holistic review in university admission using natural language processing for essays and recommendation letters. *arXiv preprint arXiv:2306.17575*.

Liu, L. T.; Wang, S.; Britton, T.; and Abebe, R. 2023. Reimagining the machine learning life cycle to improve educational outcomes of students. *Proceedings of the National Academy of Sciences*, 120(9): e2204781120.

Lux, T.; Pittman, R.; Shende, M.; and Shende, A. 2016. Applications of supervised learning techniques on undergraduate admissions data. In *Proceedings of the ACM International Conference on Computing Frontiers*, 412–417.

Moore, J. S. 1998. An expert system approach to graduate school admission decisions and academic performance prediction. *Omega*, 26(5): 659–670.

Moulin, H. 1985. Choice functions over a finite set: a summary. *Social Choice and Welfare*, 2(2): 147–160.

Neda, B. M.; and Gago-Masague, S. 2022. Feasibility of machine learning support for holistic review of undergraduate applications. In *2022 International Conference on Applied Artificial Intelligence (ICAPAI)*, 1–6. IEEE.

NYC DOE. 2022. Fall 2022 High School Data. InfoHub Directory Data.

Peng, K.; and Garg, N. 2024a. Monoculture in Matching Markets. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Peng, K.; and Garg, N. 2024b. Wisdom and Foolishness of Noisy Matching Markets. In *Proceedings of the 25th ACM Conference on Economics and Computation*, 675–675.

Peng, K.; Ryu, E.; Kleinberg, J.; Tardos, E.; and Garg, N. 2025. Undermatching in New York City School Choice: Application Behavior and the Potential of Personalized Feedback. *arXiv preprint*.

Perdomo, J. C.; Britton, T.; Hardt, M.; and Abebe, R. 2025. Difficult lessons on social prediction from wisconsin public schools. In *Proceedings of the 2025 ACM Conference on Fairness, Accountability, and Transparency*, 2682–2704.

Raji, I. D.; Kumar, I. E.; Horowitz, A.; and Selbst, A. 2022. The fallacy of AI functionality. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, 959–972.

Rauls, L. 2021. Is CollegeVine's Chancing Engine Actually Accurate?

Rees, C. A.; and Ryder, H. F. 2023. Machine learning for the prediction of ranked applicants and matriculants to an internal medicine residency program. *Teaching and Learning in Medicine*, 35(3): 277–286.

Roth, A. E. 1984. Stability and polarization of interests in job matching. *Econometrica: Journal of the Econometric Society*, 47–57.

Sen, A. K. 1971. Choice functions and revealed preference. *The Review of Economic Studies*, 38(3): 307–317.

Shen-Berro, J. 2024. NYC Education Department developing tool to help applicants gauge high school admissions chances. *Chalkbeat New York*. Accessed: 2024-10-20.

Sirolly, A.; Kanoria, Y.; and Ma, H. 2024. The Impact of Race-Blind and Test-Optional Admissions on Racial Diversity and Merit. In *Proceedings of the 25th ACM Conference on Economics and Computation*, 308–308.

Staudaher, S.; Lee, J.; and Soleimani, F. 2020. Predicting Applicant Admission Status for Georgia Tech's Online Master's in Analytics Program. In *Proceedings of the Seventh ACM Conference on Learning@ Scale*, 309–312.

Toups, C.; Bommasani, R.; Creel, K.; Bana, S.; Jurafsky, D.; and Liang, P. S. 2023. Ecosystem-level analysis of deployed machine learning reveals homogeneous outcomes. *Advances in Neural Information Processing Systems*, 36: 51178–51201.

Wang, A.; Kapoor, S.; Barocas, S.; and Narayanan, A. 2024. Against predictive optimization: On the legitimacy of decision-making algorithms that optimize predictive accuracy. *ACM Journal on Responsible Computing*, 1(1): 1–45.

Waters, A.; and Miikkulainen, R. 2014. Grade: Machine learning support for graduate admissions. *Ai Magazine*, 35(1): 64–64.

Yao, H.; Choi, C.; Cao, B.; Lee, Y.; Koh, P. W. W.; and Finn, C. 2022. Wild-time: A benchmark of in-the-wild distribution shift over time. *Advances in Neural Information Processing Systems*, 35: 10309–10324.

Yokoi, Y. 2019. Matroidal choice functions. *SIAM Journal on Discrete Mathematics*, 33(3): 1712–1724.