Deep Learning for Investigating Causal Effects with High-Dimensional Data: Analytic Tools and Applications to Educational Interventions

by

Alberto Guzman-Alvarez

Bachelor of Science, University of California - Davis, 2013

Master of Arts, University of Pittsburgh, 2018

Submitted to the Graduate Faculty of the

School of Education in partial fulfillment

of the requirement for the degree of

Doctor of Philosophy

University of Pittsburgh

2023

UNIVERSITY OF PITTSBURGH

SCHOOL OF EDUCATION

This dissertation was presented

by

Alberto Guzman-Alvarez

It was defended on

March 28, 2023

and approved by

Co-Thesis Advisor: Lindsay C. Page, Associate Professor, Brown University

Co-Thesis Advisor: Xu Qin, Assistant Professor, University of Pittsburgh

Richard Correnti, Professor, University of Pittsburgh

Paul W. Scott, Assistant Professor, University of Pittsburgh

Deep Learning for Investigating Causal Effects with High-Dimensional Data: Analytic Tools and Applications to Educational Interventions

Alberto Guzman-Alvarez, PhD

University of Pittsburgh, 2023

Recent developments in machine learning have the potential to revolutionize quantitative education research. However, realizing this potential requires bridging the worlds of educational research and computer science. In my dissertation, I merged advances in deep learning and causal inference to enable researchers to assess program impacts using quasi-experimental methods with high-dimensional data.

My first dissertation paper proposes a new analytical procedure that incorporates deep neural networks to estimate propensity scores, which flexibly accommodate high-dimensional data and complex relationships between treatment selection and observable characteristics using propensity score weighting. In my analysis, I find that while logistic regression leads to low bias and small standard errors in the estimated average treatment effect in a low-dimensional data setting, machine learning approaches, particularly my deep neural network approach and bagged-CART, perform better in the high-dimensional settings.

In addition to the methodological contributions, my dissertation makes substantive contributions to the applied literature. In my second dissertation study, I evaluate a large-scale A.I. chatbot college access intervention that offered critical supports to historically and economically marginalized high school students to ease their transition into college during the COVID-19 pandemic. The study sheds light on the intervention's effectiveness and its potential for improving educational outcomes during the pandemic.

Overall, this dissertation advances the field by demonstrating the potential of machine learning and causal inference methods to advance quantitative education research. It provides a new approach for estimating propensity scores that can be used in high-dimensional settings, thereby improving the accuracy and reliability of impact assessments. The findings from the evaluation of the college access intervention offer important insights into how such programs can support students during challenging times and improve their educational outcomes, particularly for those who face systemic barriers.

Table of Contents

List of Tables

List of Figures

Acknowledgements

This dissertation is dedicated to mis padres (Eloisa Guzman and Antonio Alvarez). I am indebted to them for the sacrifices and injustices they faced when they came to this country as teenagers, and for the love and support they provided for me and my siblings. Gracias a ti, mamá, por todo el amor y cariño que me has dado toda mi vida, y los sacrificios que hiciste por los otros para darles una mejor vida. Gracias a ti, papá, por tu apoyo a nuestra educación y por enseñarnos que con educación podemos mejorar.

To my sisters (Stephanie, Adriana, Jacky), I can't express how much your support and love have meant to me throughout graduate school. This dissertation is as much yours as it is mine. To our dogs, Chico and Cookie, thank you for always greeting me with lots of kisses and tail wags when I would visit home after a long flight from Pittsburgh.

To my better half, Mike, thank you for all the love and support. During my lowest times in graduate school, you were always there to lift me up. You were the constant motivation when I felt like I couldn't write this dissertation, and you were always there to cheer me up. I love you.

To my advisor, Lindsay Page, thank you for your support and mentorship throughout these years. You have taught me how to be a good researcher, but most importantly, how to be a good colleague. Thank you for opening doors to opportunities I found unimaginable. My accomplishments are yours as well. To my co-advisor, Xu Qin, thank you for making me a better methodologist and for treating me like a peer when I didn't feel deserving. To Paul Scott, thank you for your friendship. You were such a welcoming presence when I first came to the Research Methods Department, and in you, I found a mentor and friend. To Rip Correnti, thank you for adopting me into the LSAP family and for your support throughout my time in LSAP. To Gina Garcia, thank you for being such a cheerleader to me and helping me remember to be my

own authentic self even in these academic spaces.

In addition, I have tremendous gratitude for the friends I have made during my graduate career. To emily H., thank you for your friendship. I can't describe how you have helped me process the roller coaster that is grad school. To my academic siblings Danielle, Aizat, and Aaron, thank you for helping me navigate graduate school and for your years of friendship. To Jordan, Lexy, Emily K., Eben, thank you for your friendship and the beers we shared.

After six cloudy years, Mike and I moved to Davis, CA, where I wrote the bulk of my dissertation. Thank you to my UC Davis academic family Michal Kurlander, Marcella Cuellar, Paco Martorell, and the new friends I made in the School of Education - Jaime, Mayra, Teresita, and Robbie. To Devon, thank you for hyping me up when I most needed it. Coming back to California allowed me to be closer to my close friends Marina, Kaitlin, Luis, Stephanie, Amy, and Alisa. Thank you for your support throughout these years, even when I was so far away.

Finally, I want to thank the National Academy of Education and the Spencer Foundation for their generous support in funding my dissertation work, and the Capital One Foundation.

To everyone, words cannot express the gratitude I have for your friendship and support. ¡Gracias!

1.0 Introduction

Over the last two decades, causal inference has played a central role in education research by providing tools and analytic strategies for generating evidence on the effectiveness of educational interventions and policies. The demand for rigorous evaluations by policymakers and funding agencies has driven this shift (Murnane & Willett, 2010; Rosenbaum, 2010), which has also seen an increase in sophisticated machine-learning approaches and access to high-dimensional administrative data (Einav & Levin, 2014; Song & Coleman, 2020). These advances have created a need to develop analytic techniques that can effectively handle the complexities of high-dimensional data, which is especially important for social science researchers that have access to observational data containing hundreds of covariates of student characteristics and are using causal inference methods to evaluate the effectiveness of educational interventions.

However, as more data on students become available, existing causal inference methods, many of which rely on parametric models, face challenges when applied to high-dimensional data, such as overfitting and variable selection (Collier et al., 2021; Hill et al., 2011; Keller et al., 2015; Song & Coleman, 2020). Overfitting occurs when a model is fit to noise rather than the actual relationship between covariates, resulting in inaccurate coefficient estimates (Kuhn et al., 2013). Variable selection is also challenging, given that it is hard to determine the most important variables in a large dataset, resulting in biased estimates (Buhlmann & Geer, 2011). Additionally, processing high-dimensional data requires significant computing power, making it challenging to fit models and perform analyses efficiently (Efron, 2014; Hill et al., 2011; Reardon & Stuart, 2019).

Deep neural networks (DNNs) are artificial intelligence algorithms inspired by the human brain's neural architecture; they represent a potential solution to the above challenges, thanks to advances in computing and software (Farrell et al., 2021; LeCun et al., 2015; Pang et al., 2019). DNNs are widely used in various industries for complex prediction and classification tasks and have proven effective in modeling complex high-dimensional data. However, there has been little application of DNNs to causal inference, particularly in social science research.

Randomized Control Trials (RCTs) have been the traditional approach to address questions of causality. In RCTs, students are randomly assigned to either a treatment or control group (Rosenbaum, 2010). The randomization process ensures that, on average, both groups are balanced in terms of both *observable* and *unobservable* characteristics, ruling out the possibility that any intervention impact is due to differences in group composition (Rosenbaum, 2010). However, conducting RCTs may not always be feasible for ethical, financial, or practical reasons. In such cases, researchers must rely on observational or non-experimental studies in which students either self-select or are placed in an intervention. To estimate unbiased treatment effects using observational data, researchers use quasi-experimental methods to balance *observable* characteristics between treatment groups.

Propensity score analysis is the most widely used quasi-experimental method in education research that balances *observed* characteristics between treated and untreated students by conditioning on what Rosenbaum and Rubin call a propensity score (Rosenbaum, 1987; Rosenbaum & Rubin, 1981, 1984). The propensity score represents the probability that a student would have been exposed to treatment, conditional on observable student characteristics (Rosenbaum, 1987; Rosenbaum & Rubin, 1981, 1984).

However, certain strict assumptions must be met to estimate unbiased treatment effects correctly using propensity score analysis. For example, the propensity score model must:

(1) Capture all the covariates related to the selection into treatment (Rosenbaum, 2010).

(2) Correctly model the association between the student's characteristics and treatment selection, meaning that all proper interactions and non-linear terms should be specified (Rosenbaum, 2010).

The literature suggests a "kitchen sink" approach for variable selection to meet the first condition best since penalties are high if a potential confounder is not included in the propensity score model (Karim et al., 2018; Webster-Clark et al., 2021). For the second condition, the literature suggests iterating through various model specifications. Each iteration adds interactions or non-linear terms until a balance is achieved among the observed characteristics between treated and untreated students (Rosenbaum, 2010). If these assumptions are unmet, the consequences are steep, leading to biased treatment effect estimates (Rosenbaum, 1987; Rosenbaum & Rubin, 1981, 1984).

The literature provides few methods for the critical step of estimating the propensity score in a high-dimensional setting (Wyss et al., 2018). The available research base suggests that the most widely used models for estimating the propensity score in education, logistic regression models, lead to biased treatment effect estimates when incorrectly specified (Lee et al., 2010; Lee, 2023; Lee & Little, 2017; Stuart, 2010). When modeling high-dimensional data, logistic models tend to overfit the data or cause issues with perfect separation, such that the logistic regression returns fitted probabilities that are either 0 or 1 (Hill et al., 2011). Therefore, an open question is whether DNNs represent a viable and worthy alternative for estimating propensity scores.

DNNs have already been shown to outperform logistic models and machine learning algorithms in simulation studies outside of propensity score analyses (Farrell et al., 2021; LeCun et al., 2015; Pang et al., 2019). DNNs represent an essential tool for researchers in estimating the correct propensity score model, in general, and mainly when dealing with large datasets that include hundreds of variables on students and when those variables have complex associations. Therefore, the fundamental question that motivates my dissertation is:

> How can we improve propensity score estimation in large observational datasets with modern DNN techniques?

My dissertation includes two studies: a methodological and a substantive application. In my first study, I developed and assessed a DNN-based approach for propensity score estimation against the traditional logistic model and other machine learning methods with high-dimensional data, using propensity score weighting. In my second study, I use propensity score matching to evaluate a large-scale text message campaign focused on helping students navigate college-going tasks during the COVID-19 pandemic. I did not use my DNN-based approach to evaluate this intervention for two reasons: (1) This evaluation is a rare case where the treatment assignment mechanism is known; specifically, students in the intervention had to be first-generation and qualified for a fee waiver. (2) Although we had access to a rich set of covariates after data cleaning, we only deemed 22 covariates appropriate for estimating the propensity score, negating the use of the DNN-based approach. This raises the important point that DNN-based approaches are not

necessarily needed in every application. Therefore, in my first paper, I aim to inform questions regarding the conditions under which DNN methods outperform more traditional approaches. What follows is a brief overview of my dissertation papers.

Paper 1: Evaluating modern propensity score estimation methods with high-dimensional data

Estimating the propensity score is a crucial step in propensity score analysis. Incorrect specifications of the propensity score model can result in biased treatment effect estimates (Murnane & Willett, 2010; Rosenbaum, 2010). Therefore, in this simulation study, I introduce DNNs into the field of education research and show that their unique properties, such as flexibility and lax model assumptions, are suitable and appropriate for propensity score estimation, particularly in the context of high-dimensional data.

In my first study, I assess the performance of the traditional logistic model and other machine learning methods with high-dimensional data using propensity score weighting. In addition, I overcome the limitations of existing methods by proposing a novel DNN-based approach to the estimation of propensity scores and illustrating a novel framework for generating multi-type, correlated high-dimensional data replicating the high-dimensional administrative data that is becoming available to education researchers.

Paper 2: When in-person support is not possible, can virtual outreach help? Evaluating the impact of an artificially intelligent conversational chatbot to promote college enrollment during the COVID-19 pandemic

The pandemic increased uncertainty for the class of 2020 in what ordinarily would be their timely transition to college. In response to this challenge, the Common Application (Common App), in partnership with Mainstay and the College Advising Corps (CAC), acted quickly to provide students with proactive outreach and guidance about college-going tasks via an innovative large-scale texting campaign. This outreach specifically targeted US high school students who were the first in their families to go to college and had low family income. For up to 38 weeks, a Mainstay AI chatbot named Oli sent scripted messages to students on various topics related to the college search, application, and matriculation processes. To better target the information, Oli solicited information directly from students about the types of resources they needed and

pressing questions they had. Student questions that Oli could not answer were forwarded to CAC advisers, who would follow up directly with individual students.

Using a propensity score matching approach, I investigate whether college application rates and enrollment rates were higher among students targeted for this chatbot outreach compared to their observationally similar peers who were not. In addition, I examine treatment heterogeneity based on students' application behavior, racial/ethnic identity, and the level of chatbot engagement.

2.0 Paper 1: Evaluating Modern Propensity Score Estimation Methods with High-Dimensional Data

2.1 Introduction

Traditionally, questions of causality have been addressed through RCTs, in which individuals are randomly assigned to either a treatment group that receives an intervention or a control group (Murnane & Willett, 2010; Rosenbaum, 2010). Randomization ensures that, on average, both groups are balanced on all observable and unobservable characteristics, eliminating the possibility that an intervention's effect is due to differences in group characteristics (Rosenbaum, 2010). However, ethical, financial, or practical barriers can prevent a researcher from conducting a randomized experiment in education and broader social science research. Instead, researchers must rely on non-experimental or observational studies in which students self-select or are placed in an intervention without randomization.

Observational studies complicate causal inference because, in the absence of randomization, an imbalance in the distribution of student characteristics (i.e., covariates) between treated and untreated students could lead us to incorrectly attribute a change in the outcome to the intervention or policy, as opposed to differences in group composition (Rosenbaum, 2010). A vast body of methodological literature has centered on developing quasi-experimental methods that attempt to address the internal validity issues associated with observational data by balancing the *observed* characteristics of those who received treatment and those who did not. One popular approach is propensity score analysis (Lee et al., 2010; Lee, 2023; Lee & Little, 2017; Stuart, 2010).

Propensity score analysis is a widely used quasi-experimental method in education research (Fan & Nowell, 2011; Stuart, 2010; Stuart, 2023); it aims to balance observed characteristics between treated and untreated students, similar to how randomization balances observable characteristics between treatment and control groups. The propensity score represents the probability that a student would have received treatment based on their *observed* covariates (Rosenbaum, 1987; Rosenbaum & Rubin, 1981, 1984). There

are several ways to use propensity scores to balance student covariates, including *matching*, in which treated and untreated students with similar propensity scores are paired to form matched sets; *weighting*, in which treatment groups are re-weighted so that observed covariates are balanced between groups, and *stratification*, in which the propensity score is used to group students into unique strata with similar scores, to balance observed covariates within each stratum (Pan & Bai, 2018).

However, certain strict assumptions must be met to estimate unbiased treatment effects correctly using propensity score analysis. For example, the propensity score model must:

(1) Capture all the covariates related to selection into treatment (Rosenbaum, 2010).

(2) Correctly model the association between the student's characteristics and treatment selection, meaning that all proper interactions and non-linear terms should be specified (Rosenbaum, 2010).

The literature suggests a "kitchen sink" approach for variable selection to have the best chance of meeting the first condition since the penalties are high if a potential confounder is not included in the propensity score model (Karim et al., 2018; Webster-Clark et al., 2021). To account for unmeasured confounders, sensitivity analysis has been proposed to understand how excluding such confounders could introduce bias into the treatment effect estimation (L. Li et al., 2011). The literature suggests iterating through various model specifications to meet the second condition. Each iteration adds interactions or non-linear terms until a balance is achieved among the observed characteristics between treated and untreated students (Stuart, 2010). If these assumptions are not met, it will lead to biased treatment effect estimates (Rosenbaum, 2010).

At first glance, high-dimensional data (i.e., big data) should be an asset to propensity score analysis, as conventional practices stress the importance of including all available covariates in the propensity score model to increase the chances of capturing all relevant confounders. Nevertheless, this is not the case. The performance of the traditional logistic model for estimating propensity scores tends to decrease as more variables are included in the estimation step (Hill et al., 2011).

When modeling high-dimensional data, logistic regression may overfit the data or experience issues with perfect separation, resulting in fitted probabilities that are either 0 or 1 (Hill et al., 2011). This rigidity can lead to propensity scores with little to no variability. Additionally, as the number of covariates

available to estimate the propensity score increases, it becomes increasingly challenging to iteratively model all appropriate interactions and non-linearities using a logistic model (Dorie et al., 2019; Hill et al., 2011).

With the rapid advancement of sophisticated machine learning algorithms, there are growing opportunities to develop and apply new methods for estimating propensity scores, particularly with high-dimensional data (Hernández-Blanco et al., 2019). These machine learning algorithms offer an automatic, data-driven way to capture non-linearities and non-additivity in the propensity score estimation model, potentially leading to a more balanced covariate distribution without manual iteration. Previous propensity score simulation research has focused on low-dimensional datasets (Cannas & Arpino, 2019; Lee et al., 2010; McCaffrey et al., 2004; Setoguchi et al., 2008; Stuart, 2010). To date, limited research has evaluated the performance of these machine-learning methods when applied to high-dimensional data, particularly considering complexities in both the propensity and outcome model.

Additionally, recent advances in computing and software have generated a novel machine-learning algorithm architectures that could be used to estimate propensity scores, DNNs (Hernández-Blanco et al., 2019; LeCun et al., 2015; Pang et al., 2019). DNNs are artificial intelligence algorithms modeled after the architecture of the human brain (LeCun et al., 2015; Pang et al., 2019); they are widely used across various industries for complex prediction and classification tasks and effectively model complex high-dimensional data. The flexibility of DNNs enables them to capture complex interactions and non-linearities and perform automatic variable selection (Hernández-Blanco et al., 2019; LeCun et al., 2015; Pang et al., 2019). However, there has been limited application of DNNs to causal inference, particularly in social science research.

In this study, I compare the performance of the traditional logistic regression to various machine learning methods with high-dimensional data using propensity score weighting. In addition, I overcome the limitations of existing methods by proposing a novel DNN-based approach to the estimation of propensity scores and illustrating a novel framework for generating multi-type, correlated high-dimensional data replicating the high-dimensional administrative data that is becoming available to education researchers.

The remainder of this document is structured as follows: First, I will review the relevant literature for this study. Then, I will describe my simulation approach and present my results. Finally, I will conclude

with a discussion section that focuses on the contributions of my study.

2.2 Literature Review

In this section, I will review the literature on causal inference and propensity score analysis, focusing specifically on machine-learning approaches for estimating propensity scores. Additionally, I will draw from the literature on neural network approaches, including deep neural networks, to motivate the development of my DNN-based method for estimating propensity scores.

2.2.1 Neyman–Holland–Rubin Model for Causal Inference

The Neyman–Holland–Rubin model, also known as the Rubin causal model, forms the basis of my approach to causal inference (Holland, 1986; Neyman, 1923; Rubin, 1976, 2005). A core tenet of the Rubin causal model states that a treatment effect is the difference between two potential outcomes for an individual (Rubin, 2005). Consider a dichotomous treatment variable (Z), where $Z_i = 1$ represents the ith student being in the treatment group—such as a college access program—and $Z_i = 0$ represents the student not being in the treatment group. Let Y_{iz} be a potential outcome for student i depending on treatment assignment Z, such that an individual student has two potential outcomes. Y_{i1} is the potential outcome had student i participated in the college access program, and Y_{i0} is the potential outcome had that same student not participated. Note, the potential outcomes framework is based on the stable unit value assumption (SUTVA), which states there is only one version of the treatment condition and that the potential outcomes of an individual are not influenced by the treatment assignment of any other individual (Rosenbaum, 2010). Therefore, the treatment effect of the program for student i would be the difference in their two potential outcomes:

$$\tau_i = Y_{i1} - Y_{i0} \tag{2.1}$$

However, in the real world, we do not simultaneously observe both potential outcomes for the same student. We can only observe student i's potential outcomes in the college access program if they participated; this is why the fundamental problem of causal inference is a missing data problem (Holland, 1986). Although we cannot directly observe these potential outcomes, under certain assumptions, we can estimate the average

treatment effect (ATE) as the population average of all individual treatment effects, defined as:

$$ATE = \mathbb{E}(Y_{i1} - Y_{i0}) = \mathbb{E}(Y_{i1}) - \mathbb{E}(Y_{i0}) \tag{2.2}$$

where $\mathbb{E}(Y_{i1})$ is the expected value of all students in the treatment group and $\mathbb{E}(Y_{i0})$ is the expected value of all students in the control group. In the context of a true experiment in which students are *randomly* assigned to treatment conditions, we can straightforwardly estimate the ATE since, with a large enough sample size, both treated and control students will be balanced on all observable and unobservable characteristics. A randomized experiment ensures that Y_1 and Y_0 are independent of treatment assignment (Z), and therefore, the treatment effect can be regarded as causal (Rosenbaum, 2010).

In addition, we may also be interested in estimating the ATE among those who received the treatment. In this case, we may be interested in the average treatment effect on the treated (ATT) (Rosenbaum, 2010), which is defined as:

$$ATT = \mathbb{E}(Y_{i1} - Y_{i0}|Z_i = 1) = \mathbb{E}(Y_{i1}|Z_i = 1) - \mathbb{E}(Y_{i0}|Z_i = 1) \tag{2.3}$$

2.2.2 Observational Studies

In behavioral and social science research, conducting a randomized experiment may not always be possible due to ethical, financial, or practical limitations (Bai, 2011; Pan & Bai, 2018). Such limitations lead researchers to rely on non-experimental or observational data, where students either self-select or are assigned to an intervention without randomization. This reliance creates a challenge for causal inference; an imbalance in the distribution of student characteristics (i.e., covariates) between the treated and untreated students could lead to incorrect attributions of changes in the outcome to the intervention rather than differences in group composition (Austin, 2011).

For example, consider our earlier example of the college access program. The underlying treatment assignment would be unknown if students were not randomized into treatment or control groups but instead

self-selected into the program. It could be that students who self-select into the program are generally more interested in going to college than those who did not sign up. Given this imbalance in college interest, we may incorrectly attribute higher rates of college going to the treatment instead of differences in college interest between treated and untreated students. In other words, without randomization, observational studies cannot ensure that a student's outcome is independent of treatment assignment (Z) (Rosenbaum, 2010).

In order to overcome the challenges posed by observational data, various statistical methods have been developed to estimate unbiased treatment effects. These methods include instrumental variable approaches, regression discontinuity, synthetic control, and propensity score analysis (Rosenbaum, 2010). For this dissertation, I focus on propensity score analysis, a quasi-experimental method that attempts to balance observed group differences using a balancing score derived from student-level covariates.

2.2.3 Propensity Scores

Rosenbaum and Rubin first introduced the concept of propensity scores to enhance the accuracy of capturing unbiased treatment effects in observational studies (Rosenbaum, 1987; Rosenbaum & Rubin, 1981, 1984). Propensity score analysis has become one of social and behavioral researchers' most widely used quasi-experimental methods, particularly in applied educational research (Lee et al., 2010; Stuart, 2010).

Rosenbaum and Rubin define a propensity score, $e(x)$, as the probability of assignment to treatment conditional on observed pre-treatment covariates, defined as:

$$e(x) = Pr(Z = 1|X) \tag{2.4}$$

Propensity scores are typically estimated as predicted probabilities from a logistic regression model. The binary outcome variable indicates whether a student received treatment and is being predicted based on individual student characteristics (X). The estimated propensity scores range from 0 to 1, with values closer to 1 indicating that an individual is more likely to be assigned to treatment. According to Rosenbaum and Rubin, if the propensity scores are balanced between the treatment and control groups, the distribution of observed covariates between the two groups will also be balanced (Rosenbaum, 1987; Rosenbaum & Rubin, 1981, 1984). This prediction means that, in theory, two students with similar propensity scores should have a similar distribution of observed covariates.

Researchers can adjust for the imbalance in the joint covariate distribution between the treatment and control groups by conditioning on the propensity score using various techniques, including matching, stratification, and weighting. Matching involves pairing treated students with control students based on the proximity of their propensity scores, while stratification involves grouping students into unique strata based on their propensity scores (Bai, 2011). Weighting involves reweighing students in the sample so that the distribution of propensity scores in the control group is similar to that of the treated group (Pan & Bai, 2018). These methods aim to approximate true randomization conditions by using the propensity score to eliminate bias in the treatment effect due to observed confounding and approximate the true causal effect

(Pan & Bai, 2018). However, for this to be achieved, certain assumptions must hold.

2.2.3.1 Assumptions for Identification and Estimation of Causal Effects

Several assumptions must be met to identify and estimate an unbiased treatment effect using propensity scores. These include SUTVA, the strong ignorability assumption, and the region of common support (Rubin, 2005). It is important to note that, while necessary, these assumptions are insufficient for unbiased causal inference using propensity scores. Furthermore, it can be challenging to assess the validity of these assumptions in practice, which can impact the validity of the estimated treatment effects (Bai, 2011).

The first assumption, SUTVA, states that there is only one version of the treatment condition and that the potential outcome of an individual is not influenced by the treatment assignment of any other individual (Pan & Bai, 2018; Rosenbaum, 2010). Thus, the outcome of a student is independent of whether another student is assigned to the treatment or control group.

The strong ignorability assumption states that the potential outcome of an individual is conditionally independent of treatment assignment if we successfully conditioned on all covariates related to treatment, $Y_1, Y_0 \perp Z|X$. Randomization satisfies, in theory, this assumption in experimental studies, which ensures that treatment assignment is independent of an individual's outcome (Rosenbaum, 2010). However, this assumption holds in observational studies only if *all* covariates related to both treatment assignment and outcome are included in the propensity score model (Rosenbaum, 1987; Rosenbaum & Rubin, 1981, 1984). In practice, it is unlikely that all such covariates are included, and sensitivity analysis is necessary to assess the influence of unmeasured confounding (L. Li et al., 2011).

Finally, the region of common support (i.e., positivity) assumption states that there must be sufficient overlap in the estimated propensity score distribution between the treatment and control groups (Rosenbaum, 2010). This overlap is crucial to ensure that suitable comparisons can be found for treated individuals. The adequacy of common support can be assessed by plotting the distribution of the propensity scores in both the treatment and control groups and evaluating if there is sufficient overlap between the two distributions.

Estimating causal effects from observational data is possible if these assumptions are satisfied. How-

ever, some of these assumptions are untestable (Rubin, 2004; Rubin, 2005). For instance, in observational data, the underlying mechanism of treatment assignment is usually unknown, making it difficult to determine if all covariates related to treatment assignment have been included in the propensity score model. As a result, propensity score analysis should be viewed as a method to reduce, but not eliminate, the bias in estimated treatment effects in observational studies. It is crucial to carefully consider which covariates are included in the propensity score model and to conduct sensitivity analysis to assess the impact of unmeasured confounding.

2.2.3.2 Covariate Selection

The selection of covariates in propensity score analysis should be based on variables that are theoretically grounded and related to both treatment assignment and the outcome, also known as confounders (Brookhart et al., 2006; Buhlmann & Geer, 2011). However, there is a debate in the literature about which covariates should be included (Austin, 2011; Karim et al., 2018). Some simulation studies have found that including confounders or covariates that only affect the outcome leads to more precise estimates of treatment effects (Austin et al., 2007). While, including only covariates that affect treatment assignment leads to increased variance in the estimated treatment effect with no reduction in bias (Brookhart et al., 2006).

In practice, determining which covariates are confounders or related to the treatment or the outcome can be challenging. To reduce the risk of excluding potential confounders, the literature suggests using a "kitchen sink" approach to variable selection, where all available covariates are included in the estimation of the propensity scores (Karim et al., 2018; Webster-Clark et al., 2021). Additionally, it is crucial to ensure that the propensity score model captures the correct functional form of the covariates, including all relevant interactions and non-linear terms (Pan & Bai, 2018; Rosenbaum, 2010). If the propensity score model is misspecified, it will result in biased treatment effect estimates (Guo et al., 2014, 2020).

To balance the observed characteristics between treated and untreated students, iterating through various model specifications that include interactions and non-linear terms among the covariates is recommended (Rosenbaum, 2010). However, this iterative process can become challenging and time-consuming as the number of available covariates increases, especially when conducting propensity score analysis with

high-dimensional data.

2.2.4 Estimating Propensity Scores

Estimating the propensity score ($e(x)$) is a crucial step in propensity score analysis, as a misspecified propensity score model can result in propensity scores with little variability or extreme scores that lead to biased treatment effect estimates (Hill et al., 2011). Propensity scores can be estimated using parametric or non-parametric models as long as the model outputs a probability that is bounded between 0 and 1. Logistic regression is a widely used parametric model for estimating propensity scores, particularly in applied educational research (Stuart, 2023).

2.2.4.1 Logistic Regression

Logistic regression is the most common parametric model used to estimate propensity scores in the social science literature due to its ease of interpretation and familiarity with many applied researchers and its ability to accommodate continuous and categorical covariates (Keller et al., 2015). The logistic model is defined as follows:

$$logit(Z = 1|X) = log\left(\frac{Pr(Z = 1|X)}{1 - Pr(Z = 1|X)}\right) = \beta_o + \beta_1 X_1 + \cdots + \beta_p X_p \tag{2.5}$$

where Z is a binary indicator equal to 1 if a student is in the treatment group and 0 otherwise, regressed on X a vector of observed covariates. The logistic regression outputs predicted probabilities that are continuous and bounded between 0 and 1 (i.e., propensity scores).

Although logistic regression is the most commonly used method for estimating propensity scores, it may not always be the best option for inference (Lee et al., 2010; Lee, 2023). Several simulation studies have found that using logistic regression for propensity score modeling can result in biased treatment effect estimates when the model is misspecified, especially in complex data where non-linear and non-additive terms are not included in the propensity score model (Cannas & Arpino, 2019; Lee et al., 2010; McCaffrey et al., 2004; Setoguchi et al., 2008). This situational bias occurs because parametric models, such as logistic regression, require assumptions about covariates' functional form and distribution (Hill et al., 2011). When estimating propensity scores with many covariates, this problem can be particularly pronounced (Hill et al.,

2011). In high-dimensional settings, the logistic model may produce propensity scores outside the desired range [0,1] (Hill et al., 2011). However, non-parametric machine learning algorithms have been proposed as alternatives for estimating the propensity score (Cannas & Arpino, 2019; Lee et al., 2010; McCaffrey et al., 2004; Setoguchi et al., 2008).

2.2.4.2 Machine Learning Approaches

Machine learning algorithms have recently gained popularity in the causal inference and propensity score literature due to these algorithms being highly flexible and able to model complex functional forms iteratively without explicit manipulation from the researcher (Athey, 2015; Cui et al., 2020; Grimmer, 2015). By making a simple modification, such as applying a logistic activation function to the output layer of a neural network, these algorithms can output bounded probabilities that can be utilized as propensity scores. Given the success of machine learning algorithms for prediction tasks, it is reasonable to think they are worthy candidates for the propensity score estimation problem, which is a prediction problem. Broadly, machine learning algorithms used in propensity score estimation can be classified into tree-based, ensemble methods, and—the focus of my dissertation—neural network-based approaches.

2.2.4.3 Tree-Based Methods

One popular machine learning approach to estimating propensity score is classification and regression trees (CART) (Denison et al., 1998). This algorithm recursively divides data into subsets based on individual covariates to predict the probability of membership assignment of a given individual. To generate propensity scores, the outcome is set to a binary indicator of treatment assignment (Z). This algorithm splits covariates by level of importance, with the first split being the covariate that can produce the most distinctive split. For example, if a student's age is related to the treatment assignment, the first split would divide the data into students with $age > 15$ and $age < 15$. The algorithm continues by splitting the data by the next most influential covariate, creating a tree-like structure (see Figure 2.1). Splitting stops when the data are binned into unique "branches," that minimize the prediction error such that an additional split would not improve the prediction of treatment assignment. The final "branches" of the model represent the individual groups

of students with similar propensity scores (Denison et al., 1998). The output of CART is the predicted probabilities, which can be used as propensity scores. Like logistic regression, the CART algorithm can handle continuous and categorical covariates. However, unlike logistic regression, CART is insensitive to outliers and can automatically model interactions and higher-order terms (Westreich et al., 2010; Wyss et al., 2014). Through a simulation study, Lee et al. (2010) found that CART outperformed logistic regression in reducing bias and balancing covariates when the data-generating mechanism had complex associations.

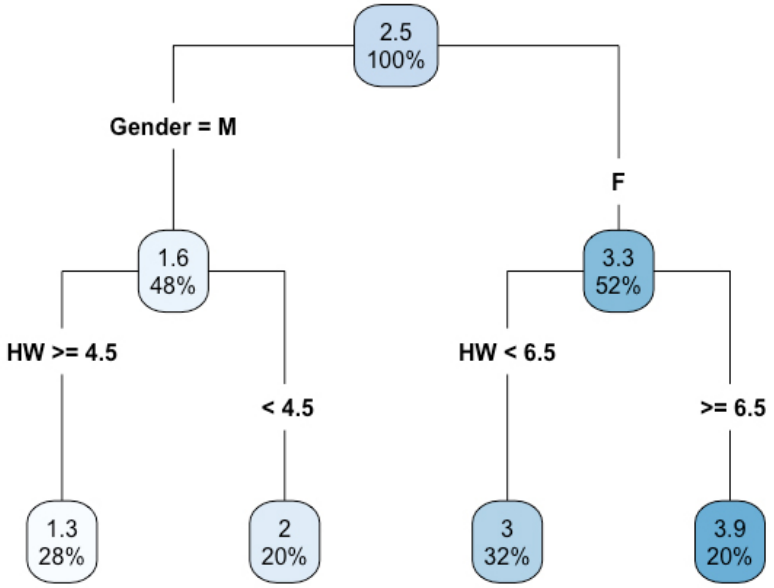


Figure 2.1: Simple Decision Tree Predicting Students GPA Based on Gender and Homework Grade

However, a well-known issue with CART is its tendency to overfit the data (Lee et al., 2010). A modified version of CART was developed to address this issue; it prunes back branches that do not contribute to reducing prediction error, referred to as pruned-CART (Westreich et al., 2010). Although pruning can alleviate some of the changes in overfitting, a significant drawback of CART is its reliance on a single tree, which may be weak in predicting the propensity score.

2.2.4.4 Ensemble Methods

Ensemble methods are a family of algorithms that generate multiple trees to predict treatment assignment (Lee et al., 2010; Stuart, 2010; Stuart, 2023). By combining the predictions from multiple trees,

ensemble methods can improve the accuracy of predictions since many weak trees together can create a better prediction than any single tree (Lee et al., 2010). A popular ensemble method is bootstrapped aggregated CART, referred to as bagged-CART, which fits multiple CARTs on bootstrapped data samples (Lee et al., 2010). Growing trees on bootstrapped samples reduce the chances of overfitting the data. Each tree will generate a probability that an individual will be assigned to treatment. The final probability of membership is based on aggregating the probabilities of assignment across all trees (i.e., forest). A more robust ensemble method is random forest (Suk et al., 2021). Compared to CART and Bagging, the random forest algorithm grows trees by randomly selecting covariates and individuals to grow individual trees. With each new tree, the algorithm "learns" the best combination of covariates to generate a final probabilistic prediction based on the "forest" it has created. These random forest techniques have shown significant promise in predicting propensity scores in complex data associations (Cannas & Arpino, 2019; Lee et al., 2010). However, with the recent advances in computing power, non-tree-based methods, such as neural networks, have increased in popularity (Collier et al., 2021; Collier & Leite, 2021; Dreiseitl & Ohno-Machado, 2002; Keller et al., 2015; Westreich et al., 2010).

2.2.5 Artificial Neural Network Architectures

2.2.5.1 Artificial Neuron

The building block for artificial neural networks was initially developed by psychologist Frank Rosenblatt (1958) with their idea of a perceptron or artificial neuron, see Figure 2.2. An artificial neuron is a simplified mathematical model of neurons in our brain (LeCun et al., 2015). Neurons are biological switches that take input signals from other neurons that cause the neuron to fire. Neurons can be thought of as simple processing units. In our brain, these individual neurons are connected into vast networks of billions of neurons, where the outputs of one neuron become inputs of another, allowing for the transmission of complex information (LeCun et al., 2015).

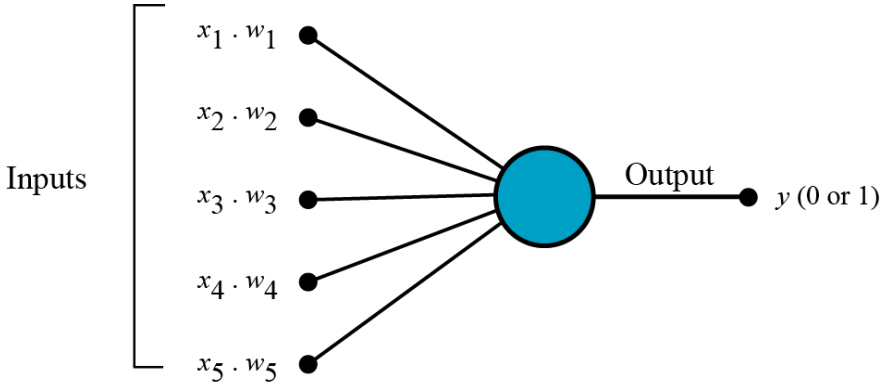


Figure 2.2: Structure of a Perceptron - Artificial Neuron

The artificial neuron receives a set of inputs $X_i = (X_1, X_2, \ldots, X_p)$ with corresponding weights $w_i = (w_1, w_2, \ldots, w_p)$ that represent the strength of each input variable. The inputs are multiplied by their corresponding weights and then summed with a bias term β_0 to create a linear transformation of the inputs. The weight and bias terms are similar to the slope and intercept in linear regression. The linear transformation is then passed through an activation function $g(z)$ to generate an output $(\hat{y})$ that is either a 1 or 0, or a value between 0 and 1 if the activation function is set to be linear, such as the logistic function (also refered to as the sigmoid function).

The artificial neuron in Equation 2.6 can be used to solve a binary classification problem. In our case, we are interested in classifying each student in our sample as belonging to the treatment or control

condition based on observed covariates. If the activation function $g(z)$ is set to a logistic function, the desired propensity score will be guaranteed an output value between 0 and 1. For the neuron to make accurate predictions for each student, it must "learn" the correct weight and bias parameter specifications, similar to finding the correct slope and intercept in bivariate data.

$$\hat{y} = g(z) = g(\beta_0 + \sum_{i=1}^{p} X_i w_i) \tag{2.6}$$

This learning process starts by feeding each observation in the sample into the neuron. The first pass initializes the weights and biases set to random numbers. The outcome variable is set to the binary indicator of treatment assignment, so a predicted probability of treatment assignment is calculated for each student. However, the initial pass of the data is likely to generate incorrect predictions as the weights and bias terms are initialized to random numbers. In other words, the prediction of treatment assignment ($\hat{y}$) will be far from the actual treatment assignment (y), leading to a high associated loss.

It is important to note that this learning process is an optimization problem that can be solved through various optimization algorithms such as gradient descent or stochastic gradient descent (Goodfellow et al., 2016). The choice of optimization algorithm will impact the speed and accuracy of the learning process.

The goal of the artificial neuron and many machine learning algorithms is to correctly learn the values of the weights and bias parameters that minimize a loss function, such as the mean squared error (MSE). The neuron will update the bias and weight parameters iteratively for each subsequent data pass to minimize the loss function. This update is typically done using a process known as backpropagation. After a user-defined number of iterations, the neuron will be optimized with bias and weight terms that best predict treatment assignment. A single neuron may be appropriate for simple classification problems with linear covariate associations. However, as the complexities of covariate associations increase, more neurons may be needed to learn complex functional forms.

2.2.5.2 Single-Layer Neural Network

Following the development of the artificial neuron, researchers began to model more complex processing units composed of interconnected artificial neurons organized in layers. This work culminated in creating the single-layer feed-forward neural network (NN), as shown in Figure Figure 2.3. NNs are referred to as single-layer because they have a single middle layer of interconnected artificial neurons. These NNs can learn very complex data representations by generating various non-linear representations of the data through multiple artificial neurons (Derry et al., 2023). These non-linear outputs are passed into a final layer that pools the outputs of the individual neurons into a non-linear function, $f(x)$, generating a prediction ($\hat{y}$). By having multiple artificial neurons calculating predictions on various non-linear combinations of the input variables, these networks can learn complex non-linear and non-additive associations between covariates.

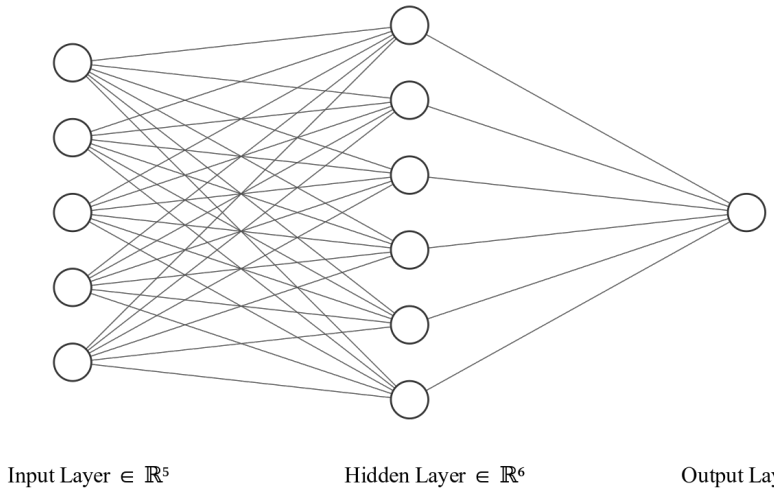


Figure 2.3: Structure of Single-Layer Neural Network

The single-layer NN comprises a series of artificial neurons organized into three layers (Derry et al., 2023). The first layer is the input layer of p variables, $X_i = (X_1, X_2, \dots, X_p)$, our observed covariates. These input variables are fully connected to the second layer—a hidden layer—which comprises K hidden neurons, $A = (A_1, A_2, \dots, A_k)$. The hidden neurons are identical to the artificial neurons mentioned previously, but the activation function is swapped out for a non-linear function. Each hidden neuron calculates its prediction based on its unique non-linear combination of variable inputs and associated weight and bias parameters. The last layer is the output layer, which pools the non-linear outputs of the hidden neurons and passes them as inputs to a non-linear output function, $f(x)$, which generates the final prediction ($\widehat{y}$). The interconnected neurons in NN allow it to learn more complex associations between covariates and treatment assignment than a single artificial neuron (Derry et al., 2023; Hernández-Blanco et al., 2019).

We can express a NN generally as [1]:

$$\widehat{y} = f(x) = B_0 + \sum_{k=1}^{K} B_k h_k(X) = B_0 + \sum_{k=1}^{K} B_k g(w_{k0} + \sum_{i=1}^{p} w_{ki} X_i) \tag{2.7}$$

where each hidden unit $A_k, k = 1, \dots, K$ is created from a weighted linear combination of input variables, $X_1, X_2, \dots, X_p$ that are applied to an activation function $g(z)$ resulting in an "activation" for each hidden neuron, A_k:

$$A_k = h_k(X) = g(w_{k0} + \sum_{i=1}^{p} w_{ki} X_i) \tag{2.8}$$

w_{k0} and w_{ki} are the bias and corresponding weights for each A_k hidden neuron. We can regard these activations as individual predictions of each hidden neuron. These predictions are then pooled and passed to the output function ($f(x)$), which has its bias term, B_0, and produces a final prediction ($\widehat{y}$).

The learning process of a NN works similarly to that of a single neuron, but instead of learning only one set of weights and bias terms, it learns multiple sets simultaneously. The ultimate goal of the NN is to minimize the loss function by learning the correct weights and bias parameters. Unlike a single

[1] This notation was adapted from James et al. (2013).

artificial neuron, NN can efficiently learn complex non-linear relationships between covariates and treatment assignments (James et al., 2013). A handful of studies have evaluated the performance of NN in estimating propensity scores in a low-dimensional setting (Cannas & Arpino, 2019; Collier et al., 2021; Keller et al., 2015; Setoguchi et al., 2008). These studies suggest that NNs represent a suitable method for estimating the propensity score and generally outperform logistic regression in reducing bias and balancing covariates.

The development of NNs has led to the creation of more complex NN architectures, such as DNNs or multilayer neural networks (LeCun et al., 2015). These networks consist of multiple hidden layers and varying numbers of hidden neurons.

2.2.5.3 Deep Neural Networks

The DNN architecture captures even more complex relationships among covariates than a single-layer NN (Hernández-Blanco et al., 2019; Pang et al., 2019). This effectiveness is due to the multiple hidden layers in the network, making it "deep". The diagram in Figure 2.3 shows a DNN with two hidden layers (L_1, L_2).

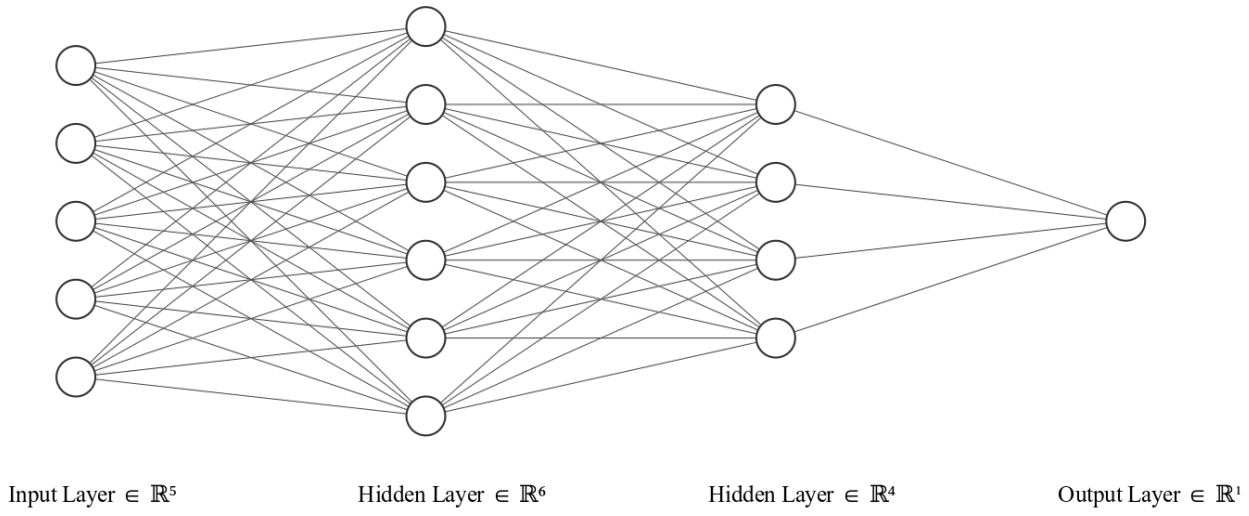


Figure 2.4: Structure of Deep Neural Network with Two Hidden Layers

The first hidden layer functions like the middle layer in an NN (as described in Equation Equation 2.8). The first layer in a DNN can be represented as:

$$A_k^{(1)} = h_k^{(1)}(X) = g(w_{k0}^{(1)} + \sum_{i=1}^{p} w_{ki}^{(1)} X_i) \tag{2.9}$$

Each hidden unit in the first layer, $A_k^{(1)}, k = 1, \ldots, K$ is created from a weighted linear combination of the input variables, $X_1, X_2, \ldots, X_p$. The weighted linear combination is then applied to a non-linear activation function $g(z)$, resulting in output for each hidden neuron in the first layer, $A_k^{(1)}$.

The second hidden layer L_2 takes as inputs the outputs of $A_k^{(1)}$ of the first hidden layer and computes a new non-linear transformation, $A_k^{(2)}$. The new activations of the second layer are calculated as follows:

$$A_l^{(2)} = h_k^{(2)}(X) = g(w_{l0}^{(2)} + \sum_{k=1}^{K_1} w_{lk}^{(2)} A_k^{(1)}) \tag{2.10}$$

where each hidden unit in the second layer $A_l^{(2)}, l = 1, \ldots, K$ is a function of the output of the weighted non-linear combinations of $A_k^{(1)}$, this process can be extended to additional hidden layers. As data move from layer to layer, DNNs can approximate increasingly complex functional forms (James et al., 2013). The complexity should inform the choice of the number of hidden layers in a DNN of the problem and the computational resources available (James et al., 2013). While more hidden layers can allow the network to capture increasingly complex functional forms, they also have a higher computational cost. In general, three hidden layers are sufficient for capturing almost any functional form in problems outside of image and speech recognition (James et al., 2013).

2.2.5.4 Deep Neural Networks in Causal Inference

The popularity of DNNs in causal inference has increased in recent years due to their ability to uncover complex relationships in high-dimensional data with improved precision compared to traditional statistical and machine learning methods (Hernández-Blanco et al., 2019; Pang et al., 2019). This capability to capture intricate relationships in large-volume data makes DNN algorithms suitable for applications in propensity score analysis and other causal inference methods. DNNs have garnered substantial attention in industry and academic research, making substantial advances in image recognition and natural language processing

(James et al., 2013). For example, ChatGPT a DNN-based learning model has recently taken the world by storm and has permeated into the educational discourse. The recent investment by Google in software development has made it easier to develop DNN models using high-level application programming interfaces (APIs) such as Keras, which is a popular deep learning library for building and training neural networks (Pang et al., 2019).

To date, DNNs have received limited attention in the broader literature on causal inference, with most research focused on medical or genomic applications (Iglesias et al., 2021; Kale et al., 2015). DNNs have not yet been widely adopted in social science, likely due to a limited understanding of implementing these algorithms (Farrell et al., 2021). As high-dimensional datasets become more prevalent in social science research, it is expected that DNNs will become more widespread and a promising area of investigation in the future. Despite this potential, there has been limited research that incorporates DNNs into propensity score analysis.

Only two medical peer-reviewed studies have applied DNNs for propensity score estimation (Weberpals et al., 2021; Whata & Chimedza, 2022). Weberpals et al. (2021) used data from around 130,000 cancer patients to construct a simulated dataset with 31 baseline covariates, where the treatment was a fictional cancer drug, and the outcome was a binary survival indicator. Employing a specific DNN architecture with three hidden layers, known as an "autoencoder", they found that DNN reduced confounding bias in treatment effect estimates compared to the traditional logistic model with manual variable selection. Conversely, Whata & Chimedza (2022) compared the performance of logistic regression to a DNN with four hidden layers through a simulation (with 15 covariates) and a real-world application. In their simulation, they varied the level of complexity in the population selection model but not the outcome model. The results showed that DNN outperformed logistic regression regarding covariate balance, classification accuracy, and absolute relative bias.

Both studies demonstrate the growing promise in using DNNs for causal inference and suggest that DNNs may be a valid approach for estimating the propensity score. My dissertation builds upon these studies by comparing DNNs to traditional logistic regression and established machine learning methods

in the propensity score simulation literature. Additionally, my study will test the complexity of both the outcome and selection models, making it the first to examine the use of DNNs for propensity score estimation outside of medical research and with high-dimensional data that resembles high-dimensional administrative datasets encountered in educational research.

2.3 Method

My first dissertation study focuses on a statistical simulation that evaluates the performance of several modern machine learning techniques, including CART, bagged-CART, random forest, and a single-layer neural network. These techniques will be compared to the traditional logistic regression and my proposed DNN-based approach in estimating the propensity score to recover the ATE using propensity score weighting. I will specifically examine the impact of different data conditions (i.e., low vs. high-dimensional context) and the inclusion of complexities in the population treatment and outcome models on estimating the ATE and covariate balance. The choice of these specific conditions is motivated by the need to understand how the performance of the machine learning techniques varies across diverse data structures, which is crucial for determining their applicability and robustness in real-world scenarios. Additionally, assessing the impact of complexities in the treatment and outcome models will provide insights into the strengths and limitations of each technique, guiding researchers in selecting the most appropriate method for their studies.

My study will use a Monte Carlo simulation design informed by previous propensity score estimation studies conducted by Setoguchi et al. (2008), Lee et al. (2010), Cannas & Arpino (2019), Keller et al. (2015), Collier et al. (2021). My simulation code was built upon Cannas & Arpino (2019) publicly available code and will follow best practices in statistical simulation as outlined by (Morris et al., 2018). In the following sections, I will detail the simulation design for this study, including the data generation process, estimands of interest, specific method specifications, and performance measures.

2.3.1 Data-Generation Mechanisms

The data generation mechanism for my simulation was informed by the evaluation I conducted as part of my second dissertation study, which involved a large-scale evaluation of an AI chatbot intervention using high-dimensional administrative data. I based the coefficient values of the covariates and error variances on those data. By basing my simulation on real-world data, I aim to address the limitations of statistical simulations that may not generalize well to real datasets (Huber et al., 2013).

2.3.1.1 Covariates

I used a programmatic approach to generate $(X_1, X_2, X_3, ..., X_p)$ correlated covariates. Specifically, I simulated three covariate conditions with varying covariates generated, with $p = 20, 100, 200$. The 20 covariate conditions represented the average number of covariates used in typical propensity score analyses in educational research (Huber et al., 2013; Powell et al., 2020). The remaining two conditions, with $p = 100$ and $p = 200$, were chosen as they represent high-dimensional datasets commonly encountered in education research (Bird et al., 2022; Einav & Levin, 2014; Grimmer, 2015; Hill et al., 2011; Song & Coleman, 2020).

Much of the literature on propensity score simulation generates uncorrelated covariates (Cannas & Arpino, 2019; Lee et al., 2010; Setoguchi et al., 2008). However, this does not represent the correlated variables typically found in administrative data. To address this, all covariates were generated from random draws from a multivariate normal distribution with $\mu = 0$ and $\sigma = 1$, with a specified correlation matrix with correlations ranging from -0.3 to 0.3, using the mvrnorm function from the MASS package in R (Ripley et al., 2013). Additionally, by inducing bounded correlations that straddle 0, I aim to circumvent potential issues that may arise when correlations approach -1 or 1, such as collinearity.

In order to better mimic the actual structure of variables in administrative data sets and to preserve the correlation structure, I used the inverse cumulative distribution function (CDF) approach (B. Li et al., 2021) to convert the initial random pull of covariates from the multivariate normal distribution to three distributions: a normal distribution with a mean of 0 and standard deviation of 1, a uniform distribution with a range of 0 to 1, and a Bernoulli distribution with a probability of success set at 0.5. The purpose of this step is to create more realistic and heterogeneous data that reflects the distribution of variables typically observed in real-world datasets, while maintaining the same correlation structure. Although some correlations may be slightly suppressed, they should be equal on average across the simulation. The covariates were constructed such that half affected both treatment and outcome (i.e., confounders), a quarter affected treatment only, and the remaining quarter affected the outcome only. This approach is in line with other propensity score simulation studies (Cannas & Arpino, 2019; Lee et al., 2010; Setoguchi et al., 2008).

2.3.1.2 Sample Size

I used a fixed sample size of 10,000 for all conditions. This sample size is larger than typical in many propensity score simulation studies (n = 500 to 2,000). The choice of this sample size is motivated by my goal to mimic real-world high-dimensional administrative data. Additionally, it was necessary to have a sufficiently large sample size to obtain accurate estimates, particularly when employing computationally intensive methods.

2.3.1.3 Population Treatment Assignment Models

I generated the population treatment assignment models using standard logistic regression. These models can also be referred to as population propensity score models, as they generate the true probability of being assigned to treatment. The following logistic regression model represents the general structure of these models.

$$e(x) = Pr(Z = 1|X) = (1 + exp(-(\beta_0 + \sum_{i=1}^{p} \beta_i X_i)))^{-1} \tag{2.11}$$

X represents covariates and their associated coefficients (β). The output, $Pr(Z = 1|X)$ is the probability of assignment to the treatment condition (i.e., the propensity score, $e(x)$). The intercept, β_0, was set to 0.25, and the remaining coefficients, β_i, were randomly assigned values between -0.4 and 0.4, which were based on the corresponding standardized coefficient values in the Common App evaluation.

I generated two population treatment assignment models, Base(T) and Complex(T). To add complexity to the complex models, I included interactions (non-additivity) and higher-order terms (non-linearity) for the confounders and covariates related to treatment.

Base(T): includes only main effects, which assumes a linear relationship between covariates and treatment assignment. I consider this the base model, which is expressed using the following formula:

$$e(x)^{Base} = (1 + exp(-(\beta_0 + \sum_{i=1}^{p} \beta_i X_i)))^{-1} \tag{2.12}$$

X represents covariates and their associated coefficients β. The output is the assignment probability to the treatment condition (i.e., propensity score).

Complex(T): my simulation's most complex population treatment assignment model. To construct it, I incorporated quadratic terms for half of the covariates that are both confounders and related to treatment and interactions with another half of the same group of covariates. This design aims to account for complex relationships between covariates and treatment assignment.

$$e(x)^{Complex} = e(x)^{Base} + \sum_{j=1}^{J}[\beta_j X_j^2 + \beta_j X_j X_{j+1}] \tag{2.13}$$

Using the propensity scores generated from these population selection models, I generated a binary treatment assignment variable by comparing an individual's true propensity score to a random draw from a uniform distribution between 0 and 1. If the propensity score of an individual was higher than the random draw of the uniform distribution, the individual was assigned to treatment such that $Z = 1$, otherwise $Z = 0$. This approach should approximate a probability of assignment to the treatment of 0.5, which is the probability of assignment to a randomized experiment with binary treatment.

2.3.1.4 Population Outcome Models

In addition to the population treatment assignment models, I also generated two population outcome models. To vary the level of complexity between the covariates and the outcome, I included interactions and higher-order terms on covariates related to outcome and overall confounders. Adding complexities to the outcome model is relatively novel in the propensity score literature. However, it is more representative of the complex associations found in datasets in educational research.

The continuous outcome, Y, was generated through regression models. The values of α were based on the Common App evaluation, such that the intercept α_0 equaled -0.18, and the remaining α coefficients received a random value between -0.2 and 0.3. X_i represents p covariates and their associated coefficients (α_i). The error term ϵ was set to have a mean of 0 and a variance of 0.17, based on the error term in the Common App evaluation. The binary treatment variable Z was multiplied by γ, representing the population

treatment effect, and set to 0.3. This treatment effect is consistent with the small effects typically observed in education research, where the average effect size equals 0.28 SD (Richardson, 2011).

The general structure of the population outcome models is defined as follows:

$$Y = \alpha_0 + \gamma Z + \sum_{i=1}^{p} \alpha_i X_i + \epsilon \tag{2.14}$$

Base(Y): the base model; includes only main effects, which assumes a linear relationship between covariates and outcomes. I consider this the base model, which is expressed using the following formula:

$$Y^{Base} = \alpha_0 + \gamma Z + \sum_{i=1}^{p} \alpha_i X_i + \epsilon \tag{2.15}$$

Complex(Y): describes the complex model, which includes the linear terms in addition to non-linearities and non-additivities, which were generated in an identical way to how they were generated in the population treatment outcome models:

$$Y^{Complex} = Y^{Base} + \sum_{j=1}^{J} [\alpha_j X_j^2 + \alpha_j X_j X_{j+1}] \tag{2.16}$$

2.3.1.5 Propensity Score Estimation Methods

Using the simulated data, I estimate propensity scores using standard logistic regression and popular machine learning techniques commonly used in the propensity score estimation literature. This estimation includes a CART, bagged-CART, random forest, and a single-layer neural network. The propensity scores are estimated using all p covariates as main effects in all cases. This scenario is unique in that all confounders are included in the propensity score estimation, which is not typically the case in practice. However, given my interest in evaluating the ability of these methods to recover the true unbiased treatment effect, this approach is well-suited. Future research should examine the performance of these methods when a subset of confounders is omitted from the propensity score estimation model.

Here I present the specification of each method used to estimate propensity scores:

- **Logistic regression**: standard logistic regression with only main effects, using the R *glm* command (Team, 2009).
- **Classification and Regression Trees (CART)**: using the R *rpart* package (Therneau et al., 2015) with default parameters and recursive partitioning.
- **Bagged CART**: bootstrapped aggregated trees using the R *ipred* package (Peters et al., 2009). I changed the default parameter to use 100 bootstrap replicates to align with the Lee et al. (2010) simulation.
- **Random Forest**: parallel tree generation on subsamples based on randomly selected covariates using the R *randomForest* package with default parameters (Liaw et al., 2002).
- **Single-Layer Neural Network (NN)**: I used a simple three-layer NN that consisted of an input layer, one hidden layer, and an output layer. The input layer consisted of p covariates. The hidden layer comprised hidden neurons equal to 1/3 of the input covariates, a common practice in NN applications (Boger & Guterman, 1997). I used a ReLU activation function in the hidden layer for computational efficiency. The ReLU function introduces non-linearity into the network and allows it to learn more complex relationships between input covariates and output. In the output layer, I used a sigmoid activation function to obtain probabilities bounded between 0 and 1. The sigmoid function smoothly maps any input value to a probability between 0 and 1, making it suitable for binary classification tasks. The NN was trained and fit using the R Keras package (Chollet et al., 2015), which interfaces with Python.
- **Deep Neural Network (DNN-2)**: I used a four-layer deep neural network with an input layer, **two** hidden layers, and an output layer. The input layer consisted of p covariates, and each hidden layer comprised hidden neurons equal to 1/3 of the input covariates. Similar to the NN, I used the ReLU activation function in the hidden layers. In contrast, the output layer consisted of a single neuron with a logistic activation function, which provides bounded probabilities between 0 and 1. I trained and fit the DNN-2 using the R Keras package (Chollet et al., 2015).

- **Deep Neural Network (DNN-3)**: The last neural architecture I tested was a five-layer deep neural network with an input layer, **three** hidden layers, and an output layer. A more complex DNN structure, such as this one, can be beneficial when dealing with high-dimensional and non-linear data, as it enables the network to learn more complex and intricate patterns, ultimately improving its predictive performance. The output layer was a single neuron with a logistic activation function to produce bounded probabilities between 0 and 1. The ReLU activation function was also used in the hidden layers. The DNN-3 was also trained and fit using the R Keras package (Chollet et al., 2015), which interfaces with Python.

2.3.1.6 Training Methodology for Neural Network Approaches (NN-1, DNN-2, DNN-3)

A major challenge in using NN and machine learning algorithms is the need to specify values for hyperparameters. Hyperparameters are parameter values not learned from the data but are set before training the model and significantly impact the model's performance (James et al., 2013). Finding the optimal values for these hyperparameters, also known as tuning, is typically achieved through cross-validation, which involves testing a range of values for the targeted parameters to determine the optimal parameters for the NN (James et al., 2013). Choosing incorrect hyperparameters can result in widely varying outcomes.

Hyperparameter specification in neural networks is critical to their performance. The hyperparameters include the number of hidden layers, the number of hidden neurons, the learning rate, and weight decay. The learning rate determines the speed at which the weights are updated during training, while weight decay regulates the weights to prevent overfitting. For this simulation, the number of hidden neurons and activation and loss functions was selected based on established conventions in the field.

Various techniques for hyperparameter tuning have been developed to optimize the model's performance, such as k-fold cross-validation and grid search. However, in this simulation, the ADAM optimizer was utilized (Kingma & Ba, 2014). This optimizer eliminates the need for hand-tuning the learning rate; it is a widely used algorithm that combines the benefits of other algorithms, adapting the learning rates of

each parameter based on historical gradient information, making it more efficient. The ADAM optimizer uses multiple training runs of the data to determine the optimal values of the weights and bias terms in the NN, making it a computationally efficient process without requiring researcher intervention.

The neural network approaches I tested (NN-1, DNN-2, DNN-3) were trained using a standard 80/20 split, with 80% of the sample used for training and 20% for validation. The ADAM optimizer was employed to automatically find the optimal learning rate that minimizes the binary cross-entropy loss function, which is a standard in binary classification problems and measures the difference between predicted and actual probabilities. Early stopping was used to stop the training process when the loss function values did not change after five training runs, and L2 regularization with a value of 0.001 was applied to penalize large weights. These precautions minimized the changes that the neural networks overfit the data.

ReLU activation was used in the hidden layers for computational efficiency, and a logistic activation function was used in the output layer to generate probabilities between 0 and 1. The training was conducted over 100 epochs with a batch size of 64. The choice of hyperparameters was consistent across all three methods to ensure a fair comparison of their performance. The use of the ADAM optimizer, early stopping, and L2 regularization all contributed to the efficient training of the neural networks and optimizing their performance.

2.3.2 Estimating Treatment Effects - Propensity Score Weighting

To estimate the causal effect of the treatment on the outcome, I used a propensity score weighting approach, specifically the inverse probability of treatment weighting (IPTW) (Lee et al., 2010). This method involves reweighing observations based on their estimated propensity score to achieve a covariate balance between the treated and untreated units. The weight for each treated observation was equal to $1/e_i(x)$, and for untreated units, the weight was $1/(1 - e_i(x))$. The estimated propensity score, $e_i(x)$, was determined for each unit i. In accordance with the different methods given above, this weighting scheme was used to estimate the ATE.

I used the R *survey* package to calculate the ATE (Lumley, 2020). The outcome variable (Y) was

predicted using only the binary treatment assignment variable, with the ATE weights included. The ATE point estimate, as well as the robust standard error of the ATE, were saved.

2.3.2.1 Estimating Standard Error of the ATE

The proper method to estimate standard errors when using IPTW weights targeting the ATE remains a topic of debate in the literature. While some researchers support using simple methods, such as the standard deviation of the weighted outcome (Rosenbaum & Rubin, 1984), others argue that more advanced methods, such as bootstrapping (Reifeis & Hudgens, 2022), are necessary to account for the bias introduced by the weighting process. Bias in the weighting process can arise from the propensity score estimation itself, as any inaccuracies in the estimated propensity scores can lead to incorrect weights, potentially distorting the estimate of the treatment effect. Bootstrapping, a resampling technique that involves drawing multiple random samples with replacement from the original data, can help alleviate this bias by providing a more accurate estimate of the sampling distribution of the treatment effect, accounting for the variability introduced by the propensity score estimation process.

Studies that have used propensity score weighting to estimate treatment effects find that the choice of standard error estimator can significantly impact the results. For example, Stuart (2010) found that different standard error estimators can lead to vastly different conclusions about the presence of treatment effects. While, Calonico et al. (2015) found that using a bootstrap estimator led to more accurate confidence intervals than using the weighted outcome's standard deviation.

Given these debates, it is vital to choose an appropriate standard error estimator based on the specific goals of the analysis and the trade-off between computational feasibility and statistical accuracy. Robust standard errors are preferred in this context because they account for the potential biases and uncertainties introduced by the propensity score estimation process, as well as any potential model misspecification. By providing more accurate estimates of the variability in the treatment effect, robust standard errors can lead to more reliable hypothesis testing and confidence interval estimation. To account for the uncertainty in the estimated propensity score, I used the "robust" standard errors calculated from the R survey package, as recommended by Robins et al. (2000).

2.3.3 Performance Metrics

To gauge the performance of each propensity score estimation method, I used the following metrics, which are widely used to assess model performance in the propensity score simulation literature (Cannas & Arpino, 2019; Lee et al., 2010; Setoguchi et al., 2008):

- **Bias**: The difference between the estimated treatment effect and the true population treatment effect of 0.3. Bias values show how far off the estimated ATE is from the true population treatment effect and allow us to determine if we overestimate or underestimate the true ATE.
- **Relative Bias (Bias)**: the difference between the estimated treatment effect and the population treatment effect, divided by the true population treatment effect of 0.3. A low relative bias value indicates that the estimated treatment effect is close to the true population treatment effect. In contrast, a high relative bias value indicates that the estimated treatment effect deviates significantly from the true ATE.
- **Estimated Standard Error of the ATE**: is simply the standard error associated with the estimated ATE.
- **Mean Squared Error (MSE)**: the average of the squared differences between the estimated and population treatment effects. MSE is a good representation of bias and variance and a common performance metric in the machine learning literature (Athey, 2015).
- **95% Confidence Interval Coverage**: an indicator of whether the population treatment effect of 0.3 is found within the estimated 95% confidence interval.
- **Average Standardized Absolute Mean Distance (ASAM)**: is a measure of covariate balance. After calculating the ATE, the absolute value of the standardized difference of means between the treated and untreated group was calculated for each covariate and averaged across all covariates. Lower ASAM values indicate better covariate balance.
- **Power**: is the probability of correctly rejecting the null hypothesis (no treatment effect) when the alternative hypothesis (a treatment effect exists) is true. Power is typically calculated for a prede-

termined significance level (e.g., alpha = 0.05) and is a function of the sample size, effect size, and variability in the data. Higher power indicates a greater likelihood of detecting a true treatment effect, if one exists.

- **Weights**: a known issue with PSW is that extreme weights can result in biased treatment effects. Therefore, I examine the distribution of the estimated weights.

2.3.4 Simulation Scenarios

In total, I tested three different covariate conditions (p = 20, 100, 200) with varying levels of complexity in the treatment and outcome models (Treatment population model = Base, Complex; Outcome population model = Base, Complex). I evaluated seven propensity score estimation methods, including logit, CART, BAG, forest, NN-1, DNN-2, and DNN-3. The sample size was fixed at 10,000. By fully crossing all these conditions, I tested 84 scenarios; each simulated 1,000 times.

2.3.5 Software

The simulations were performed on the University of Pittsburgh Center for Research Computing cluster, utilizing high-memory CPU and GPU nodes to handle the complexity and high-dimensional data generated. Both R (version 4.1.0) and Python (version 3.7) were used, along with popular deep learning libraries Keras and TensorFlow.

2.4 Results

In this section, I present the results of my Monte Carlo simulation study, which compares the performance of various methods in estimating the ATE through Propensity Score Weighting. The methods include my DNN-based approach, traditional logistic regression, and other commonly used machine learning-based approaches such as CART, bagged-CART, and random forest. The simulation aimed to evaluate the accuracy of these methods in producing a propensity score used in estimating the ATE (ATE = 0.3) with IPTW, under different levels of covariate conditions (p = 20, 100, 200) and with varying levels of complexity in the treatment and outcome models (Treatment population model = Base, Complex; Outcome population model = Base, Complex).

The results are presented in four main parts. First, I assess covariate balance through the average standardized absolute mean difference (ASAM). Second, I examine the bias in the ATE estimation. Third, I evaluate the variability of the ATE estimation through the standard error (SE), MSE, and 95% confidence interval coverage. Finally, I present the mean IPTW weights and discuss the results regarding statistical power.

2.4.1 Covariate Balance

I begin by considering the performance of a commonly used and directly observable metric for deciding which method to use, covariate balance (Lee et al., 2010). Covariate balance is assessed using the average standardized absolute mean difference (ASAM) between treated and untreated units. Maintaining a covariate balance reduces the risk of bias in the estimated treatment effect. In my simulation, I focused on assessing balance through ASAM and took the average ASAM for all covariates in the propensity score estimation model. However, in practice, each covariate should be assessed separately. A mean ASAM value of 0 indicates a perfect balance in the mean between treated and untreated units across all covariates. I consider two commonly used thresholds for deciding whether adequate covariate balance has been achieved: the more typical 0.2 thresholds (Imbens & Rubin, 2015) and the more stringent threshold of 0.1 (Austin, 2009).

The results of the ASAM evaluation can be found in Figure 2.5, with thicker and thinner dashed lines

indicating an ASAM value of 0.2 and 0.1, respectively. All methods achieved adequate covariate balance in the low-dimensional condition, with ASAM values below 0.2. However, using the more stringent cutoff of 0.1, both CART and random forest failed to reach this standard. As the number of covariates increased, logistic regression produced extreme imbalance (ASAM = 74.9), while the other machine learning approaches maintained good covariate balance with ASAM values below 0.1. NN-1, DNN-1, DNN-2 and bagged-CART achieved the best balance, with an ASAM value of 0.06.

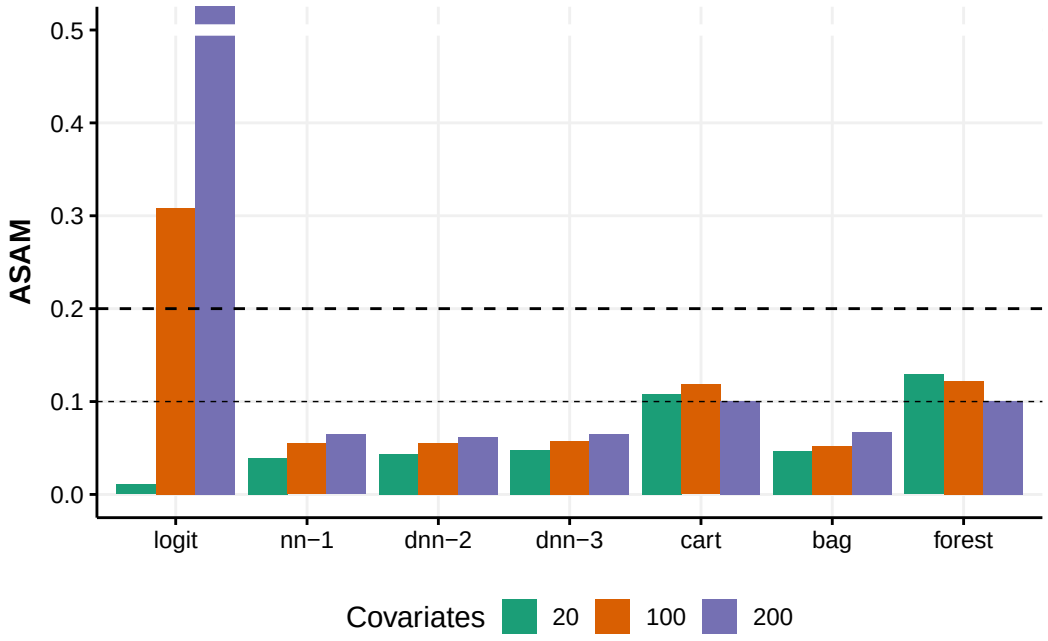


Figure 2.5: Average Standardized Absolute Mean Difference (ASAM) Across the Population Treatment and Outcome Model Conditions

2.4.1.0.0.1 Base - Population Outcome and Selection Models Figure 2.6, when the outcome and selection models only had main effects, all methods achieved good covariate balance using the 0.2 threshold across all covariate conditions. When using the more stringent 0.1 threshold, bagging, the NN-based approaches, and logistic regression produced optimal balance. All methods appear insensitive to the number of covariates in this base condition.

2.4.1.0.0.2 Complex - Population Outcome and Selection Models When complexities are introduced into the treatment and outcome models, there is a slight increase in imbalance, with the exception of logistic regression, which produces an extreme imbalance in the high-covariate condition. In the low-

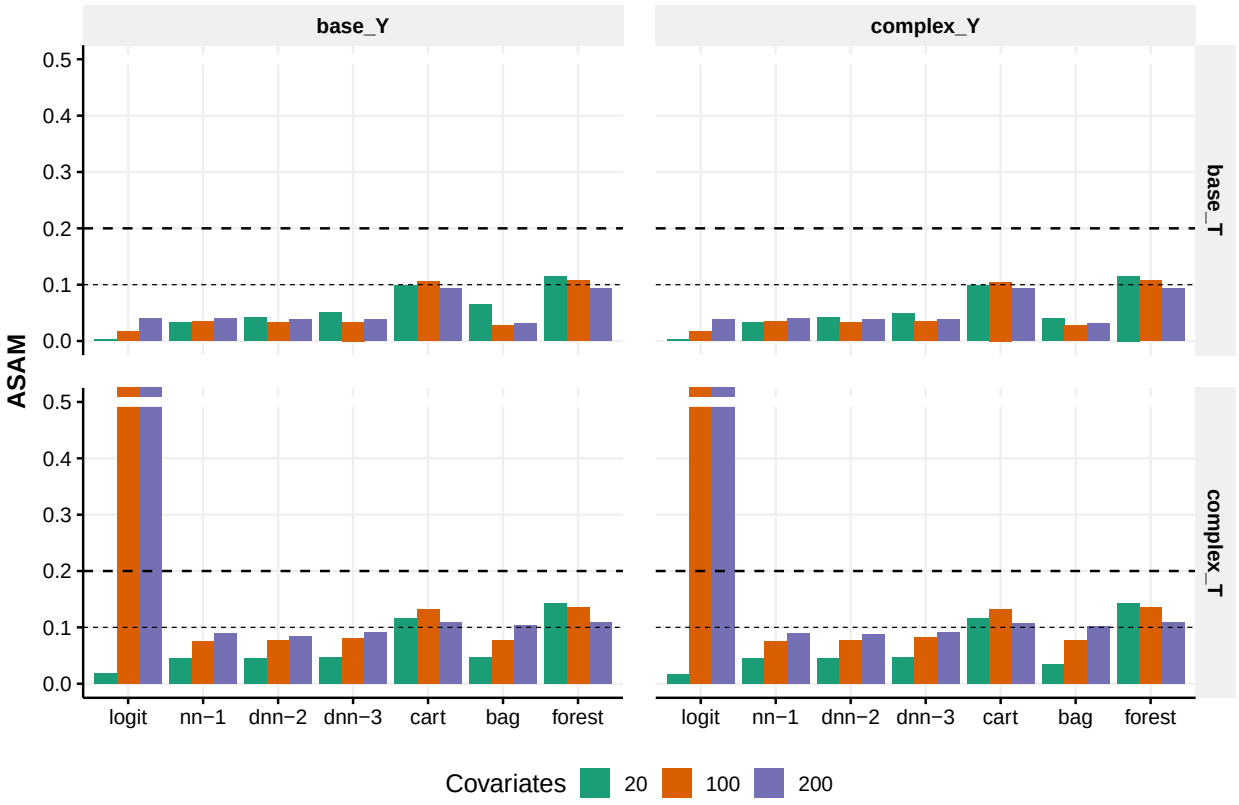


Figure 2.6: Average Standardized Absolute Mean Difference (ASAM) by Data Generating Mechanisms

covariate condition, all methods perform well with ASAM values below 0.2. However, when using the more stringent 0.1 threshold, CART and random forest fail to achieve covariate balance. Logistic regression produced particularly extreme imbalances among covariates in the high-covariate condition, with an ASAM value of 286 for 200 covariates. In contrast, my DNN-2 approach and NN-1 achieved adequate covariate balance with an ASAM value of 0.08. With the most stringent threshold, the NN-based approaches were the only methods that achieved covariate balance.

2.4.1.1 Deviations from the Selection and Outcome Models

When dealing with different specifications of covariate conditions and complexities in the outcome and selection models, covariate balance is relatively robust. However, in the context of higher covariates, specifically 100 and 200, logistic regression is sensitive to complexities in the selection models and shows an increase in imbalance.

2.4.1.1.1 Summary In this simulation, most methods demonstrated good covariate balance with a 0.2 threshold across all covariate conditions, with the exception of logistic regression in high-dimensional cases. However, when using the more stringent 0.1 threshold, the bagging method, NN-1, DNN-2, and DNN-3, performed even better. In high-dimensional contexts, logistic regression produced a significant imbalance, while the neural network-based methods, particularly DNN-2, achieved the best balance with an ASAM value of 0.06, followed by bagged-CART. These results suggest that when considering complexities in both the outcome and selection models, NN-based methods provide the best covariate balance among all methods and are a reliable choice in high-dimensional cases.

2.4.2 Bias of ATE Estimation

2.4.2.1 Bias

In this section I present the evaluation of the bias in the ATE across the different population treatment and outcome model conditions to understand the overall performance of each method. Bias is the difference between the estimated and true ATE (set to 0.3). In other words, it measures how much, on average, a

method tends to over- or under-estimate the true ATE. Figure 2.7 displays my results for the bias of the ATE by method and covariate conditions (p = 20, 100, 200). The solid dots represent the mean bias, and the extending lines are the associated Monte Carlo 95% confidence intervals.

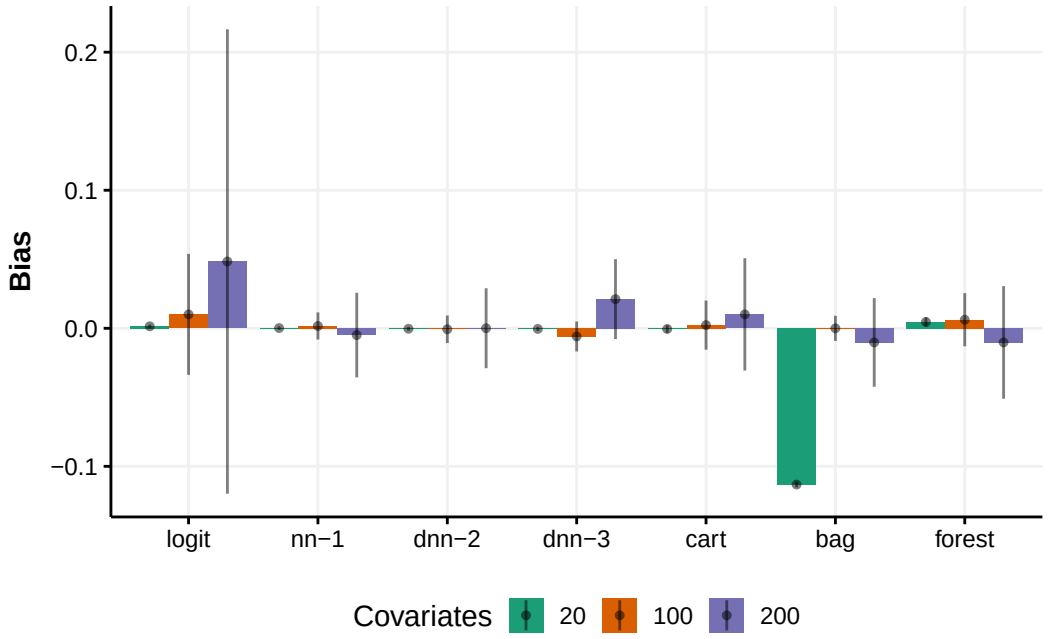


Figure 2.7: Bias of the ATE Across the Population Treatment and Outcome Model Conditions

In the low-dimensional condition (p = 20), all methods accurately estimate the true ATE with relatively little bias and good precision, except for bagged-CART, which substantially underestimates the true ATE. As I increase the number of covariates to 100, the magnitude of bias also increases, with logistic regression overestimating the ATE. At the same time, all other machine learning approaches estimate the ATE with less bias; the results become more pronounced in the highest covariate condition (p = 200) compared to the low-dimensional condition (p = 20). Logistic regression, on average, overestimates the true ATE in addition, its point estimate is severely imprecise, as indicated by a wide Monte Carlo 95% confidence interval. In contrast, the machine learning methods estimate the true ATE with significantly less bias and lower sampling variability. Among these methods, my DNN-2 approach produces the least biased ATE estimate compared to all other machine learning methods and exhibits lower sampling variability.

2.4.2.2 Relative Bias

In Figure 2.8, I present the results for the relative bias for each of the methods and covariate conditions. The relative bias is calculated as the difference between the estimated ATE and the true ATE of 0.3, divided by the true ATE. For example, if the estimated ATE is 0.5 and the true ATE is 0.3, the relative bias would be (0.5-0.3)/0.3 = 0.67 or 67%. This result means that the method has overestimated the true ATE by 67%. Based on the sensitivity analysis literature, I set my tolerance level for bias at less than 10%. Studies generally consider the ATE robust to unmeasured confounding if including an unmeasured confounder changes the ATE by no more than 10% (L. Li et al., 2011).

In Figure 2.8, the solid dots represent the mean relative bias, and the extending lines are the associated 95% Monte Carlo confidence interval. The dashed lines indicate the +- 10% tolerance level.

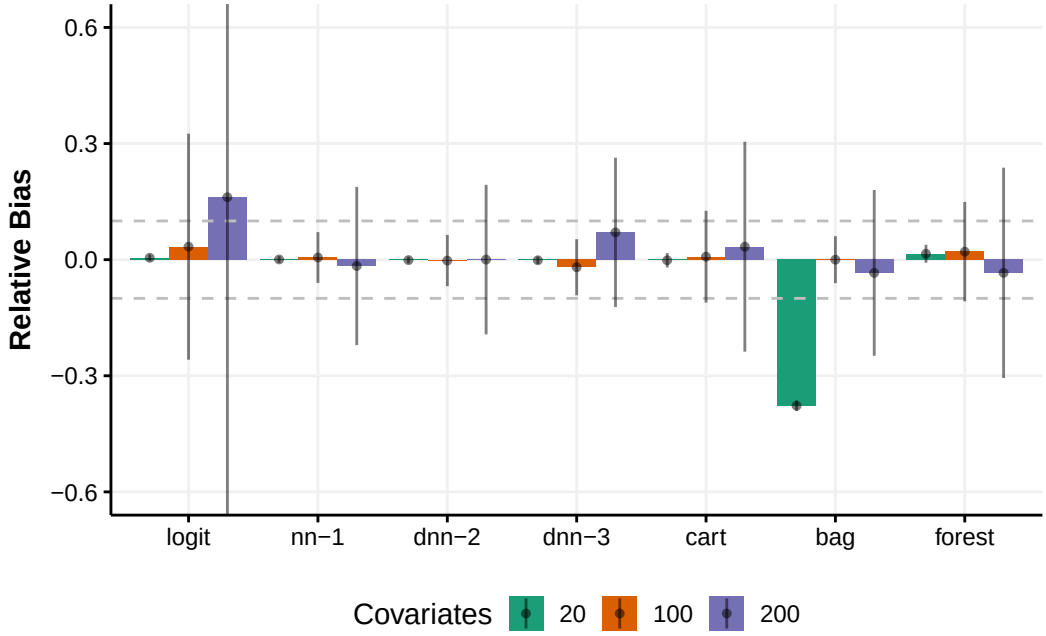


Figure 2.8: Relative Bias of the ATE Across the Population Treatment and Outcome Model Conditions

In the low-dimensional condition (p = 20), nearly all methods accurately estimated the ATE with little bias and high precision. I found that logistic regression had the least relative bias, while all other methods produced less than 2% relative bias. The exception was bagged-CART, which had a large amount of bias (relative bias = -37%), meaning that it severely underestimated the true ATE by around -37% of

the true ATE. As the number of covariates increased from 20 to 100 and 200, logistic regression became substantially more biased compared to all other methods, with substantially more imprecision, reaching its maximum relative bias in the 200-covariate condition, where it overestimated the true ATE by nearly 16%. All machine learning approaches performed well in the 100-covariate condition, producing bias below 4%. In this scenario, my DNN-2 approach and bagged-CART produced the lowest bias estimates, where as all other methods produced estimates exceeding the 10% bias treshold. In the highest-dimensional covariate condition (p = 200), all methods had an increase in bias, with an accompanying increase in sampling variability. However, the machine learning approaches performed exceptionally well compared to logistic regression. My DNN-2 approach produced the least biased results with a mean relative bias of less than 1%, followed by NN-1 and DNN-3. The NN-based approaches produced less bias and exhibited less sampling variability than all other methods, in addition to bagged-CART. However, none of the methods in the 200-covariate condition had 95% Monte Carlo confidence intervals within the 10% threshold. All machine learning approaches had much narrower confidence intervals than logistic regression, which had extreme sampling variability. These results imply that if I used a machine learning method to estimate an ATE in a high-dimensional setting, I would increase my probability of capturing the true population ATE much more with DNN and bagged-CART than if I used logistic regression to estimate the propensity scores.

2.4.2.2.1 Complexities in the Population Outcome and Selection Models Next, I evaluated the performance of each method in terms of relative bias when complexities were introduced into the treatment and outcome model. Recall that the data-generating models *Base T* and *Base Y* only included main effects in the treatment and outcome model, while *Complex T* and *Complex Y* included non-additivity and non-linearities in the form of quadratic and interaction terms among the covariates.

Figure 2.9 presents a faceted plot where the upper-left facet represents the condition where the population treatment and outcome models only include main effects (i.e., Base T & Base Y). The lower-right facet represents the condition where the population treatment and outcome models include quadratic and interaction terms (i.e., Complex T and Complex Y). Deviations from the diagonal facets indicate conditions where only one of the models includes quadratic and interaction terms, while the other only includes main

effects.

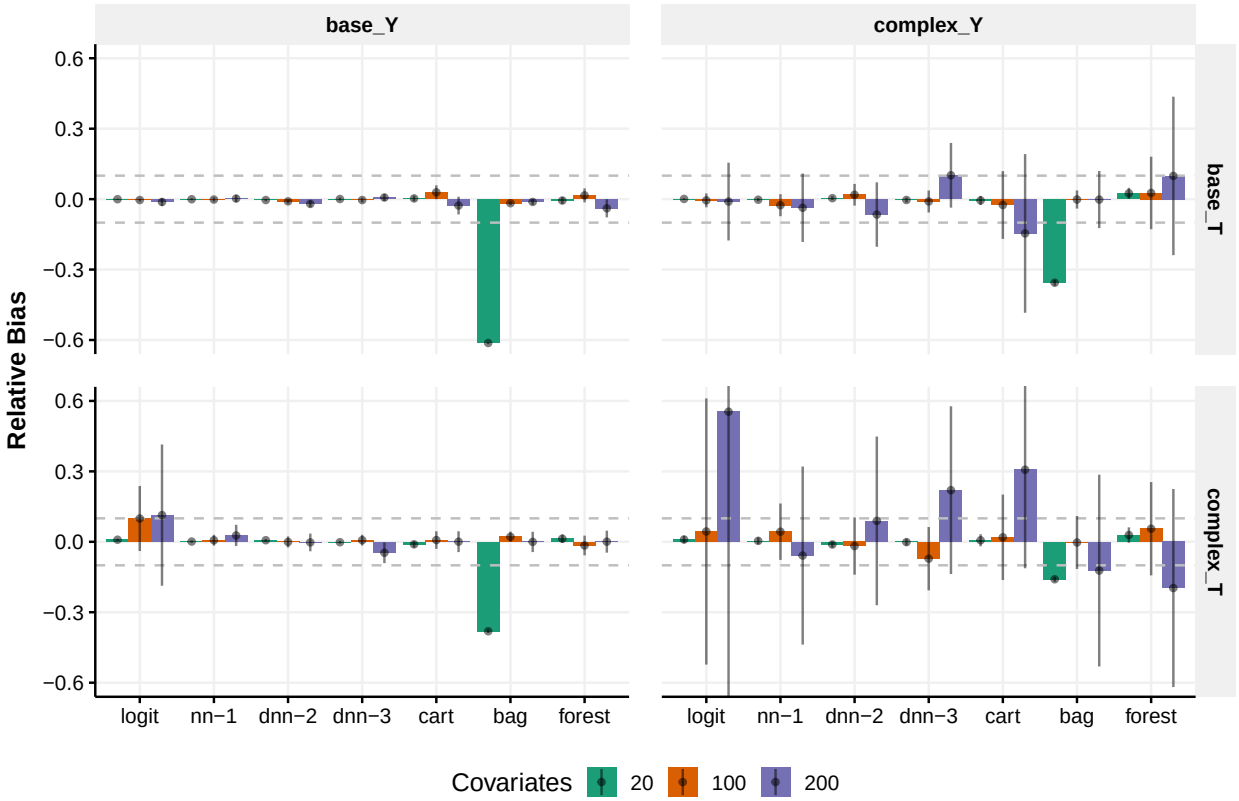


Figure 2.9: Relative Bias of the ATE by Data Generation Conditions

2.4.2.2.1.1 Base - Population Outcome and Selection Models When the treatment and outcome models only include the main effects, nearly all methods perform well, regardless of the number of covariates. The only exception is bagged-CART, which produced the most biased ATE estimate with 20 covariates. Logistic regression produced the least biased estimates (relative bias less than 0.001%) when the population treatment and outcome models included only main effects, which was expected because when estimating the propensity score using the various methods, only main effects are included in the model. This result means that in the Base Y and Base T conditions, the population selection model is equal to the propensity score estimation model. As the number of covariates increased, all methods continue to perform well, with bias within the 10% threshold. This finding suggests that when covariates have simple associations and a large sample size, all methods produce unbiased ATE with a high level of precision, regardless of the number of covariates included, with the exception of the bagged-CART, which should not be used in a

low-covariate condition.

2.4.2.2.1.2 Complex - Population Outcome and Selection Models In contrast to the models where only main effects were included in the population treatment and outcome models, the lower-right facet of Figure 2.9 displays the results when both the population treatment and outcome models include quadratic and interaction terms. When complexities are included in the population models, much more bias is introduced into the ATE estimate with increased sampling variability across methods and covariate conditions. All methods performed well when only 20 covariates were included, with relative bias just within the 10% interval, except for bagged-CART.

However, as the number of covariates increased, there was a significant increase in bias. With 100 covariates, all methods performed well, with mean relative bias well within the 10% threshold. However, regarding sampling variability, only NN-1 bagged-CART and DNN-2 produced bias estimates with Monte Carlo confidence intervals close to but not necessarily within the 10% threshold. In contrast, logistic regression produced very biased estimates, indicated by its large confidence interval. The differences became more pronounced in the high-covariate condition (p = 200); specifically, logistic regression severely overestimated the ATE by more than half of the true ATE (relative bias = 55%). The least biased estimates were produced by the NN-1 and DNN-2 approaches (relative bias = 5% and 9%, respectively), which was far lower than the relative bias produced by all other machine learning approaches. However, even though they produced the least biased results, they still had fairly large sampling variability, although not as large as that produced by logistic regression in the same condition (n = 200).

2.4.2.2.1.3 Deviations - Population Outcome and Selection Models In the lower-left facet of Figure 2.9, I present the results when the selection model is complex, while the outcome model only contains the main effects. In the upper-right facet, I show the results when the outcome model is complex, but the selection model is simple, with only the main effects. Overall, the relative bias performance is sensitive to complexities in either the outcome model or both the treatment and outcome models, especially when the number of covariates is high. This result implies that if either the outcome model or both models are complex, the accuracy of the ATE estimate may be affected.

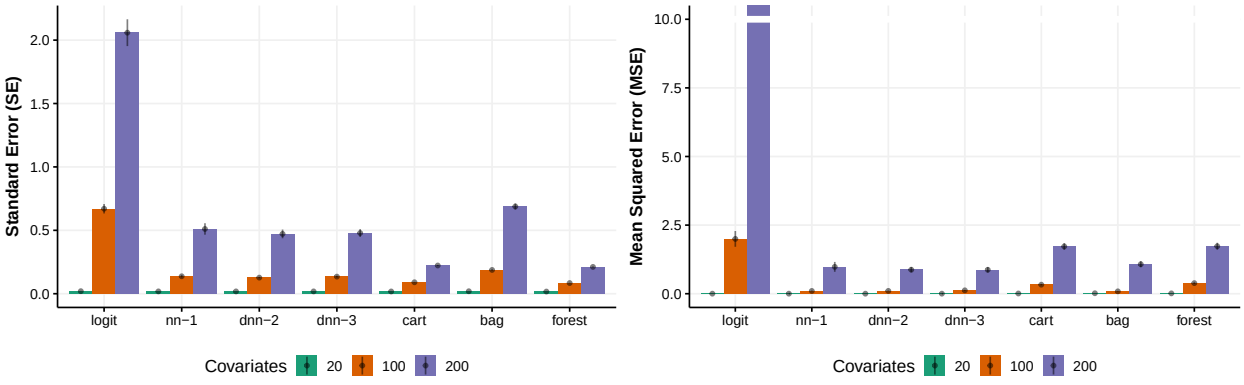
2.4.2.2.2 Summary My findings suggest variability in the bias of the estimated ATE depending on the method and covariate conditions. In the low-dimensional condition (p=20), most methods accurately estimated the true ATE with little bias and little sampling variability. As the number of covariates increased, the magnitude of bias also increased, with logistic regression overestimating the true ATE, while all other machine learning approaches produced less biased estimates. In the highest covariate condition (p=200), logistic regression severely overestimated the true ATE, while the machine learning methods estimated the true ATE with significantly less bias. Notably, my NN-1 and DNN-2 approaches were robust methods for estimating the ATE in the high-dimensional covariate setting, even when complexities were included in the population treatment and outcome model. In this condition, they produced bias estimates that were nearly 11 times lower than those produced by logistic regression in that same condition.

My findings also indicate that the presence of complexities in the population treatment assignment and outcome models significantly affects the accuracy of the ATE estimate. When both the treatment and outcome models include quadratic and interaction terms, there is a significant increase in bias and increased sampling variability across methods and covariate conditions. Logistic regression tends to perform poorly in high-dimensional, complex cases because the estimating model is still a linear model that does not account for quadratic and interaction terms. As a result, logistic regression may struggle to capture the true underlying relationships between the covariates and the treatment assignment, leading to biased propensity score estimates and, consequently, biased ATE estimates. These results highlight the importance of considering the complexities of the population treatment and outcome models when choosing a propensity score estimation method and suggest that more flexible, non-linear methods may be better suited for handling complex relationships in the data.

2.4.3 Variance in ATE Estimation: SE, MSE, and 95% CI Coverage

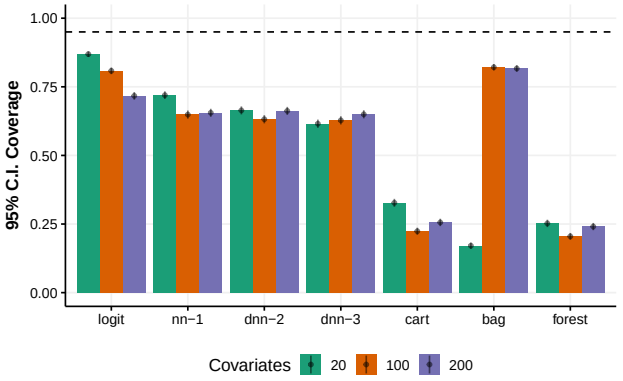
In this section, I evaluated the estimated variance in the ATE based on the SE, MSE, and 95% confidence interval coverage rate. The optimal method should have a low bias in the estimated ATE, minimal variability in terms of tight SE and MSE and a high 95% confidence interval coverage rate. To summarize my findings, I used the multi-panel Figure 2.10 to compare the performance of each method in terms of SE,

MSE, and 95% confidence interval coverage rate across different population models.



(a) Standard Error (SE)

(b) Mean Squared Error (MSE)



(c) 95% CI Coverage

Figure 2.10: Variability in ATE Across the Population Treatment and Outcome Model Conditions

2.4.3.1 Estimated Standard Error (SE) and Mean Squared Error (MSE)

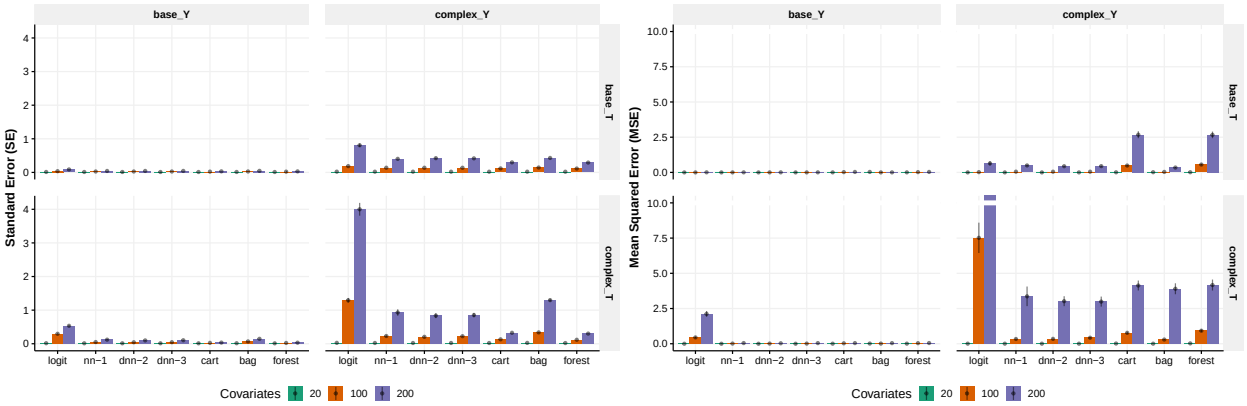
In the first panel of Figure 2.10, I present the standard error (SE) results of the estimated ATE by covariate condition. In the low-dimensional condition, all methods produced small standard errors. However, as the number of covariates increased, the SE of logistic regression and NN-1 increased significantly. At the same time, other machine learning approaches, such as CART and random forest, had smaller SEs (SE = 0.22 and 0.21, respectively). Despite the differences in the SE, all methods produced narrow 95% Monte Carlo confidence intervals for the estimated ATE, indicating high precision in the ATE estimate.

A high SE implies a high degree of uncertainty in the estimated ATE, with a wide range of possible values. In a true ATE of 0.3, even an SE of 0.2 (the lowest estimated SE) could be substantial. Such errors

could lead to incorrect conclusions about the effectiveness of the treatment and reduce the reliability of the ATE estimate. A high SE can also indicate that the model is not capturing the underlying relationship between the treatment and outcome, making the estimated ATE unreliable. Therefore, it is crucial to choose a method that produces a low SE given its relationship to statistical inference, in addition to low bias, to ensure reliability of the estimated ATE.

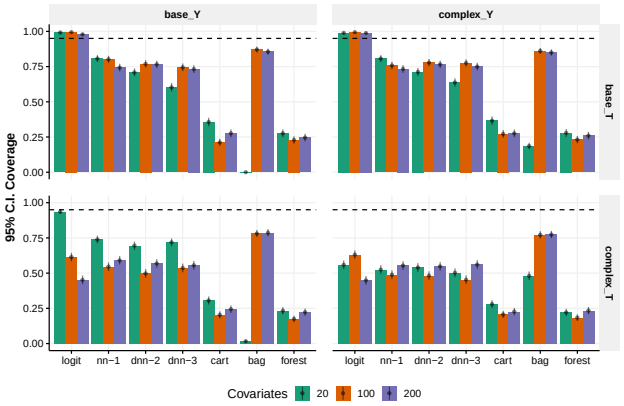
In the second panel of Figure 2.10b, I present the results for the mean squared error (MSE) of the estimated ATE. The MSE measures the average squared difference between the estimated and true values, with a lower MSE indicating a better fit. This metric provides a comprehensive summary statistic that considers both the bias and variance of the estimated ATE. Whereby an MSE > 1 indicates, on average, more that one point difference betwee the true ATE and the estimated ATE. In the low-dimensional condition (p=20), all methods produced low MSE values. However, as the number of covariates increased (p=200), logistic regression produced extremely high MSE values (MSE = 29.4). At the same time, the other machine learning approaches maintained low MSE values, with the best performance achieved by the DNN-2 and DNN-3 approaches (both with MSE = 0.87), which were 35 times lower than the MSE produced by logistic regression.

2.4.3.1.1 Complexities in the Population Outcome and Selection Models In Figure 2.11, I further investigate the performance of each method by evaluating SE, MSE, and 95% confidence interval (CI) coverage in the presence of complexities in the treatment and outcome models. The figure is divided into three panels, each representing the SE, MSE, or 95% CI coverage rate. In the upper-left facet, I show the results when only the main effects are included in both the population treatment and outcome models (Base T & Base Y). In the lower-right facet, I display the results when both the population treatment and outcome models include quadratic and interaction terms (Complex T and Complex Y). The deviations from the diagonal facets represent conditions where only one of the models includes quadratic and interaction terms while the other only includes main effects. These results provide insight into how the presence of complexities in the treatment and outcome models affects the performance of each method in terms of SE, MSE, and 95% CI coverage rate.



(a) Standard Error (SE)

(b) Mean Squared Error (MSE)



(c) 95% CI Coverage

Figure 2.11: Variability in ATE by Data Generating Mechanisms

2.4.3.1.1.1 Base - Population Outcome and Selection Models In the upper-left panels of Figure 2.11, I present the results when only the main effects are included in both the population treatment and outcome models. Regardless of the covariate condition, all methods performed well and produced low values for both the SE and MSE of the estimated ATE. This result indicates that the estimated ATE is precise and accurate when only the main effects are present in both models.

2.4.3.1.1.2 Complex - Population Outcome and Selection Models As the complexity of the population treatment and outcome models increased, I observed a substantial increase in the SE and MSE of the estimated ATE. In the 100-covariate condition, logistic regression had a high MSE value (MSE = 7.51). However, when the number of covariates increased to 200, logistic regression resulted in extremely high SE and MSE values (SE = 4.00; MSE = 115.1). In contrast, my DNN-2 and DNN-3 approaches had significantly lower MSE values, with MSE values below 2, which were nearly 60 times lower than the MSE value for logistic regression in the high-dimensional condition.

2.4.3.1.1.3 Deviations - Population Outcome and Selection Models As I observed with the ATE bias evaluation results, both the SE and MSE were more sensitive to complexities in the outcome model than when complexities were only included in the selection model. When complexities were only included in the outcome model, I found that the CART and random forest methods produced larger MSE values than all other methods. In the highest covariate condition, all other methods, including logistic regression, produced MSE values well below 2.5.

2.4.3.2 95% Confidence Interval Coverage

Finally, in the third panel (Figure 2.10c), I present my findings for the 95% confidence interval (CI) coverage rate of the ATE across different population outcome models. The dashed line indicates the desired coverage rate of 95%. This rate assesses how well each method captures the true ATE of 0.3 within the estimated 95% confidence interval. A good estimator has high coverage and reliably captures the true population ATE within the interval.

The machine learning approaches appear relatively insensitive to the different covariate conditions.

However, logistic regression seems more sensitive and shows a decrease in coverage rate as the number of covariates increases. Overall, only logistic regression in the low-covariate condition approached the nominal 95% confidence interval. In the high-covariate conditions, all methods had relatively low coverage, with coverage rates below 75%, while CART and random forest showed incredibly low coverage rates below 30%. The exception was bagged-CART, which had a coverage rate of 81%, followed by logistic regression with 71%. The high coverage rate of bagged-CART is likely due to the method using multiple decision trees based on bootstrap samples of the data, which reduces the variance of the model and improves its performance, especially in high-dimensional data. Additionally, it is important to note that logistic regression in the highest covariate condition also exhibited large bias and SEs. Thus, its relatively good performance in this condition may be due to the large amount of bias and sampling variability in the ATE.

2.4.3.2.1 Complexities in the Population Outcome and Selection Models In the third panel of Figure 2.11, I present the results of the 95% confidence interval (CI) coverage rate of the ATE by model complexities. The results show that the coverage rates were highly influenced by the complexities in the population selection model, with all methods having lower coverage rates than when only the main effects were included in the population selection model. Additionally, even when both the population outcome and selection models only had main effects, all methods other than logistic regression still had suboptimal coverage rates. Conversely, when complexities were included in both the outcome and selection model, logistic regression had inadequate coverage, especially in the high-covariate condition, where it had a coverage rate below 50%. The highest coverage rate was achieved by bagged-CART, at around 77%, followed by DNN-3, with a 56% coverage rate. A lower 95% CI coverage for the machine learning methods may indicate that they are not accurately capturing the relationship between the treatment and outcome, resulting in an unreliable estimate of the ATE.

2.4.3.3 Model Based versus Empirical Standard Error

A method can have low bias and low standard errors, but if the estimated confidence intervals fail to capture the true parameter at an acceptable rate (i.e., low coverage), it can lead to unreliable estimates of the ATE. Bias measures the systematic error in the estimate of the ATE, while accuracy refers to the

closeness of the estimate to the true value, taking both bias and variability into account. Low bias and low standard errors indicate a good fit between the estimated and true values, but they do not guarantee the estimate's accuracy.

One possible scenario that could lead to low coverage is that the estimated standard errors are biased away from the empirical standard errors. To analyze whether my estimated standard errors are biased away from the empirical standard errors, I computed the relative percent error in the standard errors using the following Equation 2.17:

$$RelativePercentError = 100(\frac{ModelBasedSE}{EmpiricalSE} - 1) \tag{2.17}$$

where Model Based SE is the estimated SE and Empirical SE is the empirical standard error taken as the standard deviation of the ATE estimates. A value greater than 0 indicates that the method overestimates the empirical SE, while a value below indicates that the method underestimates the empirical SE. Figure 2.12 presents the results of the relative percent error in SEs by method across population models and covariate conditions. The solid black dot indicates the mean relative error, and the diverging lines indicate the associated Monte Carlo 95% CI.

Regardless of the number of covariates, all methods underestimate the empirical SE. CART and random forest have the worst performance, with an underestimation of the empirical SE by more than 75%. While bagged-CART and all NN approaches (NN-1, DNN-2, DNN-3) underestimate the empirical SE by a smaller margin, it is still not ideal. This fact likely contributes to the suboptimal 95% CI coverage rates, especially in complex model conditions. These results may indicate that it would be beneficial to move away from a robust sandwich estimator, which relies on the normality assumption of the data, and instead use a bootstrapped SE that provides a more appropriate estimate of the uncertainty in the estimate of the ATE.

2.4.4 Weights Assessment

In this section, I present the propensity score weight assessment results. The assessment is an important step in ensuring unbiased treatment effects as it helps to identify extreme weights, which can result in

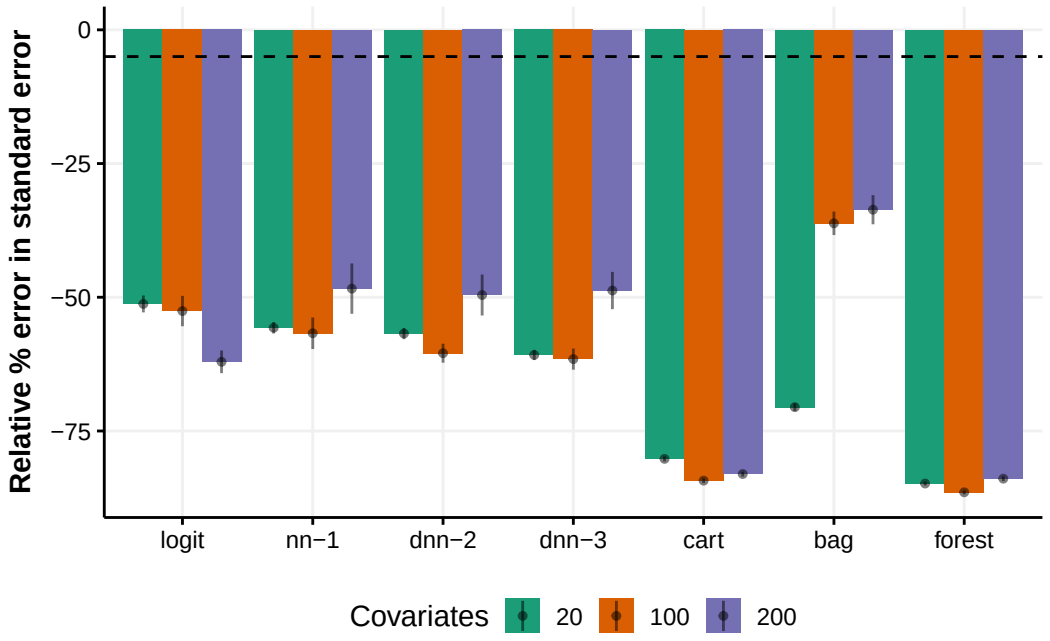


Figure 2.12: Relative % Error in the Estimated Standard Error Across the Population Treatment and Outcome Model Conditions

bias. The IPTW method was used to estimate the ATE, and the mean IPTW weight was calculated for each method. A considerable departure from 1 in the mean IPTW weight indicates the presence of extreme weights.

In Figure 2.13, the mean IPTW weights are displayed. All machine learning approaches produced weights below 2.5. In the low-dimensional case, all methods performed well with mean weights below 2. However, logistic regression produced extreme weights in high-dimensional cases with a mean of 22,587,000. This high value is likely due to logistic regression producing extreme propensity scores when 200 covariates are included in the estimation step, likely due to overfitting.

2.4.4.0.1 Complexities in the Population Outcome and Selection Models When evaluating the IPTW weights by the different complexities in the population outcome and selection models, I found that all methods produced acceptable weights below 2 (Figure 2.14) regardless of the condition. The only exception to this was logistic regression, which produced extreme weights whenever complexities were introduced into the population selection model. This effect was not seen when the base selection model only included the main effects.

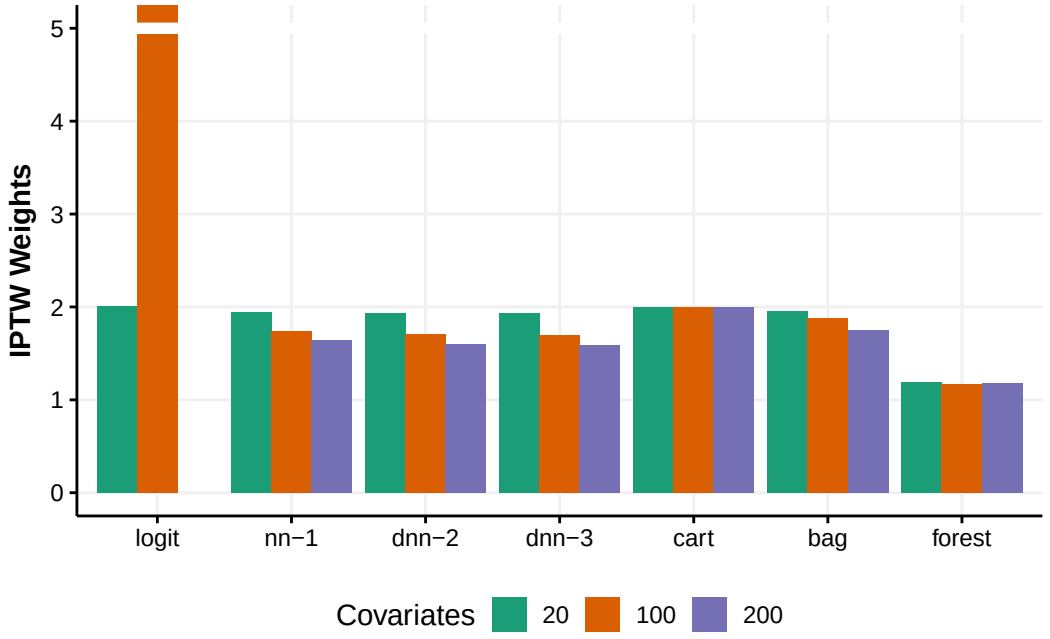


Figure 2.13: Mean IPTW Weights Across the Population Treatment and Outcome Model Conditions

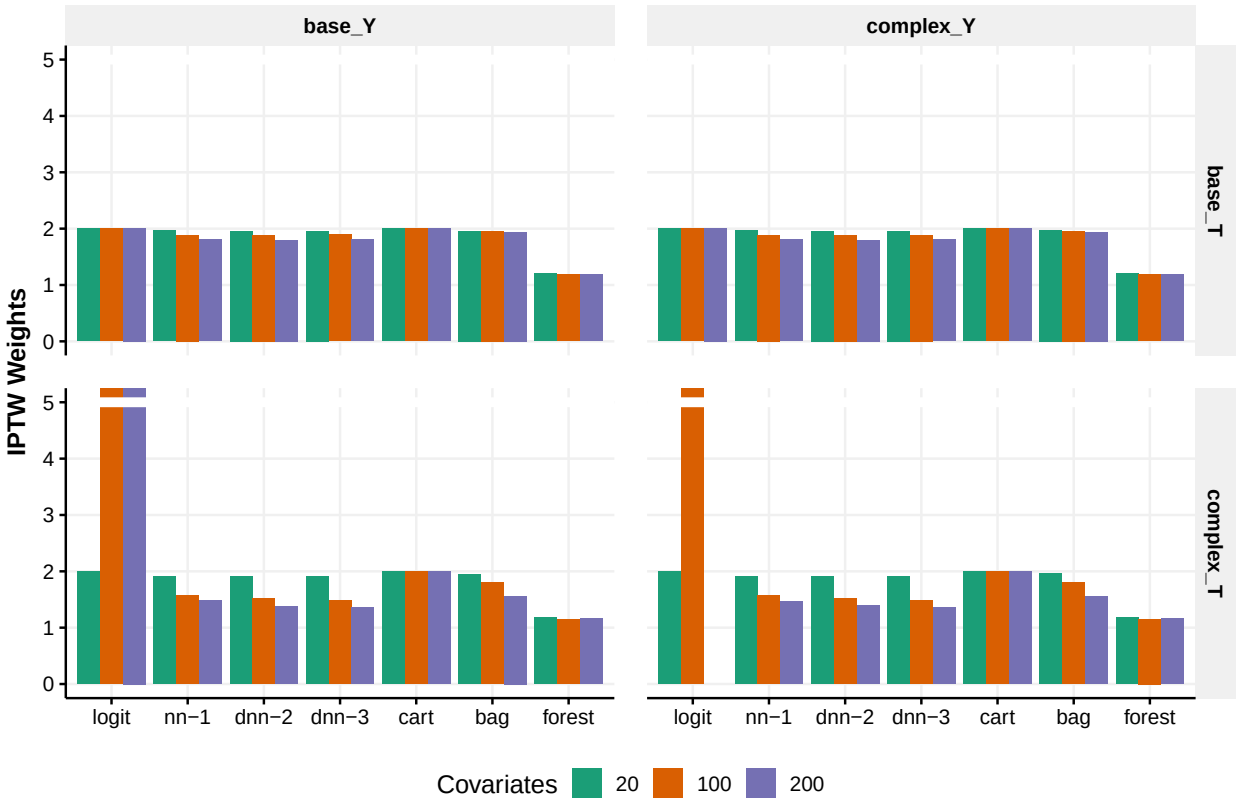


Figure 2.14: IPTW Weights by Data Generating Mechanisms

2.4.4.0.2 Summary The results from the IPTW weight analysis showed that all machine learning approaches produced acceptable weights below 2.5, except for logistic regression, which produced extreme weights in high-dimensional cases and when complexities were introduced into the population selection model. Overall, these results suggest that all methods, except for logistic regression, are robust to the complexities in the population outcome and selection models in producing acceptable IPTW's.

2.4.5 Power

Finally, I evaluated the statistical power of each method. Statistical power refers to the ability to detect a significant treatment effect at a specified alpha level, usually .05. High power indicates a low rate of type II error, meaning that the method can accurately detect real differences in the treatment effect. However, factors such as sample and effect size also impact statistical power. In this simulation, the sample size was fixed at 10,000, and the treatment effect was relatively small. The results of the power assessment are summarized below, but it is important to note that the results should not be generalized outside of the specific ATE of 0.3 used in this simulation.

In Figure 2.15, I display the results of the statistical power assessment, which was calculated based on a standard alpha level of 0.05. The results were averaged across the various population treatment and outcome models. In the low-dimensional condition, all methods were able to detect a significant treatment effect; however, as the number of covariates increased, a decrease in power was observed. This decrease was particularly pronounced for logistic regression, while the other machine learning methods showed a less steep drop in power.

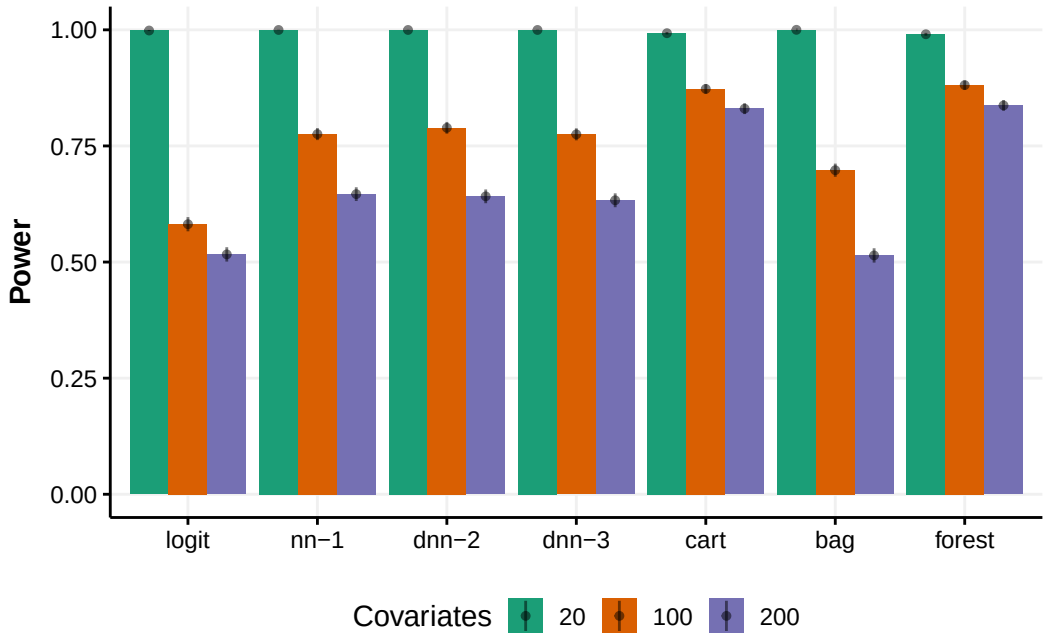


Figure 2.15: Power Across the Population Treatment and Outcome Model Conditions

2.5 Discussion and Conclusion

Quasi-experimental methods have gained popularity in educational research for estimating treatment effects without randomization (Imbens & Rubin, 2015; Rubin, 2005). One such method is propensity score weighting, which balances observed covariates between treatment and control groups using an estimated propensity score. However, accurately estimating the propensity score model can be challenging, especially in high-dimensional data settings (Hill et al., 2011).

The purpose of my study was to address the limitations in the existing literature on propensity score estimation in high-dimensional data settings. I compared the performance of a DNN approach with traditional logistic regression and other machine learning-based methods, including CART, bagged-CART, and random forest, to provide valuable insights into these methods' accuracy in estimating the population ATE. I evaluated these methods' performance under different covariate conditions and when complexities were introduced into the treatment selection and outcome models. This evaluation is critical given recent simulation studies have found that the presence of confounding variables and complexities in the treatment selection and outcome models can significantly impact the accuracy of the estimated ATE (Cannas & Arpino,

2019).

My results suggest that the choice of method for propensity score estimation should be based on the context and dimensionality of the data. In low-dimensional settings (p=20), I found that most methods, including logistic regression, produced accurate ATE estimates with little to no bias. However, as the number of covariates increased, the magnitude of bias also increased, with logistic regression overestimating the ATE. The other machine learning methods I tested produced less biased results, including CART, bagged-CART, random forest, and the neural network-based approaches (NN-1, DNN-2, DNN-3).

In high-dimensional settings (p=200), logistic regression severely overestimated the true ATE. On the other hand, the machine learning methods I tested, including the DNN-2 approach, could estimate the true ATE with significantly less bias and better covariate balance. When choosing between DNNs and bagged-CART, applied analysts should consider factors such as data complexity, presence of non-linear relationships, computational resources, and the desired level of interpretability. DNNs may be better suited for cases with more complex relationships, while bagged-CART may be preferred when interpretability is crucial and computational resources are limited.

Traditional logistic regression may not be suitable for estimating the ATE in high-dimensional covariate conditions, especially in the presence of non-linearity and non-additivity. Machine learning-based approaches, particularly DNNs, are promising alternatives. My findings align with other studies and emphasize the importance of the bias and variability of the estimated ATE when selecting the optimal estimation method. My study extends the literature by testing various methods with more complex and higher dimensional data than previous studies on propensity score estimation (Cannas & Arpino, 2019; Lee et al., 2010; Setoguchi et al., 2008).

Like other simulation studies, my study has limitations in terms of generalizability, meaning that my findings may not necessarily generalize outside the tested scenario. Although this simulation is based on real-world data from the Common App evaluation, making it more generalizable than simulations based on arbitrary data pulls, further research is still needed to evaluate the performance of the DNN-2 approach and other machine learning methods in different settings, including real-world datasets. To address this, I plan

to include a real-world data example in my publication, demonstrating how the DNN approach and other methods can be used to estimate treatment effects using actual data.

Another limitation of my study relates to inference. Although the DNN approach and bagged-CART produced unbiased estimates and outperformed the logistic model in the high-dimensional context, their coverage rates were suboptimal. This performance is likely due to the model-based SEs underestimating the true empirical SE. To address this, I plan to conduct a follow-up analysis using bootstrapping to estimate the SEs. Bootstrapping does not require assumptions of normality, which may be violated in high-dimensional and complex data, leading to poor estimation of SEs even with robust sandwich estimation (Efron & Tibshirani, 1986). This fact highlights the importance of considering the bias introduced in estimating SEs in any simulation study using propensity score estimation with high-dimensional data. The literature around variance estimation in propensity score weighting is heavily debated, and my study contributes to this ongoing debate.

Apart from the limitations mentioned earlier, it is essential to perform sensitivity analyses to assess the robustness of the treatment effect estimates obtained from these methods. Procedures for conducting sensitivity analyses is an area of ongoing research, and future work should focus on providing guidance for applied analysts on how to perform sensitivity analyses with these approaches.

In conclusion, traditional logistic regression may not be suitable for estimating the ATE in high-dimensional covariate conditions, especially in the presence of non-linearity and non-additivity. Machine learning-based approaches, particularly DNNs, are promising alternatives. It is essential to consider the limitations and challenges associated with these methods, choose the one best suited for the specific data context, and perform sensitivity analyses to assess the robustness of the treatment effect estimates. By doing so, researchers can make more informed decisions when estimating treatment effects and contribute to a better understanding of the effectiveness of different educational interventions.

Moving forward, several steps can be taken to ensure that these machine learning-based methods for propensity score estimation, such as DNNs, and their guidance are more accessible to applied research audiences. First, future research should focus on the development of user-friendly software packages and

tutorials that enable researchers to implement these methods without extensive programming knowledge. Second, more comprehensive guidelines for selecting the best method based on data context and research goals should be developed, including recommendations for model selection, hyperparameter tuning, and diagnostics. Third, as mentioned earlier, sensitivity analysis should be an integral part of this research, and guidelines for conducting sensitivity analyses with machine learning approaches should be established. By addressing these needs, the research community can provide applied analysts with the necessary tools and knowledge to make better-informed decisions when estimating treatment effects through observational data.

3.0 Paper 2: When In-Person Support Is Not Possible, Can Virtual Outreach Help? Evaluating the Impact of an Artificially Intelligent Conversational Chatbot to Promote College Enrollment During the COVID-19 Pandemic

3.1 Introduction

In spring 2020, schools and colleges across the US began to shut down and pivot to remote learning to protect students and their communities from the rapid spread of the COVID-19 virus. At the same time, high school seniors who had applied to college before the pandemic began receiving acceptance notifications, even though the pandemic's impact on college enrollment was unknown. On the one hand, the COVID-19 crisis hampered business-as-usual operating procedures for colleges. As a result, students, particularly those planning to live on campus during college, may have been more likely to postpone their college enrollment. On the other hand, higher education is a counter-cyclical industry, such that college enrollment rates tend to be higher during economic downturns. For recent high school graduates, the impact of the pandemic on the retail and service sectors severely limited job opportunities. By April 2020, the unemployment rate in the United States reached 14.5% – the highest rate recorded in recent history – with the service and retail sectors experiencing the bulk of job losses. (US Bureau of Labor Statistics, 2021). As a result, college may have become more appealing to young adults who would otherwise enter the workforce after high school.

To address this uncertainty, the Common Application (Common App), a non-profit organization dedicated to assisting students with college applications, partnered with Mainstay, an EdTech company focused on using behavioral intelligence for college access, and the College Advising Corps (CAC), a non-profit that places recent college graduates as college counselors in underserved high schools. These partnered organizations acted quickly to provide students with proactive outreach and guidance on college-going tasks through an innovative, large-scale chatbot campaign. This outreach targeted around 174,000 US high school students who were the first in their families to attend college and came from low-income families. The majority (83%) of these students were also racially marginalized. Due to various systemic barriers, this

subset of students were disproportionately likely to have had limited access to quality health care, living, and working conditions during the pandemic (Hernandez et al., 2022; Ornelas & Solorzano, 2004; Oromaner & Oakes, 1986; Soria et al., 2020). Additionally, this group of students were less likely to have ready access to personalized college-going guidance and potentially stood to benefit from additional support during their precarious college transition.

Over 38 weeks, an artificially intelligent (AI) chatbot named *Oli* sent students scripted messages on various college search, application, and enrollment-related topics. Staff developed the scripted messages at the Common App in collaboration with CAC. They included messages reminding students about upcoming deadlines, such as "Hey , #CollegeSigningDay is this Friday, May 1st. Have you decided what school you're going to attend?" To better target the information, Oli solicited information directly from students about the types of resources they needed and pressing questions they had. The chatbot "learned" how to interact with students by first having staff at the partnered organizations seed the chatbot with frequently asked questions and answers. As Oli interacted with students, it built an ever-increasing knowledge base for student questions that aided the chatbot in providing more targeted and individualized real-time responses. As the intervention rolled out, Common App and CAC staff could monitor interactions and correct and add to Oli's knowledge base. Student questions that Oli could not answer were forwarded to college advisers at the CAC, who would follow up directly with individual students.

In recent years, researchers have developed an interest in understanding how insights from behavioral economics can inform strategies to support students to and through college (Oreopoulos, 2020, 2021). Several of these studies have involved designing and testing proactive communication strategies using text messaging as the primary mode of communication (Page & Scott-Clayton, 2016). Recent studies have investigated text messaging communication employing artificially intelligent chatbot capability (Nurshatayeva et al., 2021; L. Page et al., 2020; Page & Gehlbach, 2017). These studies have demonstrated that chatbots can be effective in assisting students to complete pre-matriculation tasks and enroll in college (Nurshatayeva et al., 2021; L. Page et al., 2020; Page & Gehlbach, 2017). However, there are numerous open questions regarding the scalability of these chatbot interventions. Text nudge interventions have mixed results when implemented at scale, with many studies finding minimal to no effects on college-related outcomes (Bird et al., 2021;

Oreopoulos, 2020).

Therefore, my second dissertation study aims to contribute to the literature on the scalability of chatbot interventions by evaluating the large-scale chatbot campaign implemented by the Common App, College Advising Corps, and Mainstay aimed at low-income, first-generation students. Specifically, I investigate whether the college applications and enrollment rates were higher among students targeted for this chatbot outreach compared to their observationally similar peers who were not targeted. In addition, I examine treatment heterogeneity based on students' application behavior, racial/ethnic identity, and the level of chatbot engagement.

The remainder of the section is structured as follows: First, I review the relevant literature for this study. Then, I describe the data and my analytic approach. Finally, I conclude with a discussion of findings.

3.2 Literature Review

In this section, I review the literature on college access in general as well as during the COVID-19 pandemic. In addition, I draw on literature examining how behavioral text message nudge interventions and chatbots can assist high school students in overcoming obstacles to college access.

Before reviewing the literature, I first define how I conceptualize first-generation students – students whose parents do not have any postsecondary education experience – and the systemic barriers they face in college access. I take an anti-deficit approach to my framing of first-generation students – many of whom also tend to be economically and racially marginalized (Harper, 2010; Hernandez et al., 2022). Anti-deficit practices explore how marginalized students persist in the face of social and structural barriers to reframe students' experiences from ones of struggling to ones of resiliency (Harper, 2010). Since most deficit approaches focus on what students lack, anti-deficit strategies focus on what students bring that leads to their resilience and persistence. Although I take an anti-deficit stance in my review of the literature, I recognize that the majority of the existing literature on college access, especially that focuses on economically and racially marginalized students, is written from a deficit perspective (McLewis, 2021).

It is important to acknowledge the structural and social barriers that lead to disparate postsecondary outcomes for first-generation students. For example, economically and racially marginalized individuals face structural barriers to accessing quality education, housing, and employment conditions . Decades of federal redlining in housing and the policy of using local taxes to fund schools, for instance, produce racial and socioeconomic disparities in educational opportunities and family wealth (Hernandez et al., 2022; Ornelas & Solorzano, 2004; Oromaner & Oakes, 1986; Soria et al., 2020). In addition, racially marginalized students face social and systemic barriers and federal policies that have historically limited their access to high-quality primary and secondary education (Hernandez et al., 2022; McLewis, 2021; Ornelas & Solorzano, 2004). For example, policies that funnel racially marginalized students into underserved schools with limited access to college prep courses, and school tracking policies that exclude racially marginalized students from gaining access to the college prep tracks that would qualify them to apply to college (Oakes & Rogers, 2007; Ornelas & Solorzano, 2004; Oromaner & Oakes, 1986). As a result, disparities in college access and completion based

on parental education, as well as social and structural barriers, are significant contributors to the rising income inequality in the US.

3.2.0.1 College access and returns to a college education

Despite increased access to higher education over the past several decades, disparities remain in college attainment by socioeconomic status and parental education (Dynarski, Page, et al., 2022; Dynarski, Nurshatayeva, et al., 2022). Students whose parents have a college degree are more likely to attend college than those whose parents did not complete high school (Avery et al., 2021; Dynarski, Page, et al., 2022). Moreover, students from higher-income backgrounds are more likely to enroll and complete postsecondary education than students from lower-resourced families (Garriott, 2020; Ornelas & Solorzano, 2004). By age 24, roughly 13% of students from the lowest family income quartile earn a bachelor's degree, compared to 64% of students from the highest quartile (Institute, 1956).

Despite the rising cost of higher education, the non-pecuniary and long-term economic returns to a college education remain positive (Dynarski, Page, et al., 2022). Adults with a bachelor's degree typically earn and accumulate more wealth than those without a degree (Dynarski, Page, et al., 2022). Higher education is associated with lower unemployment rates, greater civic engagement, and better health outcomes Avery et al. (2021). Nevertheless, the rate of economic return is higher for continuing education students – students whose parents had some postsecondary education experience – than for first-generation students (Fry, 2021) . Given that more than half of college students identify as first-generation (56%), substantial policy efforts have focused on helping first-generation college-bound students enroll in college (Carrell & Sacerdote, 2017; Herbaut & Geven, 2020; Page & Scott-Clayton, 2016). However, the COVID-19 pandemic introduced unprecedented complexity to these efforts and substantially hindered college access for first-generation students (Hoover, 2020).

3.2.0.2 College access during the COVID-19 pandemic

The COVID-19 pandemic added uncertainty to the lives of high school seniors planning to attend college in the fall of 2020. In the early stages of the pandemic, one in every six recent high school graduates

reconsidered enrolling in college, and nearly two-thirds expressed reservations about attending their first choice institution (Howell et al., 2021). The most frequently cited reason for not enrolling at their first-choice institution was concern about their family's ability to pay for college (Howell et al., 2021).

Now, nearly three years into the pandemic, the adverse effects of the pandemic on college enrollment have become clear. Enrollment for first-time undergraduates fell to historic lows in fall 2020, falling nearly 10 percentage points from fall 2019, with lower immediate college enrollment rates for students from low-income schools compared to high-income schools, where low-income schools are defined as schools where at least 50 percent of the students are eligible for free or reduced-price lunch (Clearinghouse, 2021). Furthermore, students from low-minority schools were more likely to enroll in college right away than those from majority-minority high schools, defined as schools where at least 40 percent of students are Black or Latinx (Clearinghouse, 2021).

A recent report from the College Board provides additional information about the detrimental effects of the pandemic on college enrollment. Using data from nearly 10 million US students, the College Board calculated high-level descriptives and regression-adjusted models to determine the proportion of fall 2020 enrollment rates attributable to COVID-19 (Howell et al., 2021). As a result of the pandemic, enrollment rates at community colleges decreased by roughly 12 percentage points. Most concerning is the decline in enrollment rates among first-generation, low-income students (Howell et al., 2021).

The decline in college enrollment rates for first-generation students is likely a result of the pandemic's disproportionate impact on racially and economically marginalized communities. During the pandemic, these communities experienced reduced access to high-quality healthcare and working conditions and lowered expected wages (Hoover, 2020; Lee et al., 2021; Molock & Parchem, 2022). Prior to the pandemic, the average Black and Latino household had a net worth of approximately $17,000, while the average White household held $171,000 (Center for American Progress, 2021). As a result, Black and Latino households may have had a more challenging time adjusting to the economic shock caused by the pandemic. These households, on average, have fewer liquid assets and wealth to cover unexpected expenses and wage loss (Center for American Progress, 2021). This may have resulted in first-generation students – notably racially marginalized

students – prioritizing immediate health and economic stresses, such as supporting their families financially or dealing with lost wages, over college enrollment. Indeed, students from higher-income households were far more likely to enroll in college immediately after graduation (65%) than those from low-income high schools (49%) (Clearinghouse, 2021). Furthermore, first-generation students were nearly twice as likely to be concerned about paying for education expenses in fall 2020 than continuing education students (Soria et al., 2020). In sum, there has been an unprecedented change in college enrollment caused by the COVID-19 pandemic, particularly for first-generation, economically, and racially marginalized students. Next, I review the literature on well-documented challenges to college access.

3.2.0.3 Challenges in college access

First-generation students face many barriers during the college application, financing, and enrollment processes (Avery et al., 2021; Page et al., 2019; Page et al., 2022; Page & Scott-Clayton, 2016). To apply for college, students must complete several complex tasks, including creating a list of colleges, deciding which colleges to apply to, and applying for financial aid via a time-consuming financial aid application (the Free Application for Federal Student Aid - FAFSA). Students typically obtain federal, state, and grant aid – aid that does not have to be repaid – by completing the FAFSA form, which contains over 100 questions about the students' and their families' financial assets. Several behavioral nudge interventions have demonstrated that proactive outreach to students in completing the FAFSA increases college enrollment (Castleman & Page, 2014; Page et al., 2019; Page et al., 2022; Page & Scott-Clayton, 2016). Assisting students with FAFSA submission is only half the battle. The FAFSAs of a subset of first-generation and low-income students are selected for additional scrutiny through a federal auditing system, leading to lower college enrollment rates among low-income, first-generation students (Guzman-Alvarez & Page, 2021). Additional obstacles remain once a student accepts a college admission offer, notably during the summer before college.

The summer before college enrollment is fraught with additional obstacles to college enrollment (Castleman et al., 2012, 2014; Castleman & Page, 2015), given that students must complete a number of pre-matriculation administrative tasks before stepping foot on campus, such as submitting high school transcripts, determining which and how much financial aid to accept, taking placement exams, and success-

fully enrolling in courses (Castleman et al., 2012, 2014; Castleman & Page, 2014, 2015). If students fail to complete these steps, they may ultimately fail to enroll in college, a process termed "summer melt" (Castleman et al., 2014; Castleman & Page, 2014, 2015). Summer melt affects 10 to 20 percent of all college-bound high school students, with higher rates for first-generation, racial, and economically marginalized students. Students who, due to an array of systemic barriers, have reduced access to information and support to help them navigate these administrative tasks (Cataldi et al., 2018; Ives & Castillo-Montoya, 2020).

3.2.0.4 Unique challenges in college access among first-generation students

For first-generation students with limited family or school-based resources that can facilitate the college application process, navigating these pre-matriculation tasks adds an additional layer of complexity. Moreover, these students must navigate the transition to college while balancing work and family obligations (Barber et al., 2020; Hernandez et al., 2022). First-generation students are more likely to attend schools with fewer college prep or Advanced Placement (AP) courses, which have been linked to higher college enrollment (Ornelas & Solorzano, 2004). Additionally, first-generation students are more likely to attend schools with limited or no access to college counselors, who could assist students with completing college applications and navigating the complex financial aid process (Castleman & Page, 2014; Naughton, 2021; Page et al., 2019). Nevertheless, even when students *do* have access to a college counselor, these counselors often manage high caseloads and cannot provide individualized support to students who would benefit the most (Naughton, 2021). Therefore, many scholars have focused on developing interventions focused on assisting these students with pre-matriculation tasks by providing access to individualized advising from a professional college counselor (Castleman et al., 2012, 2014; Castleman & Page, 2014, 2015).

3.2.0.5 Importance of college advising

A large body of work has been conducted to assess the efficacy of behavioral interventions designed to assist students in meeting the key steps and deadlines required to enroll in college. One favored approach is to deliver targeted information about the college-going process to students through "low-touch" efforts such as proactive text-based outreach or "high-touch" face-to-face college advising (Oreopoulos, 2020). Access to

college advising can positively affect college enrollment and persistence (Arnold et al., 2015). Castleman et al. (2012) provide one of the earliest pieces of evidence of the positive impact of providing proactive outreach and support around key college-going tasks to low-income students during the summer prior to college. Using a randomized trial, Castleman et al. (2012) demonstrate that access to "college-transition" counselors during the summer before college increases students' immediate college enrollment by 14 percentage points. These counselors supported students in reviewing financial aid packages, addressing gaps in the aid they were offered, and completing required pre-matriculation paperwork. One limitation of this study was that only "choice" schools were included in their sample. Therefore, questions remained unanswered as to whether this support was generalizable to public schools.

A follow-up study by Castleman, Page and Schooley (2014) provided evidence of the generalizability of counseling support in public schools. The researchers conducted two college counseling interventions with large urban public school districts that were more nationally representative in Boston and Fulton County, Georgia. Like Castleman et al. (2012) the study provided high school seniors with college counseling support in the summer before college. For example, counselors helped students develop a list of personalized tasks they needed to complete to enroll in college in the fall. Counselors communicated with students using various modalities, including in-person, text, phone, and email. Students who received proactive college counseling in the summer before college increased their college enrollment by 8 to 12 percentage points among low-income students.

According to Naughton (2021), the pandemic exacerbated pre-existing "cracks" in the ability to provide students with advising services, transforming them into "craters." This is primarily due to high schools shifting from in-person to virtual advising. College advisors felt ineffective in providing virtual support and guidance once high schools transitioned to virtual advising. Advisors' ability to contact students was severely limited, leaving students without support to help them transition to college (Naughton, 2021). Naughton explains that advisers in high schools with higher college enrollment rates were significantly less affected than those in high schools with lower college enrollment rates, i.e., the students who required the most assistance. Therefore, strategies that bring virtual proactive outreach and support around critical college-going tasks directly to students who require it most have the potential to improve college-going outcomes for students

in these settings.

3.2.0.6 Behavioral nudge interventions

Behavioral nudge interventions have proven to be a low-cost means of providing college students with counseling support (Oreopoulos, 2020; Page & Scott-Clayton, 2016). These interventions condense complex information and deadlines into brief, to-the-point messages that are delivered to students via a familiar mode of communication – text messages (Oreopoulos, 2020). In the context of college access, these studies have focused on preventing "summer melt" by assisting students with well-defined but complex tasks such as submitting financial aid applications, submitting high school transcripts, and registering for classes (Castleman et al., 2012, 2014; Castleman & Page, 2014, 2015).

Castleman and Page have conducted several randomized controlled studies utilizing informational text nudges to remedy summer melt (Castleman et al., 2014; Castleman & Meyer, 2020; Castleman & Page, 2014, 2015; Castleman & Page, 2016). For instance, Castleman and Page (2015) use a personalized text message campaign that reminds students about key college-related tasks required to enroll in their receiving college institution. In addition to reminders, students were invited to request follow-up support from a counselor via text message. The text message campaign increased college enrollment by approximately 7 percentage points among students with limited access to college access supports. The low-cost nature of these interventions has increased interest in their use over the past decade (Oreopoulos, 2020). Recent studies have investigated the scalability of text nudge interventions, with findings that have been much more mixed. A recent study by Bird et al. (2021) suggests that nudges may fail when scaled up. Bird and colleagues investigated student nudges using multiple modalities, such as email and text messages, to encourage students to complete the FAFSA and enroll in college. This study is unique because it targeted nearly 800,000 students across the US who used the Common Application or the Texas admission system. No form of outreach (email or text message) increased financial aid receipt, college enrollment, or persistence. One possible explanation for this finding is that scaling nudge campaigns can be difficult when implemented by a large organization, such as the Common App, because students may lack a personal connection to a large organization, as opposed to their local high school. In order to deliver nudges to a large number of students, the authors suggest that

the messaging may be too "generic and one-way" and lack personalization, resulting in students not taking actionable steps to enroll in college.

In a recent working paper, Page et al. (2022) hypothesize what factors may contribute to the (in)effectiveness of informational text-based nudges in the college access space. One of their primary arguments is that students may be more receptive to text-based nudging when the messaging focuses on tasks and processes that are time-sensitive and have real consequences if not completed, such as resolving a registration hold. In addition, they provide more nuance to the claim made by Bird et al. (2021) that student messaging needs to be personalized to be salient to students. The ostensible sender of the messaging matters. Students may be more receptive to outreach and communication delivered by organizations or individuals that students know and trust (Debnam, 2017; Page et al., 2020). Recent studies have utilized technological advancements to help bridge this personalization gap.

3.2.0.7 Chatbots in college access

Recent efforts in the literature on behavioral nudges have focused on utilizing artificial intelligence (AI) chatbots to provide personalized assistance to students navigating college applications and enrollment. These conversational chatbots can provide students with personalized information about deadlines and answers to specific questions about the college application process. These bots are trained by college staff to develop a "database" of answers to frequently asked questions regarding college attendance. Over time, the chatbot "learns" how to respond to a broader range of student questions. Once trained and deployed successfully, the chatbot can assist students with answering questions about vital college-going tasks and requirements.

To date, three empirical studies have demonstrated the effectiveness of chatbots on college access (Nurshatayeva et al., 2021; L. Page et al., 2020; Page & Gehlbach, 2017). The first of these studies was conducted by Page and Gelbach (2017) at Georgia State University (GSU). Page and Gelbach collaborated with GSU and an ed-tech startup (Mainstay; formerly AdmitHub) to develop "Pounce," a conversational chatbot. Pounce assisted students accepted to GSU with navigating pre-matriculation tasks. The chatbot outreach led to a 3.3 percentage point increase in timely enrollment at GSU and helped students with pre-matriculation tasks like signing up for orientation.

A follow-up study at GSU (L. Page et al., 2020) shifted the focus of the chatbot outreach from helping students enroll at GSU to supporting them once they arrived at GSU by assisting them in completing tasks such as resolving registration holds and seeking campus-based resources. A novel aspect of this randomized study was its focus on the main campus of GSU-Atlanta, a four-year university, and the Perimeter campus of GSU, a two-year college. The chatbot effectively changed student behavior regardless of institution type when outreach addressed serious and time-sensitive administrative processes, such as resolving registration holds. In contrast, outreach was ineffective when chatbots assisted students with "non-urgent" tasks, such as gaining access to supplemental, academic, social, and career-related support. This study demonstrates that chatbot interventions work best when they target discrete, time-sensitive, and well-defined tasks. These findings are consistent with previous text-nudge interventions in which students were less responsive to nudges related to less time-sensitive tasks such as future job prospects (Oreopoulos, 2021). It is important to note that the previously mentioned studies occurred within the GSU system. Consequently, it was unknown whether the positive effects of chatbot interventions could be replicated in institutions outside the GSU system.

Nurshatayeva et al. (2021) argue that the effectiveness of chatbot interventions are context-dependent. Nushatayeva and colleagues replicate the work of Page and Gelbach (2017) using "PeeDee," a chatbot at East Carolina University (ECU). This study provides additional evidence of treatment heterogeneity by determining for which students and under what circumstances the intervention was most effective. Researchers were primarily interested in the use of PeeDee to improve students' completion of pre-matriculation tasks and their eventual enrollment at ECU. They find no overall effect of the chatbot on college enrollment but did find an impact on loan acceptance. Students in the treatment group had an 8 percentage point increase in accepting a student loan compared to the control group. Nushatayeva et al. attribute the null overall results on college enrollment to ECU's comparatively more affluent student body. The authors did find that the chatbot increased enrollment at ECU and course enrollment by 3 percentage points, specifically for first-generation students. This positive effect is likely a result of first-generation students receiving crucial information regarding college enrollment tasks that they cannot obtain through familial support.

Collectively, these studies demonstrate that chatbots can help to improve student success with the transition to college. Nevertheless, chatbot-based nudges are not effective in all contexts. These interventions

appear most effective when chatbots assist students with time-sensitive, actionable tasks, such as clearing registration holds, rather than "non-urgent" tasks, such as searching for on-campus resources. In addition, the sender of the messages matter. Students appear most receptive to communication and messaging from a source they know and trust. Text-based nudge interventions in which the ostensive messenger is well-known to the student or affiliated with a receiving institution – such as Pounce at GSU – have relatively low opt-out rates (5 percent). In contrast, when the messaging comes from a less well-known messenger, the opt-out rates are significantly higher, as high as 25 percent. These findings are consistent with the results of the more mature literature on general nudges in college access – context matters. Yet, little is known about the scalability of these interventions, such as at the national level, where hundreds of thousands of students are targeted.

Therefore, my second dissertation study contributes to the literature on the scalability of chatbot interventions by evaluating the large-scale chatbot campaign implemented by the Common App, College Advising Corps, and Mainstay aimed at low-income, first-generation students.

3.3 Data

We leveraged several different data sources to evaluate the effectiveness of the outreach campaign. First, we utilized administrative records from the Common App, which provided a rich data source of any information a student provided on their Common App application, including student demographics, high school academic achievement, and college entrance exams. Second, Mainstay provided access to the data from the chatbot platform utilized to provide outreach to students. This platform captured rich data on the content and timing of the chatbot communication among students, the chatbot, and CAC advisers. Using the Mainstay data, we generated variables that captured student opt-out requests and engagement throughout the intervention. Finally, we matched students in our analytic file to data from the National Student Clearinghouse (NSC) to observe if and where students enrolled in college in fall 2020 and persistence in subsequent terms.

In order to evaluate the effectiveness of the outreach campaign, our analysis relied on matching students selected to receive outreach to similar students in the same high school who were not selected for outreach. Common App and partners selected for outreach all students from the high school class of 2020 who had created a Common App account and met two specific criteria: 1) they would be first-generation college students, and 2) they had a low family income, as indicated by qualifying for a Common App fee waiver. In particular, some Common App students met one but not both criteria for inclusion in the outreach (i.e., first-generation college student *or* qualified for a fee-waiver). We use these students as our key source of comparison in the analyses. Therefore, our outreach ("treatment") group includes students who met both inclusion criteria (i.e., first-generation college student *and* qualified for a fee-waiver) while our control students met one but not *both* criteria.

From the Common App we received student-level data for the entire cohort of students who had created a Common App account and had intended to apply to college – hoping to enroll in fall 2020 – including students who were selected for the outreach ($n = 173,776$) and those who were not ($n = 1,229,232$), for a total of nearly 1.5 million student records. Once we cleaned the data, we reduced our analytic sample to include students who lived in the US, had a valid cell phone number, and attended a high school where at

least one student was targeted for outreach. Our final analytic sample included nearly half a million students ($N = 406,236$), including 142,837 students who were targeted for outreach and 263,399 who were not treated and therefore qualified as potential control students.

3.4 Method

3.4.1 Propensity Score Matching

It is important to note that the outreach team did not randomly assign the outreach for this campaign. Randomly assigning the outreach would ensure that students who received the outreach and those who did not would be balanced on both observable *and* unobservable characteristics. Given the lack of randomization, a simple comparison of outcomes between students who received the outreach and those who did not may – incorrectly – conclude that the outreach affected the outcome. To guard against this to the fullest extent possible, we matched students in the outreach group to demographically similar students in the same high school who were not treated. These matched students are then observationally similar to students in the outreach group and, therefore, a reasonable comparison group. Analytically, we do this via propensity score matching.

Our matching approach relies on estimating a propensity score, which is the conditional probability that a student would have been exposed to treatment, conditional on a set of observable characteristics. We then use this propensity score to match treated students to control students with similar estimated propensity scores, resulting in a control group that is demographically similar to the outreach group. Propensity score matching allows us to more accurately evaluate the impact of the outreach than a simple comparison of outcomes between those students who received the outreach and those who did not.

Specifically, we estimate a propensity score model of the following general form Equation 3.1:

$$Outreach_{is} = \beta_0 + \beta_1 X + \lambda_{is} \tag{3.1}$$

Where *Outreach* represents a binary indicator coded as 1 for students who received the outreach and

0 if a student did not. X is a vector of student-level characteristics. These include age, gender, English spoken at home, dependent flag, number of high schools attended, high school GPA, SAT/ACT performance, college credit exams count, TOEFL indicator, sibling college indicator, submitted at least one Common App application before start of outreach, and race/ethnicity indicators.

To increase balance on our covariates, we placed additional restrictions on our matching algorithm to exact match on specific variables. λ represents a vector of covariates on which we exact matched students. These include students' high school, underrepresented minority indicator, an indicator for submitting at least one Common App application before the start of outreach, and missing indicators for high school GPA and SAT/ACT performance. Exact matching on these variables allows us to match outreach students to control students who *exactly* match them on the predefined set of variables. For example, exact matching on high school forces the matching algorithm to only find potential matches for treated students within the same high school. This allows us to control for the fact that students across different high schools have varied school experiences; therefore, matching within schools provides more reasonable matched control students.

Given our nearest-neighbor matching approach, one control student can be a perfect match for more than one treatment student. Therefore, we allowed the matching algorithm to match an outreach student to multiple control students (i.e., matching with replacement). Additionally, we restrict matches to within .50 standard deviations of a propensity score. This allows us to guard against having treatment students matched with control students with a large difference in propensity scores (Guo et al., 2014).

To examine how well our matching procedure balanced student characteristics, we look at the overall distribution of the propensity score among treatment and untreated students and covariate balance before and after matching. Figure 3.1 presents the propensity score distribution between treatment and control students before and after matching. Overall, our matching procedure did an excellent job of matching treated students to control students, as indicated by the overlap in the distributions on the right panel in Figure 3.1, indicating that our matched sample has propensity scores that fall within the region of common support (Bai, 2011).

In addition to analyzing the distributional differences in the propensity score, we also looked at baseline

characteristics between treatment and control before and after matching. Table 3.1 compares balance of student characteristics before and after matching. Overall, control group students (who are either "low-income" or "first generation" but not both) had higher GPA class rank (70.1% vs. 64.3%) and SAT scores (794.9 vs. 671.3) than treatment group students (who are both "low-income" and "first generation"). The treatment group also included a substantially larger percentage of minority students (28% Black and 41% Latinx) than the control group (19% Black and 23% Latinx). We achieve excellent balance on all covariates included in our matching approach, with no standard mean difference below 0.1 (Rosenbaum & Rubin, 2022). Our final matched sample includes a treatment group consisting of 99,593 students and a matched control of 61,553.

Finally, we modify our primary propensity score model to explore variation in outreach impact on college enrollment by students' application behavior, racial/ethnic subgroups, and level of engagement with the chatbot. Where this is the case, we rerun our matching algorithm within our subgroup of interest.

3.4.2 Outcome Model

We ran a series of regression models to estimate the relationship between outreach and student outcomes. Specifically, we regressed each outcome on an outreach indicator and student-level characteristics, including matching weights.

Our regressions took the following general form Equation 3.2:

$$Y_{is} = \beta_0 + \beta_1 Outreach_{is} + X\theta + \lambda_s + \epsilon_{is} \tag{3.2}$$

Where Y_{is} represents our outcome of interest for student i in school s. *Outreach* is a binary indicator coded as 1 for students who received the outreach and 0 if a student did not. X is a vector of student-level characteristics such as age and gender. In addition, we included high school fixed effects (λ), which soak up any remaining variation in the outcome due to differences across high schools. β_1 is our coefficient of interest and represents the mean controlled difference in the outcome between those who received the outreach and those who did not. We clustered our robust standard errors at the high school level.

Although we took a robust matching approach in constructing our control group, there could still be differences between our outreach and control group on student-level characteristics. Therefore, by including the same vector of student-level covariates (X) as was included in Equation 3.1, in our regressions, we can account for any remaining imbalance between student characteristics that we observe and increase the precision of our impact estimates. This approach is commonly referred to as a "doubly robust" approach (Rosenbaum, 2010).

In order to account for our nearest-neighbor matching approach with replacement, we include matching weights (w) in our regression models. Where within each matched group – which includes at least one treated unit and at least one control unit – each treated unit gets a $w = 1$, and each control unit is given a preliminary weight of $w = n_{ti}/n_{ci}$, where n_{ti} is the number of treated units in matched group i and n_{ci} is the number of control units in matched group i. Then each control unit's weight is added up across the groups in which it was matched and scaled to sum to the number of uniquely matched control units.

3.5 Limitations

The main limitation in our analysis is our ability to attribute a causal relationship between the outreach and the effects we estimate. Given that students were not randomly assigned to the outreach, we cannot make causal claims about the effectiveness of the outreach. However, our robust matching approach allows us to reduce some – but not all – the bias that could exist in explaining the relationship between receiving outreach and outcomes. Note that we are only matching on *observable* characteristics. There could be *unobservable* characteristics that the matching procedure cannot consider that could influence outcomes absent the intervention.

Furthermore, important observable differences do remain between the outreach group and the comparison group, even after matching. Students selected for outreach were first-generation *and* low-income, while control students were either first-generation *or* low-income. The outreach team selected students in this manner to reach as many students as possible, specifically students who would likely benefit from this sort of outreach. Although this was a wise approach for targeting students in need of support, it created some analytic obstacles for our impact analysis that cannot be remedied through matching. In particular, in any matched pair of low-income students, the treatment group student is first-generation and the control group student is not, and similarly, in any matched pair of first-generation students, the treatment group student is low-income and the control group student is not.

Students in the outreach group have two factors that – due to an array of systemic issues – may disadvantage them in terms of college enrollment. Therefore, we may expect that – even absent the intervention – students in the outreach group likely have lower college enrollment rates than those who did not receive the outreach. Given how the outreach groups were defined, we cannot observe college enrollment outcomes for control students who were first-generation *and* low-income, which would be the natural comparison for our outreach group. This is because, for the class of 2020, the intervention was targeted to *all* students who met these two criteria. After presenting our findings, we discuss two different strategies we used to investigate what the differences between these two groups might have been absent the chatbot outreach. Overall, we find that students who met both factors had somewhat lower enrollment rates than those who exhibit only

one of the two factors.

3.6 Results

We primarily focus on college application submission and enrollment in the fall term after students graduate high school as our primary outcome of interest. Additionally, we analyzed whether effects varied according to student characteristics and engagement in chatbot communication. We specifically look at impacts for racially marginalized students, students who opted out of the chatbot outreach, and students who had a high level of engagement with the chatbot.

3.6.1 College Application Submission

Table 3.2 shows the impact of the outreach on submitting at least one college application via the Common App. Students targeted for outreach had somewhat lower application submission rates than their matched controls. 86.1% of the control group submitted at least one college application, while 84.7% of outreach students did so. This differential is equal to around -1.5 percentage points and is statistically significant.

Next, we analyzed whether impacts varied based on whether a student submitted at least one college application before or after the start of the intervention. The outreach had no impact on submitting at least one application prior to the start of the intervention, as expected. However, outreach students were less likely to submit an application after the start of the intervention (-1.3 percentage point difference).

3.6.2 Overall College Enrollment

Next, we present the effects on college enrollment outcomes (Table 3.3). Students in the outreach group had somewhat lower fall 2020 enrollment rates compared to the control group. 76.7% of students in the outreach group enrolled in college in fall 2020, compared to 78.7% of control students, a 2 percentage point differential. We observed effects of a similar magnitude across all other terms.

Additionally, we explored if the outreach impacted whether a student took a gap semester or year. The outreach did not seem to impact whether students took a gap semester. However, we observed a slight increase in outreach students taking a gap year. Oli and the CAC advisors provided no information or

advice about taking time off before enrolling in college, so it seems unlikely that the outreach influenced these choices given the underlying uncontrolled differences between treatment and control groups.

We further unpacked enrollment impacts by subsetting our sample to only students who had submitted no college applications prior to the start of the intervention. Given the timing of the outreach in late spring, these students missed most of the college application deadlines and therefore could benefit from this kind of outreach. However, the outreach had no impact across all enrollment outcomes (Table 3.4).

3.6.3 Fall 2020 College Enrollment

The Common App and its partners were primarily interested in immediate college enrollment after students completed high school. Therefore, we analyzed the impact of the outreach on fall 2020 enrollment by 4-year versus 2-year institutions, private versus public, and full-time enrollment (Table 3.5). The somewhat lower enrollment we observed for outreach students compared to students who did not receive the outreach was likely driven by students forgoing enrolling in 4-year institutions – specifically 4-year privates (-3.3 percentage point difference) – and enrolling in 2-year institutions (1.9 percentage point difference).The CAC advisors sometimes suggested two-year colleges to students who expressed concerns about college affordability but did not otherwise take any position on choosing between a two-year and a four-year college.

Additionally, outreach students had lower full-time enrollment rates (63.9%) than control students (66.8%).

3.6.4 Racially Marginalized Students

Next, we examined whether the outreach had a differential impact on students who belonged to a racially marginalized group. We defined a student as racially marginalized if a student self-identified as non-white or bi/multi-racial.

In Table 3.6 we present the impact of the outreach on fall 2020 college enrollment outcomes. We observed impacts of a similar scale as the entire sample. Racially marginalized students in the outreach group had relatively lower fall 2020 enrollment. There was a modest negative difference in enrollment in

fall 2020 of around 2.1 percentage points, specifically driven by outreach students not enrolling at 4-year institutions by around -4.4 percentage points, but instead enrolling at 2-year institutions (2.2 percentage point difference).

3.6.5 Outreach Participation

Using data from Mainstay regarding student engagement with the bot, and whether students decided to opt-out of receiving outreach from Oli, we constructed high engagement and opt out measures. Specifically, we defined high engagers as a student in the top 25th percentile of the total number of text messages sent throughout the outreach. We created the opt-out indicator by flagging students who explicitly messaged "STOP" to Oli at any point during the outreach or received a "goodbye" message from Oli, indicating that the student had requested to opt-out .[1]

3.6.6 Opt-Out

Throughout the outreach campaign, students had the option to opt-out of receiving outreach by directly messaging Oli. Around 16% of students requested to opt out. In Table 3.7 we examine to what extent impacts varied among students who opted out. Across all outcomes, we see no effect on college enrollment. This finding is not surprising, given that most of the students who opted out did so in the first few weeks of the intervention and therefore received little to no communication from the chatbot.

3.6.7 Oli Engagement

A final question we explored was whether impacts varied for those who engaged highly with Oli, i.e., those in the top 25th percentile of total messages sent throughout the outreach. Due to the low engagement throughout the intervention, a student who sent more than 9 messages was flagged as a "high" engager, which equaled around 17% of all outreach students.

Table 3.8 presents results for students with high engagement throughout the outreach. As opposed to our overall modest negative impacts for the whole sample, here we found a slightly positive impact on fall

[1]Due to data quality issues, we received text message data for 70% of the outreach group. Furthermore, engagement across the intervention was relatively low. On average, students sent around 8 messages throughout the 38 weeks of the outreach.

2020 enrollment for highly engaged students, around a 3 percentage point improvement over the matched controls. Students who engaged with Oli at a high rate were, on average, more likely female, came from a racially marginalized group, and had slightly lower SAT/ACT performance than students who didn't have high engagement with the chatbot. Furthermore, these students also had higher rates of college application submission than non-highly engaged students.

3.7 Results in Context

We want to underscore that important, observable differences remain between the outreach group and the comparison group – even after matching. As mentioned previously, the students selected for outreach were first-generation *and* low-income, while the control students were first-generation *or* low-income. This presented analytic obstacles in estimating outreach impacts.

In this section, we outline two different strategies we took to investigate what differences between these two groups might have been absent the chatbot outreach.

3.7.1 High School Longitudinal Study of 2009

First, we used the High School Longitudinal Study of 2009 (HSLS-09), a nationally representative sample of 9th graders in 2009 who were observed through 2016. The HSLS-09 dataset includes information on college-going. We subsetted the HSLS-09 data to a sample of students who had indicated college interest in 9th grade and had submitted at least one college application. This allowed us to mimic – although imperfectly – our 2020 Common App cohort of students who were college intending.

Next, we calculated enrollment differentials for first-generation *and* low-income students and first-generation *or* low-income students. In our 2020 cohort, low-income was defined as qualifying for a Common App fee waiver. Given the information available in the HSLS-09, we defined low-income as a student's family income falling below 130% of the poverty line (i.e., qualified for free or reduced lunch).

Overall, we found that students who were first-generation and low-income had lower college enrollment rates than students with only one of those characteristics. Students who were first-generation *and* low-income had a college enrollment rate of 79.1%. In comparison, students who exhibited only one of these factors had an enrollment rate of around 84.2% (a -5.1 percentage point difference).

Drawbacks of relying on the HSLS-09 as a source for informing this differential include its age (it observes a cohort of students who completed high school several years prior to 2020) and differences in how we define the subsample of students, relative to the focal groups of interest in our analysis of the Common App data.Therefore, we turn to another data source to generate an additional comparison.

3.7.2 Common App 2021 Cohort

During the 2021 school year, Common App conducted an unrelated randomized controlled study (RCT) with a cohort of students similar to our 2020 sample. We were able to capitalize on the sample employed in this RCT to further investigate enrollment differentials. We worked with Common App to receive high-level descriptives for the cohort of students who did not receive the 2021 intervention. Specifically, we examined fall 2021 enrollment overall and disaggregated by 4-year versus 2-year institutions for students in the 2021 study control group. These differentials allowed us to observe enrollment outcomes for first-generation *and* low-income students and first-generation *or* low-income students from the class of 2021. Unlike the HSLS-09 dataset, the 2021 Common App dataset allows us to define low-income similarly to our 2020 cohort (i.e., qualified for a Common App fee waiver).

Overall, we found that students who met both factors had somewhat lower enrollment rates than those who exhibit only one of the two factors. Students who were both first-generation *and* low-income had a fall 2021 enrollment rate of 72%, while students who only exhibited one of these factors had a 73% enrollment rate (a -1 percentage point difference).

Ideally, we would have matched students in our 2020 sample to demographically similar students in the 2021 sample, allowing us to estimate the group differentials and adjust our impact estimates directly. Unfortunately, this was not possible due to data sharing agreements. However, we were able to reduce the 2021 sample to students who attended high schools in our final matched 2020 sample. We found a similar enrollment differential in fall 2021 of around -1 percentage point.

Based on the differentials of both these data sources, we conclude that – absent any intervention – students who are first-generation *and* low-income have lower college-going rates than those who only hold one of those identities. Therefore, we reason that absent the intervention we would expect that students in the outreach group would have lower college enrollment rates than the comparison group, with differences on the order of -1 to -5 percentage points. These differences suggest that to find a positive effect, the impact of the outreach would have had to be larger than these differentials. Therefore, the modest negative enrollment for outreach students we estimate for the entire sample (-2 percentage points) likely reflect these differentials

– and not – detrimental effects of the chatbot outreach.

3.8 Discussion and Conclusion

The outreach undertaken by Common App and its partners cast a wide net in hopes of helping students through the typically challenging transition to college during the uncertainty of the COVID-19 pandemic. As part of our investigation we investigated whether the outreach improved students' college-going outcomes. The core analytic approach to our analysis relied on comparisons between observationally similar students who nevertheless differed in a critical way. Those in the treatment group were both first-generation *and* from low-income households. Those in the comparison group had one but not both of these characteristics.

Historical evidence suggests that low family income and status as a first-generation college student serve as complementary risk factors for not enrolling in college. On average, high school seniors who are first-generation *and* low-income are less likely to enroll in college than those who are either first-generation *or* low-income but not both.

While the design choice to provide outreach to all students in the graduating class of 2020 directed services to the subgroup of students that were likely most at risk in the early stages of the pandemic, it also adds an additional layer of complexity to the task of evaluating the effect of this outreach. Since students with two risk factors all received outreach, any contemporaneous comparison group will consist of students facing fewer barriers on average to college enrollment.

In the absence of outreach, students in any comparison group can be expected to be more likely to enroll in college than those selected for the intervention. That is, a typical "treatment" vs. "control" group comparison likely leads to bias and an underestimate of the effects of the outreach on college enrollment. The large size of the intervention and the availability of Common App data for students who were not offered outreach enabled us to estimate the difference in enrollment rates for "treatment" vs. "control" groups with standard errors well less than 1 percentage point. Still, large sample sizes cannot compensate on their own for underlying differences in these two groups of students.

We used propensity score matching to create an analysis sample of students who received outreach that is best suited for comparison to a similar group of students who were not offered outreach. More specifically, our matching procedure identified pairs of students who did and did not receive outreach where

each pair attended the same high school and are broadly similar in terms of six demographic variables and four academic indicators. To the degree that first-generation *and* low-income students face greater barriers than first-generation *or* low-income students do, using these variables in the matching procedure should help to reduce the underlying difference in college enrollment rates for treatment and control group students expected in the absence of the intervention. Nevertheless, we still anticipated that within each matched pair, the student who received outreach would face greater barriers to college enrollment and that a comparison of enrollment rates for those in matched pairs would still underestimate the effect of advising.

Despite these challenging issues, it was still quite plausible that our evaluation would find statistically significant evidence that virtual advising increased college enrollment in 2020. One recent multi-district study estimated a 5.2 percentage point increase in four-year college enrollment as the result of text-based FAFSA outreach and support from students' own high school counselors (Avery et al., 2021). An effect of that size could well have been enough to provide a positive and significant result in this study. On the other hand, recent studies of similar outreach and support implemented at scale have found much smaller effects of virtual advising than in-person advising designed to deliver the same kind of support (Avery et al., 2021; Phillips & Reber, 2022). Overall and based on our empirical results, we conclude that the chatbot campaign had no impact on the targeted class of 2020 students submitting college applications or enrolling in college. The limited engagement of students with the bot likely explains these results. While some students were asking the bot important questions about how to enroll in college, decipher financial aid letters, and choose a major, most students did not message Oli in substantively meaningful ways. On average, students sent 8 text messages throughout the 38 week intervention, while 2.5% of students messaged a CAC adviser. A sizable portion of the study population–16%–opted out of the intervention and thus did not receive all of Oli's guidance. However, there was evidence to suggest that the chatbot had a positive impact on college enrollment for students who were highly engaged with the bot during the intervention. This group of highly engaged students may warrant further investigation, as they could shed light on what types of conversations with the bot were most helpful in supporting their postsecondary transition. However, with an intervention that cast such a wide net and that provided relatively general guidance related to college-going processes, it is perhaps unsurprising that the overall effects of the outreach were null.

Taken together, we reasoned that this outreach faced an uphill battle to improve outcomes beyond the differentials that exist absent the intervention. Even though we cannot know what the true causal impact of the outreach was, the evidence we gathered leads us to conclude that the outreach likely neither helped nor harmed students in applying to and enrolling in college.

3.9 Tables and Figures

Table 3.1: Balance of Student Characteristics

			Before Matching			**After Matching**
	Treatment	Control	Std.Mean.Diff	Treatment	Control	Std.Mean.Diff
Age	17.082	17.023	0.118	17.05	17.025	0.050
Male	0.363	0.376	0.029	0.369	0.357	0.024
English spoken at home	0.562	0.558	0.009	0.569	0.573	0.009
Has dependent	0.011	0.007	0.042	0.009	0.009	0.008
Attended > 1 High School	0.181	0.161	0.041	0.164	0.166	0.003
GPA rank	0.643	0.701	0.150	0.698	0.705	0.018
Missing GPA rank	0.238	0.207	0.071	0.178	0.178	0.000
SAT score	671.3	794.868	0.230	712.224	712.861	0.001
Missing SAT Score	0.376	0.317	0.122	0.347	0.347	0.000
College credit exams	0.708	1.135	0.222	0.806	0.848	0.022
TOEFL	0.000	0.001	0.093	0.000	0.000	0.009
Sibling attended college	0.337	0.387	0.105	0.353	0.345	0.017

			Before Matching			After Matching
Submitted one college application before outreach	0.777	0.796	0.044	0.853	0.853	0.000
American Indian/Alaskan Native	0.005	0.003	0.021	0.003	0.003	0.003
Asian	0.084	0.091	0.027	0.096	0.093	0.010
Black	0.28	0.185	0.213	0.287	0.307	0.045
Latinx	0.41	0.231	0.364	0.408	0.396	0.023
Native Hawaiian/Pacific Islander	0.003	0.002	0.011	0.002	0.002	0.011
White	0.169	0.383	0.570	0.164	0.164	0.002
Multi-racial	0.044	0.051	0.031	0.036	0.031	0.025
Race unknown	0.004	0.026	0.328	0.003	0.004	0.006
Non-resident	0.000	0.028	4.261	0.000	0.000	0.042
URM	0.826	0.563	0.694	0.833	0.833	0.000
Num.Obs	142,827	263,299		99,593	61,553	

Table 3.2: Impacts on Submitting at Least One College Application

	Overall	Before outreach	After outreach
Application Differential	-0.015***	0.004	-0.013***
	(0.003)	(0.003)	(0.002)
Control Mean	0.861	0.848	0.036
Num.Obs.	161146	161146	161146
R2	0.318	0.324	0.079
FE: school_id	X	X	X

*$p < 0.05$, ** $p < 0.01$, *** $p < 0.001$ Notes: Analyses include school-level fixed effects and sampling weights to account for our propensity score matching approach with replacement. Baseline covariates include fee waiver indicator, first-generation status, age, gender, English spoken at home, dependent flag, attended more than one high school, high school GPA, SAT/ACT performance, college credit exams count, TOEFL indicator, sibling college indicator, and race/ethnicity indicators. Additionally, we included indicators of missingness for high school GPA and SAT/ACT performance. Robust standard errors clustered at the school level, reported in parentheses.

Table 3.3: Impacts on College Enrollment Outcomes, by Term

	Fall 2020	Spring 2021	Fall 2021	Spring 2022	Gap semester	Gap year
Enrollment Differential	-0.020***	-0.031***	-0.033***	-0.038***	0.003	0.004**
	(0.004)	(0.005)	(0.005)	(0.005)	(0.002)	(0.002)
Control Mean	0.787	0.724	0.701	0.648	0.024	0.023
Num.Obs.	161146	161146	161146	161146	161146	161146
R2	0.155	0.177	0.179	0.194	0.073	0.079
FE: school_id	X	X	X	X	X	X

*$p < 0.05$, ** $p < 0.01$, *** $p < 0.001$ Notes: Analyses include school-level fixed effects and sampling weights to account for our propensity score matching approach with replacement.Baseline covariates include fee waiver indicator, first-generation status,age, gender, English spoken at home, dependent flag, attended more than one high school, high school GPA, SAT/ACT performance, college credit exams count, TOEFL indicator,sibling college indicator, submitted at least one CommonApp application before start of intervention, and race/ethnicity indicators. Additionally, we included indicators of missingness for high school GPA and SAT/ACT performance.The gap semester outcome is a binary indicator coded as 1 if a student did not enroll in Fall 2020 but did enroll in Spring 2021. The gap year outcome is also a binary indicator coded as 1 if a student did not enroll in Fall 2020 or Spring 2021, but did enroll in Fall 2022. Robust standard errors clustered at the school level, reported in parentheses.

Table 3.4: Impacts on College Enrollment Outcomes, by Term - Submitted No Application Prior to Intervention

	Fall 2020	Spring 2021	Fall 2021	Spring 2022	Gap semester	Gap year
Enrollment Differential	0.008	0.030	-0.007	0.016	0.001	0.000
	(0.016)	(0.016)	(0.016)	(0.016)	(0.006)	(0.006)
Control Mean	0.683	0.605	0.593	0.528	0.036	0.041
Num.Obs.	14842	14842	14842	14842	14842	14842
R2	0.311	0.338	0.337	0.344	0.224	0.248
FE: school_id	X	X	X	X	X	X

*$p < 0.05$, ** $p < 0.01$, *** $p < 0.001$ Notes: Analyses include school-level fixed effects and sampling weights to account for our propensity score matching approach with replacement.Baseline covariates include fee waiver indicator, first-generation status,age, gender, English spoken at home, dependent flag, attended more than one high school, high school GPA, SAT/ACT performance, college credit exams count, TOEFL indicator,sibling college indicator, submitted at least one CommonApp application before start of intervention, and race/ethnicity indicators. Additionally, we included indicators of missingness for high school GPA and SAT/ACT performance.The gap semester outcome is a binary indicator coded as 1 if a student did not enroll in Fall 2020 but did enroll in Spring 2021. The gap year outcome is also a binary indicator coded as 1 if a student did not enroll in Fall 2020 or Spring 2021, but did enroll in Fall 2022. Robust standard errors clustered at the school level, reported in parentheses.

Table 3.5: Impacts on Fall 2020 College Enrollment Outcomes

	Fall 2020	4-year	4-year public	4-year private	2-year	Full-time
Enrollment Differential	-0.020***	-0.040***	-0.007	-0.033***	0.019***	-0.029***
	(0.004)	(0.005)	(0.005)	(0.004)	(0.003)	(0.005)
Control Mean	0.787	0.671	0.457	0.217	0.121	0.668
Num.Obs.	161146	161146	161146	161146	161146	161146
R2	0.155	0.207	0.173	0.141	0.145	0.175
FE: school_id	X	X	X	X	X	X

*$p < 0.05$, ** $p < 0.01$, *** $p < 0.001$ Notes: Analyses include school-level fixed effects and sampling weights to account for our propensity score matching approach with replacement. Baseline covariates include fee waiver indicator, first-generation status, age, gender, English spoken at home, dependent flag, attended more than one high school, high school GPA, SAT/ACT performance, college credit exams count, TOEFL indicator, sibling college indicator, submitted at least one CommonApp application before start of intervention, and race/ethnicity indicators. Additionally, we included indicators of missingness for high school GPA and SAT/ACT performance. Robust standard errors clustered at the school level, reported in parentheses.

Table 3.6: Impacts on Fall 2020 College Enrollment Outcomes - Racially Marginalized Students

	Fall 2020	4-year	4-year public	4-year private	2-year	Full-time
Enrollment Differential	-0.021***	-0.044***	-0.007	-0.037***	0.022***	-0.034***
	(0.005)	(0.006)	(0.007)	(0.006)	(0.004)	(0.006)
Control Mean	0.784	0.661	0.452	0.212	0.129	0.661
Num.Obs.	97537	97537	97537	97537	97537	97537
R2	0.160	0.213	0.181	0.144	0.152	0.180
FE: school_id	X	X	X	X	X	X

*$p < 0.05$, ** $p < 0.01$, *** $p < 0.001$ Notes: Analyses include school-level fixed effects and sampling weights to account for our propensity score matching approach with replacement. Baseline covariates include fee waiver indicator, first-generation status, age, gender, English spoken at home, dependent flag, attended more than one high school, high school GPA, SAT/ACT performance, college credit exams count, TOEFL indicator, sibling college indicator, submitted at least one CommonApp application before start of intervention, and race/ethnicity indicators. Additionally, we included indicators of missingness for high school GPA and SAT/ACT performance. Robust standard errors clustered at the school level, reported in parentheses.

Table 3.7: Impacts on College Enrollment Outcomes - Opt Out

	Fall 2020	Spring 2021	Fall 2021	Spring 2022	Gap semester	Gap year
Enrollment Differential	-0.017	-0.016	-0.011	-0.011	-0.001	0.006
	(0.010)	(0.011)	(0.011)	(0.011)	(0.004)	(0.003)
Control Mean	0.787	0.724	0.701	0.651	0.025	0.022
Num.Obs.	20527	20527	20527	20527	20527	20527
R2	0.282	0.295	0.303	0.315	0.212	0.222
FE: school_id	X	X	X	X	X	X

*$p < 0.05$, ** $p < 0.01$, *** $p < 0.001$ Notes: Opt-out was constructed by flagging students who explicitly messaged "STOP" to the bot at any point during the intervention and/or received a "goodbye" message from Oli, indicating that they had requested to opt out. Due to data quality issues we only have text message data for 70% of all treated students. Analyses include school-level fixed effects and sampling weights to account for our propensity score matching approach with replacement.Baseline covariates include fee waiver indicator, first-generation status,age, gender, English spoken at home, dependent flag, attended more than one high school, high school GPA, SAT/ACT performance, college credit exams count, TOEFL indicator,sibling college indicator, submitted at least one CommonApp application before start of intervention, and race/ethnicity indicators. Additionally, we included indicators of missingness for high school GPA and SAT/ACT performance.The gap semester outcome is a binary indicator coded as 1 if a student did not enroll in Fall 2020 but did enroll in Spring 2021. The gap year outcome is also a binary indicator coded as 1 if a student did not enroll in Fall 2020 or Spring 2021, but did enroll in Fall 2022. Robust standard errors clustered at the school level, reported in parentheses.

Table 3.8: Impacts of Text Message Engagement on Fall 2020 College Enrollment - High Text Engagement

	Fall 2020	4-year	4-year public	4-year private	2-year	Full-time
Enrollment Differential	0.026**	0.002	0.032**	-0.032***	0.024***	0.005
	(0.009)	(0.010)	(0.011)	(0.009)	(0.007)	(0.010)
Control Mean	0.793	0.674	0.461	0.215	0.123	0.676
Num.Obs.	26281	26281	26281	26281	26281	26281
R2	0.256	0.304	0.279	0.253	0.251	0.272
FE: school_id	X	X	X	X	X	X

*$p < 0.05$, ** $p < 0.01$, *** $p < 0.001$ Notes: High engagement indicator is coded as 1 for students who are in the top 25th percentile (sent more than 9 text messages) of total number of text messages students sent throughout the intervention. Due to data quality issues we only have text message data for 70% of all treated students. Analyses include school-level fixed effects and sampling weights to account for our propensity score matching approach with replacement.Baseline covariates include fee waiver indicator, first-generation status,age, gender, English spoken at home, dependent flag, attended more than one high school, high school GPA, SAT/ACT performance, college credit exams count, TOEFL indicator,sibling college indicator, submitted at least one CommonApp application before start of intervention, and race/ethnicity indicators. Additionally, we included indicators of missingness for high school GPA and SAT/ACT performance.Robust standard errors clustered at the school level, reported in parentheses.

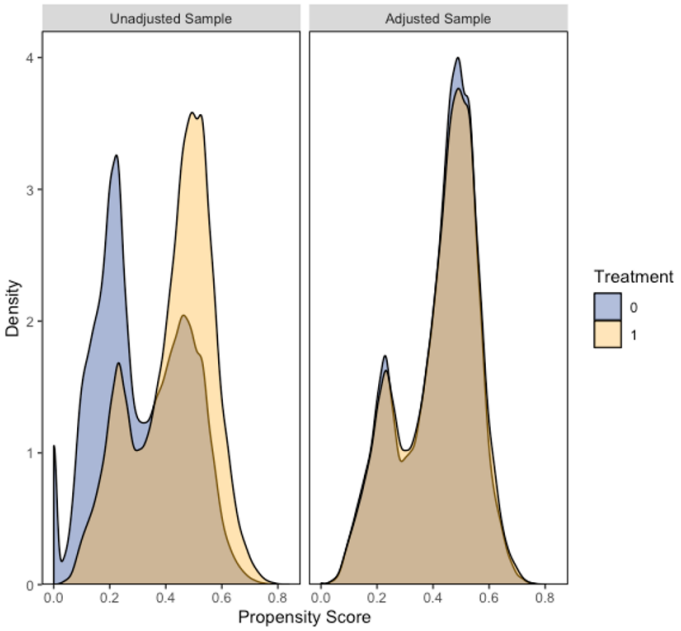


Figure 3.1: Propensity Score Distribution

Appendix A. Simulation R Code

Data Generating Function

```
#############
## WHAT DOES THIS FUNCTION DO?
# The Generate function generates simulated data based on specified conditions.
# The input is a tibble containing information about the sample size, number of
 ↪  covariates, and conditions for the population treatment and outcome models.
# The function first generates correlated normal variables and transforms them into
 ↪  normal, Bernoulli, and uniform variables. It then selects a subset of the
 ↪  covariates for use in the population treatment and outcome models.
# The population treatment and outcome models are generated based on the specified
 ↪  conditions
# and the selected covariates, and the treatment status is also generated.
# The function returns a simulated data tibble that includes the original
 ↪  covariates, treatment status, and generated outcome.
#############


Generate <- function(condition, fixed_objects = NULL) {
  # Makes the tibble of sim crossed conditions accessible to the function, from
   ↪  SimDesign package
```

```r
Attach(condition)


# Generate a mean vector of 0s

mean <- numeric(p)


# Generate a correlation matrix with correlations between -.3 to .3

cor <- matrix(runif(p^2, min = -.3, max = .3), nrow = p)

diag(cor) <- 1


# Smooth the correlation matrix to ensure it is positive definite

cor <- psych::cor.smooth(cor)


# Generate correlated normal variables

vars <- mvrnorm(n, mean, cor)


# Calculate the number of normal, Bernoulli, and uniform variables to generate

num_norm_vars <- floor(p / 2)

num_bern_vars <- floor(p / 4)

num_uniform_vars <- p - num_norm_vars - num_bern_vars


# Convert all variables to uniform variables between 0 and 1

vars_unif <- pnorm(vars)


# Convert the first num_norm_vars variables to normal variable with mean = 0 and
 ↪  sd = 1

vars_normal <- qnorm(vars_unif[, 1:num_norm_vars])
```

```r
# Convert the next num_bern_vars variables to Bernoulli variables with probability
 ↪  of success = 0.5
vars_bern <- qbern(vars_unif[, (num_norm_vars + 1):(num_norm_vars +
↪  num_bern_vars)], 0.5)


# The remainder are left as uniform variables
vars_uniform <- vars_unif[, (num_norm_vars + num_bern_vars + 1):p]


# Combine the transformed variables
vars_transformed <- cbind(vars_normal, vars_uniform, vars_bern)


# Give the columns of vars names v1,v2,etc.
colnames(vars_transformed) <- sprintf("v%d", 1:p)


# Generate variable names and store in the master_covar list
master_covar <- dimnames(vars_transformed)[[2]]


# Create p objects with names v1, v2, etc. in working environment
for (i in 1:p) {
  assign(colnames(vars_transformed)[i], vars_transformed[, i])
}


# Sample half of the covariates and save to covar_confound
covar_confound <- sample(master_covar, size = length(master_covar) / 2)
```

```r
# Sample a quarter of the covariates and save to covar_rel_outcome
covar_rel_outcome <- sample(setdiff(master_covar, covar_confound), size =
↪  length(master_covar) / 4)

# Save the remaining covariates to covar_rel_treatment
covar_rel_treatment <- setdiff(master_covar, union(covar_confound,
↪  covar_rel_outcome))

# Combine covar_confound and covar_rel_outcome, these are the covariates that will
 ↪  be used for the population outcome models
covar_for_treatment <- union(covar_confound, covar_rel_treatment)

# Combine covar_confound and covar_rel_outcome, these are the covariates that will
 ↪  be used for the population outcome models
covar_for_outcome <- union(covar_confound, covar_rel_outcome)

########################################
########################################
# Population treatment models
########################################
########################################

# Generate b coefficients for population treatment models
# Initialize b0 to 0.25
b0 <- 0.25
```

```r
# Create an empty list to store the b coefficients
beta <- vector("list", length(master_covar))

# Loop through all variables in the master covariate list
for (i in seq_len(length(master_covar))) {
  # Generate a random number between -0.4 and 0.4
  x <- runif(1, min = -0.4, max = 0.4)

  # Assign the value to a variable named b1, b2, etc.
  assign(paste0("b", i), x)

  # Store the variable names in the beta list
  b <- paste0("b", i)
  beta[[i]] <- b
}

# Extract the coefficient from the covariate name
b <- sub(".*v", "", covar_for_treatment)

# Create a new variable called element with the format "b * covar_for_treatment"
element <- paste0("b", b, " * ", covar_for_treatment)

########################################
# Population treatment model - Generate base model
########################################
if (scenarioT == "base_T") {
```

```
  # Concatenate the variables from covar_for_treatment into a single string
  equation <- paste0("(1 + exp(-(b0 + ", paste(element, collapse = " + "),
↪ ")))^-1")

  # Evaluate the equation and store the result in trueps
  trueps <- eval(parse(text = equation))
} else

########################################
# Population treatment model - Complex model
########################################
if (scenarioT == "complex_T") {
  # Sample half of the variables from covar_for_treatment
  sample_vars <- sample(covar_for_treatment, length(covar_for_treatment) / 2)

  # Create a list to store the terms
  terms <- list()

  # Iterate over the sampled variables and create the quadratic terms
  for (var in sample_vars) {
    b <- sub(".*v", "", var)
    quad_term <- paste0("b", b, " * ", var, "^2")
    terms[[var]] <- quad_term
  }

  # Sample half of the variables again from covar_for_treatment
```

```
  sample_vars2 <- sample(covar_for_treatment, length(covar_for_treatment) / 2)

  # Create a list of all possible interactions between the variables
  interactions <- combn(sample_vars2, 2, paste0, collapse = "*")

  # Iterate over the interactions and create the interaction terms
  for (inter in interactions) {
    b <- sub(".*v", "", inter)
    inter_term <- paste0("b", b, " * ", inter)
    terms[[inter]] <- inter_term
  }

  # Concatenate all of the terms together and store the result in a new variable
   ↪ called equation
  equation <- paste0("(1 + exp(-(b0 + ", paste(c(unlist(terms), element), collapse
↪ = " + "), ")))^-1")

  # Evaluate the equation
  trueps <- eval(parse(text = equation))
}

########################################
# ~~ binary treatment T
########################################

unif1 <- runif(n, 0, 1)
```

```r
T <- ifelse(unif1 < trueps, 1, 0)


########################################

########################################
# Population outcome models
########################################

########################################


# Generate a coefficients for population outcome models

# Initialize a0 to -0.18
a0 <- -0.18


# Set ATE to 0.3
g <- 0.3


# Generate error terms for population outcome models
e <- rnorm(n, mean = 0, sd = sqrt(0.17))


alpha <- vector("list", length(master_covar))


for (i in 1:length(master_covar)) {
  # Generate a random number between -0.2 and 0.3
  x <- runif(1, min = -0.2, max = 0.3)
  # Assign the value to a1, a2, a3, etc.
  assign(paste0("a", i), x)
  a <- paste0("a", i)
```

```
  alpha[[i + 1]] <- a
}

# Extract the coefficient from the covariate name
a <- sub(".*v", "", covar_for_outcome)

# Create a new variable called element with the format "a * covar_for_outcome"
element <- paste0("a", a, " * ", covar_for_outcome)

########################################
# Population outcome model - Generate base model
########################################
if (scenarioY == "base_Y") {
  equation <- paste0("a0 + g * T", " + ", paste(element, collapse = " + "), " +
↪ e")
  Y <- eval(parse(text = equation))
} else

########################################
# Population outcome model - Complex model
########################################
if (scenarioY == "complex_Y") {
  # Sample half of the variables from covar_for_outcome
  sample_vars <- sample(covar_for_outcome, length(covar_for_outcome) / 2)

  # Create a list to store the terms
```

```
  terms <- list()

  # Iterate over the sampled variables and create the quadratic terms
  for (var in sample_vars) {
    a <- sub(".*v", "", var)
    quad_term <- paste0("a", a, " * ", var, "^2")
    terms[[var]] <- quad_term
  }

  # Sample half of the variables again from covar_for_outcome
  sample_vars2 <- sample(covar_for_outcome, length(covar_for_outcome) / 2)

  # Create a list of all possible interactions between the variables
  interactions <- combn(sample_vars2, 2, paste0, collapse = "*")

  # Iterate over the interactions and create the interaction terms
  for (inter in interactions) {
    a <- sub(".*v", "", inter)
    inter_term <- paste0("a", a, " * ", inter)
    terms[[inter]] <- inter_term
  }

  equation <- paste0("a0 + g * T + ", paste(c(unlist(terms), element), collapse =
↪  " + "), " + e")
  Y <- eval(parse(text = equation))
}
```

```
  ########################################
  # Form simulated data tibble
  ########################################

  v_list <- mget(paste0("v", 1:length(master_covar)))
  dat <- as_tibble(v_list)
  dat$T <- T
  dat$Y <- Y
  dat$trueps <- trueps
  dat
}
```

Analyse Function

```
#############
## WHAT DOES THIS FUNCTION DO?
# The Analyse function is used to estimate the average treatment effect (ATE) and
 ↪  related metrics for a given condition.
# The function uses one of several methods, specified by the method argument, to
 ↪  estimate the propensity score.
# The methods used to estimate the propensity score are
# logistic regression (logit), classification and regression trees (cart), bagging
 ↪  (bag), random forest (forest),
# and three neural network models (nn-1, dnn-2, and dnn-3). Once the propensity
 ↪  score is estimated,
```

```r
# the function uses survey-weighted regression to estimate the ATE, standard error
 ↪  of the ATE, p-value, and 95% confidence interval of the ATE.
# The function also calculates the absolute standardized average mean (ASAM) for
 ↪  each covariate in the data.
#############


# function to estimate the ATE and other metrics
Analyse <- function(condition, dat, fixed_objects = NULL) {
  Attach(condition)


  # if the method is logit, then estimate the ATE using logistic regression
  if (method == "logit") {
    # estimate the propensity score using logistic regression
    mod <- glm(T ~ . - Y - trueps, data = dat, family = binomial(link = "logit"))
    # predict on the entire dataframe to generate ps
    ps <- predict(mod, newdata = dat, type = "response")
    # if the method is cart, then estimate the ATE using classification and
     ↪  regression trees
  } else if (method == "cart") {
    # estimate the propensity score using classification and regression trees
    mod <- rpart(T ~ . - Y - trueps, method = "class", data = dat)
    # predict on the entire dataframe to generate ps
    ps <- predict(mod, newdata = dat, type = "prob")[, 2]
    # if the method is bag, then estimate the ATE using bagging
  } else if (method == "bag") {
    # estimate the propensity score using bagging
```

```
  mod <- bagging(T ~ . - Y - trueps, data = dat, nbagg = 100)
  # save the propensity score to a vector
  ps <- predict(mod, newdata = dat, type = "prob")
  # if the method is forest, then estimate the ATE using random forest
} else if (method == "forest") {
  # estimate the propensity score using random forest
  mod <- randomForest(factor(T) ~ . - Y - trueps, data = dat)
  # save the propensity score to a vector
  ps <- predict(mod, newdata = dat, type = "prob")[, 2]
} else if (method == "nn-1") {
  # Preprocess data
  # Split the data into training and validation sets (80/20)
  split <- sample(2, nrow(dat), replace = TRUE, prob = c(0.8, 0.2)) # random split
↪ of data
  train_data <- dat[split == 1, ]
  validation_data <- dat[split == 2, ]
  x_train <- as.matrix(train_data[, grep("^v", names(train_data))]) # select
↪ columns that start with "v" for input features
  y_train <- as.matrix(train_data[, "T"]) # select column for treatment assignment
  x_validation <- as.matrix(validation_data[, grep("^v", names(validation_data))])
↪ # select columns that start with "v" for input features
  y_validation <- as.matrix(validation_data[, "T"]) # select column for treatment
↪ assignment


  # Define model
  p <- ncol(x_train) # number of input features
```

```r
input_layer <- layer_input(shape = c(p)) # input layer
hidden_layer <- layer_dense(units = p, activation = "relu")(input_layer)
output_layer <- layer_dense(units = 1, activation = "sigmoid")(hidden_layer)
model <- keras_model(inputs = input_layer, outputs = output_layer)

# Compile model
model %>% compile(
  optimizer = "adam",
  loss = "binary_crossentropy",
  metrics = c("accuracy")
)

# Define callbacks
early_stopping <- callback_early_stopping(monitor = "val_loss", min_delta =
↪ 0.001, patience = 5)

# Fit model
history <- model %>% fit(
  x_train,
  y_train,
  epochs = 100,
  batch_size = 64,
  validation_data = list(x_validation, y_validation),
  callbacks = list(early_stopping),
  verbose = 0
)
```

```
  # Preprocess data
  x <- as.matrix(dat[, grep("^v", names(dat))]) # select columns that start with
↪ "v" for input features

  # Predict propensity scores on entire dataset
  ps <- model %>% predict(x)
  ps <- ps[, 1]
 } else if (method == "dnn-2") {
  # Preprocess data
  # Split the data into training and validation sets (80/20)
  split <- sample(2, nrow(dat), replace = TRUE, prob = c(0.8, 0.2)) # random split
↪ of data
  train_data <- dat[split == 1, ]
  validation_data <- dat[split == 2, ]
  x_train <- as.matrix(train_data[, grep("^v", names(train_data))]) # select
↪ columns that start with "v" for input features
  y_train <- as.matrix(train_data[, "T"]) # select column for treatment assignment
  x_validation <- as.matrix(validation_data[, grep("^v", names(validation_data))])
↪ # select columns that start with "v" for input features
  y_validation <- as.matrix(validation_data[, "T"]) # select column for treatment
↪ assignment

  # Define model
  p <- ncol(x_train) # number of input features
  input_layer <- layer_input(shape = c(p)) # input layer
```

```
hidden_layer1 <- layer_dense(units = ceiling(2 * p / 3), activation = "relu",
↪ kernel_regularizer = regularizer_l2(l = 0.01))(input_layer) # first hidden layer
hidden_layer2 <- layer_dense(units = ceiling(2 * p / 3), activation = "relu",
↪ kernel_regularizer = regularizer_l2(l = 0.01))(hidden_layer1) # second hidden
↪ layer
output_layer <- layer_dense(units = 1, activation = "sigmoid",
↪ kernel_regularizer = regularizer_l2(l = 0.01))(hidden_layer2) # output layer
model <- keras_model(inputs = input_layer, outputs = output_layer)

# Compile model
model %>% compile(
  optimizer = "adam",
  loss = "binary_crossentropy",
  metrics = c("accuracy")
)

# Define callbacks
early_stopping <- callback_early_stopping(monitor = "val_loss", min_delta =
↪ 0.001, patience = 5)

# Fit model
history <- model %>% fit(
  x_train,
  y_train,
  epochs = 100,
  batch_size = 64,
```

```
    validation_data = list(x_validation, y_validation),
    callbacks = list(early_stopping),
    verbose = 0
  )

  # Preprocess data
  x <- as.matrix(dat[, grep("^v", names(dat))]) # select columns that start with
↪ "v" for input features

  # Predict propensity scores on entire dataset
  ps <- model %>% predict(x)
  ps <- ps[, 1]
} else if (method == "dnn-3") {
  # Preprocess data
  # Split the data into training and validation sets (80/20)
  split <- sample(2, nrow(dat), replace = TRUE, prob = c(0.8, 0.2)) # random split
↪ of data
  train_data <- dat[split == 1, ]
  validation_data <- dat[split == 2, ]
  x_train <- as.matrix(train_data[, grep("^v", names(train_data))]) # select
↪ columns that start with "v" for input features
  y_train <- as.matrix(train_data[, "T"]) # select column for treatment assignment
  x_validation <- as.matrix(validation_data[, grep("^v", names(validation_data))])
↪ # select columns that start with "v" for input features
  y_validation <- as.matrix(validation_data[, "T"]) # select column for treatment
↪ assignment
```

```r
# Define model
p <- ncol(x_train) # number of input features
input_layer <- layer_input(shape = c(p)) # input layer
hidden_layer1 <- layer_dense(units = ceiling(2 * p / 3), activation = "relu",
↪ kernel_regularizer = regularizer_l2(l = 0.01))(input_layer) # first hidden layer
hidden_layer2 <- layer_dense(units = ceiling(2 * p / 3), activation = "relu",
↪ kernel_regularizer = regularizer_l2(l = 0.01))(hidden_layer1) # second hidden
↪ layer
hidden_layer3 <- layer_dense(units = ceiling(2 * p / 3), activation = "relu",
↪ kernel_regularizer = regularizer_l2(l = 0.01))(hidden_layer2) # third hidden
↪ layer
output_layer <- layer_dense(units = 1, activation = "sigmoid",
↪ kernel_regularizer = regularizer_l2(l = 0.01))(hidden_layer3) # output layer
model <- keras_model(inputs = input_layer, outputs = output_layer)

# Compile model
model %>% compile(
  optimizer = "adam",
  loss = "binary_crossentropy",
  metrics = c("accuracy")
)

# Define callbacks
early_stopping <- callback_early_stopping(monitor = "val_loss", min_delta =
↪ 0.001, patience = 5)
```

```r
  # Fit model
  history <- model %>% fit(
    x_train,
    y_train,
    epochs = 100,
    batch_size = 64,
    validation_data = list(x_validation, y_validation),
    callbacks = list(early_stopping),
    verbose = 0
  )

  # Preprocess data
  x <- as.matrix(dat[, grep("^v", names(dat))]) # select columns that start with
↪ "v" for input features

  # Predict propensity scores on entire dataset
  ps <- model %>% predict(x)
  ps <- ps[, 1]
}

##############################
### calculate metrics
##############################

dat <- dat %>%
```

```
  mutate(
    ps_pred = ps,
    ps_weights = case_when(T == 1 ~ 1 / ps, T == 0 ~ 1 / (1 - ps))
  )

true_ATE <- 0.3

# calculate standardized initial bias prior to weighting
Std_In_Bias <- ((mean(dat$Y[dat$T == 1]) - mean(dat$Y[dat$T == 0])) - true_ATE) /
↪  sd(dat$Y[dat$T == 1])
Prob_Treat <- mean(dat$T)

# estimate the true_ATE with the weights
d.w <- svydesign(~0, weights = dat$ps_weights, data = dat)
fit <- svyglm(Y ~ T, design = d.w)

# save the true_ATE and se_true_ATE
ATE <- unname(coef(fit)["T"])
vcov_matrix <- vcov(fit)
ATE_se <- unname(sqrt(vcov_matrix["T", "T"]))

# extract the p-value of T
p_val <- summary(fit)$coefficients["T", "Pr(>|t|)"]

# calculate the 95% coverage
conf_interval <- confint(fit, level = 0.95)["T", ]
```

```r
lower_bound <- conf_interval[1]

upper_bound <- conf_interval[2]


ci_95 <- ifelse(lower_bound < true_ATE && true_ATE < upper_bound, 1, 0)


# calculate the mean of weights

mean_ps_weights <- mean(dat$ps_weights)


###############

# calculate the ASAM for covariates

###############


# subset the data into the treatment and comparison groups

treatment_group <- dat[dat$T == 1, ]

comparison_group <- dat[dat$T == 0, ]


# get the names of the variables that start with "v"

var_names <- names(dat)[grep("^v", names(dat))]


# initialize the ASAM_list vector

ASAM_list <- rep(NA, length(var_names))


# loop through each covariate

for (i in 1:length(var_names)) {

  # get the covariate name

  covariate <- var_names[i]
```

```
# extract the covariate data from the treatment and comparison groups
treatment_data <- treatment_group[[covariate]]
comparison_data <- comparison_group[[covariate]]

# extract the weights from the treatment and comparison groups
treatment_weights <- treatment_group$ps_weights
comparison_weights <- comparison_group$ps_weights

# calculate the means of the treatment and comparison groups
treatment_mean <- weighted.mean(treatment_data, treatment_weights)
comparison_mean <- weighted.mean(comparison_data, comparison_weights)

# calculate the variances of the treatment group
treatment_var <- wtd.var(treatment_data, treatment_weights)

# calculate the standard deviations of the treatment groups
treatment_sd <- sqrt(treatment_var)

# calculate the standardized difference of means
sd_diff <- (treatment_mean - comparison_mean) / treatment_sd

# take the absolute value of the standardized difference of means
abs_sd_diff <- abs(sd_diff)

# save the absolute standardized difference of means in the ASAM_list vector
```

```
    ASAM_list[i] <- abs_sd_diff
  }



  # calculate the mean of the absolute standardized differences of means
  ASAM <- mean(ASAM_list)


  ret <- c(
    Std_In_Bias = Std_In_Bias,
    Prob_Treat = Prob_Treat,
    ATE = ATE,
    ATE_se = ATE_se,
    mean_ps_weights = mean_ps_weights,
    ASAM = ASAM,
    p_val = p_val,
    ci_95 = ci_95
  )
  ret
}
```

Summarize Function

```
# Summarise function
Summarise <- function(condition, results, fixed_objects = NULL) {
  Std_In_Bias <- mean(results$Std_In_Bias)
  Prob_Treat <- mean(results$Prob_Treat)
```

```r
Bias <- bias(results$ATE, parameter = 0.3, type = "bias")
Abs_Per_Bias <- bias(results$ATE, parameter = 0.3, type = "bias", abs = T, percent
↪  = T)
Abs_Per_Rel_Bias <- bias(results$ATE, parameter = 0.3, type = "relative", abs = T,
↪  percent = T)
ATE_se <- mean(results$ATE_se)
MSE <- RMSE(results$ATE, parameter = 0.3, MSE = T)
Power <- EDR(results$p_val, alpha = 0.05)
coverage_95 <- mean(results$ci_95)
mean_ps_weights <- mean(results$mean_ps_weights)
ASAM <- mean(results$ASAM)
# Create a vector of the results
ret <- c(
  Std_In_Bias = Std_In_Bias,
  Prob_Treat = Prob_Treat,
  Bias = Bias,
  Abs_Per_Bias = Abs_Per_Bias,
  Abs_Per_Rel_Bias = Abs_Per_Rel_Bias,
  ATE_se = ATE_se,
  MSE = MSE,
  Power = Power,
  coverage_95 = coverage_95,
  mean_ps_weights = mean_ps_weights,
  ASAM = ASAM
)
# Return the vector
```

```
  ret
}
```

Simulation Driver 1

```
#########################################################################
# Load libraries and source functions
#########################################################################

packages <- c(
  "here",
  "tidyverse",
  "MASS",
  "Rlab",
  "Matrix",
  "psych",
  "Rlab",
  "rpart",
  "ipred",
  "randomForest",
  "nnet",
  "survey",
  "Hmisc",
  "future",
  "furrr",
  "SimDesign",
```

```
  "keras",
  "tensorflow",
  "reticulate"
)

lapply(packages, library, character.only = TRUE)

# sets working directory to root of R project
here()

######### source functions
source(here("code", "01_data_gen_fun.R"))
source(here("code", "02_analyse_fun.R"))
source(here("code", "03_summarize_fun.R"))

##############################################################################
# Generate sim design dataframe
##############################################################################

# fully-crossed simulation experiment
Design <- createDesign(
  n = c(10000),
  p = c(20, 100, 200),
  scenarioT = c("base_T", "complex_T"),
  scenarioY = c("base_Y", "complex_Y"),
  method = c("logit", "cart", "bag", "forest")
```

```
)

##############################################################################
# Run Simulation
##############################################################################

# use_virtualenv("/ihome/xqin/alg223/.virtualenvs/r-reticulate")
# use_condaenv("r-reticulate")

res <- runSimulation(
  design = Design,
  replications = 1000,
  generate = Generate,
  analyse = Analyse,
  summarise = Summarise,
  parallel = T,
  filename = "sim_results_n10000_r1000_P_e.rds",
  save_results = T
)
```

Simulation Driver 2

```
##############################################################################
# Load libraries and source functions
##############################################################################
```

```r
packages <- c(
  "here",
  "tidyverse",
  "MASS",
  "Rlab",
  "Matrix",
  "psych",
  "Rlab",
  "rpart",
  "ipred",
  "randomForest",
  "nnet",
  "survey",
  "Hmisc",
  "future",
  "furrr",
  "SimDesign",
  "keras",
  "tensorflow",
  "reticulate"
)

lapply(packages, library, character.only = TRUE)

# sets working directory to root of R project
here()
```

```r
######### source functions
source(here("code", "01_data_gen_fun.R"))
source(here("code", "02_analyse_fun.R"))
source(here("code", "03_summarize_fun.R"))

#############################################################################
# Generate sim design dataframe
#############################################################################

# fully-crossed simulation experiment
Design <- createDesign(
  n = c(10000),
  p = c(20, 100, 200),
  scenarioT = c("base_T", "complex_T"),
  scenarioY = c("base_Y", "complex_Y"),
  method = c("nn-1", "dnn-2", "dnn-3")
)

#############################################################################
# Run Simulation
#############################################################################

use_virtualenv("/ihome/xqin/alg223/.virtualenvs/r-reticulate")
# use_condaenv("r-reticulate")
```

```r
res <- runSimulation(
  design = Design,
  replications = 1000,
  generate = Generate,
  analyse = Analyse,
  summarise = Summarise,
  parallel = F,
  filename = "sim_results_n10000_r1000_NP.rds",
  save_results = T)
```

Bibliography

Arnold, K. en D., Chewning, A., Castleman, B., & Page, L. (2015). Advisor and Student Experiences of Summer Support for College- intending, Low-income High School Graduates intending, Low-income High School Graduates. *Journal of College Access*, *1*(1). https://scholarworks.wmich.edu/jca/vol1/iss1/3/

Athey, S. (2015). Machine learning and causal inference for policy evaluation. *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 5–6.

Austin, P. C. (2009). Balance diagnostics for comparing the distribution of baseline covariates between treatment groups in propensity-score matched samples. *Statistics in Medicine*, *28*(25), 3083–3107.

Austin, P. C. (2011). An Introduction to Propensity Score Methods for Reducing the Effects of Confounding in Observational Studies. *Multivariate Behavioral Research*, *46*(3), 399–424. https://doi.org/10.1080/00273171.2011.568786

Austin, P. C., Grootendorst, P., & Anderson, G. M. (2007). A comparison of the ability of different propensity score models to balance measured variables between treated and untreated subjects: A monte carlo study. *Statistics in Medicine*, *26*(4), 734–753.

Avery, C., Castleman, B. L., Hurwitz, M., Long, B. T., & Page, L. C. (2021). Digital messaging to improve college enrollment and success. *Economics of Education Review*, *84*, 102170. https://doi.org/10.1016/j.econedurev.2021.102170

Bai, H. (2011). Using Propensity Score Analysis for Making Causal Claims in Research Articles. *Educational*

Psychology Review, *23*(2), 273—278. https://link.springer.com/article/10.1007/s10648-011-9164-9

Barber, P. H., Hayes, T. B., Johnson, T. L., Márquez-Magaña, L., & signatories, 10234. (2020). Systemic racism in higher education. *Science*, *369*(6510), 1440.2–1441. https://doi.org/10.1126/science.abd7140

Bird, K. A., Castleman, B. L., Denning, J. T., Goodman, J., Lamberton, C., & Rosinger, K. O. (2021). Nudging at scale: Experimental evidence from FAFSA completion campaigns. *Journal of Economic Behavior & Organization*, *183*, 105–128. https://doi.org/10.1016/j.jebo.2020.12.022

Bird, K. A., Castleman, B. L., Song, Y., & Yu, R. (2022). Is Big Data Better? LMS Data and Predictive Analytic Performance in Postsecondary Education. *EdWorking Paper*. https://doi.org/10.26300/8xys-ym74

Boger, Z., & Guterman, H. (1997). Knowledge extraction from artificial neural network models. *1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation*, *4*, 3030–3035.

Brookhart, M. A., Schneeweiss, S., Rothman, K. J., Glynn, R. J., Avorn, J., & Stürmer, T. (2006). Variable Selection for Propensity Score Models. *American Journal of Epidemiology*, *163*(12), 1149–1156. https://doi.org/10.1093/aje/kwj149

Buhlmann, P., & Geer, S. van de. (2011). Variable selection with the lasso. In *Statistics for high-dimensional data: Methods, theory and applications* (pp. 183–247). Springer.

Calonico, S., Cattaneo, M. D., & Titiunik, R. (2015). Rdrobust: An r package for robust nonparametric inference in regression-discontinuity designs. *R J.*, *7*(1), 38.

Cannas, M., & Arpino, B. (2019). A comparison of machine learning algorithms and covariate balance measures for propensity score matching and weighting. *Biometrical Journal*, *61*(4), 1049–1072. https://doi.org/10.1002/bimj.201800132

Carrell, S., & Sacerdote, B. (2017). Why Do College-Going Interventions Work? *American Economic*

Journal: Applied Economics, *9*(3), 124–151. https://doi.org/10.1257/app.20150530

Castleman, B. L., Arnold, K., & Wartman, K. L. (2012). Stemming the Tide of Summer Melt: An Experimental Study of the Effects of Post-High School Summer Intervention on Low-Income Students' College Enrollment. *Journal of Research on Educational Effectiveness*, *5*(1), 1–17. https://doi.org/10.1080/19345747.2011.618214

Castleman, B. L., & Meyer, K. E. (2020). Can Text Message Nudges Improve Academic Outcomes in College? Evidence from a West Virginia Initiative. *The Review of Higher Education*, *43*(4), 1125–1165. https://doi.org/10.1353/rhe.2020.0015

Castleman, B. L., & Page, L. C. (2014). A Trickle or a Torrent? Understanding the Extent of Summer "Melt" Among College-Intending High School Graduates. *Social Science Quarterly*, *95*(1), 202–220. https://doi.org/10.1111/ssqu.12032

Castleman, B. L., & Page, L. C. (2015). Summer nudging: Can personalized text messages and peer mentor outreach increase college going among low-income high school graduates? *Journal of Economic Behavior & Organization*, *115*, 144–160. https://doi.org/10.1016/j.jebo.2014.12.008

Castleman, B. L., & Page, L. C. (2016). Freshman Year Financial Aid Nudges: An Experiment to Increase FAFSA Renewal and College Persistence. *Journal of Human Resources*, *51*(2), 389–415. https://doi.org/10.3368/jhr.51.2.0614-6458r

Castleman, B. L., Page, L. C., & Schooley, K. (2014). The Forgotten Summer: Does the Offer of College Counseling After High School Mitigate Summer Melt Among College-Intending, Low-Income High School Graduates? *Journal of Policy Analysis and Management*, *33*(2), 320–344. https://doi.org/10.1002/pam.21743

Cataldi, E. F., Bennet, C. T., & Chen, X. (2018). *First-Generation Students: College Access, Persistence, and Postbachelor's Outcomes*. U.S. Department of Education. https://nces.ed.gov/pubs2018/2018421.pdf

Chollet, F. et al. (2015). *Keras*. GitHub. https://github.com/fchollet/keras

Clearinghouse, N. S. (2021). *High School Benchmarks 2021 - National College Progression Rates*. https://nscresearchcenter.org/wp-content/uploads/2021/_HSBenchmarksCovidReport.pdf

Collier, Z. K., & Leite, W. L. (2021). A Tutorial on Artificial Neural Networks in Propensity Score Analysis. *The Journal of Experimental Education*, *90*(4), 1003–1020. https://doi.org/10.1080/00220973.2020.1854158

Collier, Z. K., Leite, W. L., & Zhang, H. (2021). Estimating propensity scores using neural networks and traditional methods: A comparative simulation study. *Communications in Statistics-Simulation and Computation*, 1–16.

Cui, P., Shen, Z., Li, S., Yao, L., Li, Y., Chu, Z., & Gao, J. (2020). Causal inference meets machine learning. *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 3527–3528.

Denison, D. G., Mallick, B. K., & Smith, A. F. (1998). A bayesian cart algorithm. *Biometrika*, *85*(2), 363–377.

Derry, A., Krzywinski, M., & Altman, N. (2023). Neural networks primer. *Nature Methods*.

Dorie, V., Hill, J., Shalit, U., Scott, M., & Cervone, D. (2019). Automated versus Do-It-Yourself Methods for Causal Inference: Lessons Learned from a Data Analysis Competition. *Statistical Science*, *34*(1), 43—68. https://projecteuclid.org/euclid.ss/1555056030

Dreiseitl, S., & Ohno-Machado, L. (2002). Logistic regression and artificial neural network classification models: a methodology review. *Journal of Biomedical Informatics*, *35*(5), 352—359. http://www.sciencedirect.com/science/article/pii/S1532046403000340

Dynarski, S., Nurshatayeva, A., Page, L., & Scott-Clayton, J. (2022). Addressing Non-Financial Barriers to College Access and Success: Evidence and Policy Implications. *NBER Working Paper Series*. https:

//doi.org/10.3386/w30054

Dynarski, S., Page, L., & Scott-Clayton, J. (2022). College Costs, Financial Aid, and Student Decisions. *National Bureau of Economic Research*. https://doi.org/10.3386/w30275

Efron, B. (2014). Estimation and accuracy after model selection. *Journal of the American Statistical Association*, *109*(507), 991–1007.

Efron, B., & Tibshirani, R. (1986). Bootstrap methods for standard errors, confidence intervals, and other measures of statistical accuracy. *Statistical Science*, 54–75.

Einav, L., & Levin, J. (2014). Economics in the age of big data. *Science*, *346*(6210), 1243089. https://doi.org/10.1126/science.1243089

Fan, X., & Nowell, D. L. (2011). Using propensity score matching in educational research. *Gifted Child Quarterly*, *55*(1), 74—79.

Farrell, M. H., Liang, T., & Misra, S. (2021). Deep Neural Networks for Estimation and Inference. *Econometrica*, *89*(1), 181–213. https://doi.org/10.3982/ecta16901

Garriott, P. O. (2020). A Critical Cultural Wealth Model of First-Generation and Economically Marginalized College Students' Academic and Career Development. *Journal of Career Development*, *47*(1), 80–95. https://doi.org/10.1177/0894845319826266

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT press.

Grimmer, J. (2015). We are all social scientists now: How big data, machine learning, and causal inference work together. *PS: Political Science & Politics*, *48*(1), 80–83.

Guo, S., Fraser, M., & Chen, Q. (2020). Propensity Score Analysis: Recent Debate and Discussion. *Journal of the Society for Social Work and Research*, *11*(3), 463–482. https://doi.org/10.1086/711393

Guo, S., Guo, S., & Fraser, M. W. (2014). *Propensity Score Analysis: Statistical Methods and Applications*

(Vol. 11). SAGE publications.

Guzman-Alvarez, A., & Page, L. C. (2021). Disproportionate Burden: Estimating the Cost of FAFSA Verification for Public Colleges and Universities. *Educational Evaluation and Policy Analysis*, *43*(3), 545–551. https://doi.org/10.3102/01623737211001420

Harper, S. R. (2010). An anti-deficit achievement framework for research on students of color in STEM. *New Directions for Institutional Research*, *2010*(148), 63–74. https://doi.org/10.1002/ir.362

Herbaut, E., & Geven, K. (2020). What works to reduce inequalities in higher education? A systematic review of the (quasi-)experimental literature on outreach and financial aid. *Research in Social Stratification and Mobility*, *65*, 100442. https://doi.org/10.1016/j.rssm.2019.100442

Hernandez, R., Covarrubias, R., Radoff, S., Moya, E., & Mora, Á. J. (2022). An Anti-Deficit Investigation of Resilience Among University Students with Adverse Experiences. *Journal of College Student Retention: Research, Theory & Practice*, 152102512211099. https://doi.org/10.1177/15210251221109950

Hernández-Blanco, A., Herrera-Flores, B., Tomás, D., & Navarro-Colorado, B. (2019). A systematic review of deep learning approaches to educational data mining. *Complexity*, *2019*.

Hill, J., Weiss, C., & Zhai, F. (2011). Challenges With Propensity Score Strategies in a High-Dimensional Setting and a Potential Alternative. *Multivariate Behavioral Research*, *46*(3), 477–513. https://doi.org/10.1080/00273171.2011.570161

Holland, P. W. (1986). Statistics and causal inference. *Journal of the American Statistical Association*, *81*(396), 945–960.

Hoover, E. (2020). How Is Covid-19 Changing Prospective Students' Plans? Here's an Early Look. *How Is Covid-19 Changing Prospective Students' Plans? Here's an Early Look*. https://www.chronicle.com/article/how-is-covid-19-changing-prospective-students-plans-heres-an-early-look/

Howell, J., Hurwitz, M., Ma, J., Pender, M., Perfetto, G., Wyatt, J., & Young, L. (2021). *College Enrollment*

and Retention in the Era of Covid. The College Board. https://research.collegeboard.org/media/pdf/enrollment-retention-covid2020.pdf

Huber, M., Lechner, M., & Wunsch, C. (2013). The performance of estimators based on the propensity score. *Journal of Econometrics*, *175*(1), 1–21.

Iglesias, L. L., Bellón, P. S., Barrio, A. P. del, Fernández-Miranda, P. M., González, D. R., Vega, J. A., Mandly, A. A. G., & Blanco, J. A. P. (2021). A primer on deep learning and convolutional neural networks for clinicians. *Insights into Imaging*, *12*(1), 117. https://doi.org/10.1186/s13244-021-01052-z

Imbens, G., & Rubin, D. B. (2015). *Causal inference for statistics, social, and biomedical sciences*. Cambridge University Press.

Institute, T. P. (1956). *Higher Education in the United States* (pp. 1158–1158). http://pellinstitute.org/downloads/publications-Indicators/_of/_Higher/_Education/_Equity/_in/_the/_US/_2021/_Historical/_Trend/_Report.pdf

Ives, J., & Castillo-Montoya, M. (2020). First-Generation College Students as Academic Learners: A Systematic Review. *Review of Educational Research*, *90*(2), 139–178. https://doi.org/10.3102/0034654319899707

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning* (Vol. 112). Springer.

Kale, D. C., Che, Z., Bahadori, M. T., Li, W., Liu, Y., & Wetzel, R. (2015). Causal Phenotype Discovery via Deep Networks. *AMIA ... Annual Symposium Proceedings. AMIA Symposium*, *2015*, 677–686.

Karim, M. E., Pang, M., & Platt, R. W. (2018). Can We Train Machine Learning Methods to Outperform the High-dimensional Propensity Score Algorithm? *Epidemiology*, *29*(2), 191—198. https://journals.lww.com/epidem/Fulltext/2018/03000/Can/_We/_Train/_Machine/_Learning/_Methods/_to.5.aspx?casa/_token=ZxQu19dyu-0AAAAA:cVTWRbTk6ZM/_AukNOXHWGjq56uJLoyZm8SY/

_bqlIXCcGB5fSG2Sv1ZeBSBPbf4LazDQqt7QGs1Oy2SYB5dZlZ1o

Keller, B., Kim, J.-S., & Steiner, P. M. (2015). Neural networks for propensity score estimation: Simulation results and recommendations. *Quantitative Psychology Research: The 79th Annual Meeting of the Psychometric Society, Madison, Wisconsin, 2014*, 279–291.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv Preprint arXiv:1412.6980*.

Kuhn, M., Johnson, K., Kuhn, M., & Johnson, K. (2013). Over-fitting and model tuning. *Applied Predictive Modeling*, 61–92.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436—444. 10.1038/nature14539

Lee, B. K. (2023). The central role of the propensity score in epidemiology. *Observational Studies*, *9*(1), 55–57. https://doi.org/10.1353/obs.2023.0004

Lee, B. K., Lessler, J., & Stuart, E. A. (2010). Improving propensity score weighting using machine learning. *Statistics in Medicine*, *29*(3), 337–346. https://doi.org/10.1002/sim.3782

Lee, J., & Little, T. D. (2017). A practical guide to propensity score analysis for applied clinical research. *Behaviour Research and Therapy*, *98*, 76–90. https://doi.org/10.1016/j.brat.2017.01.005

Lee, J., Solomon, M., Stead, T., Kwon, B., & Ganti, L. (2021). Impact of COVID-19 on the mental health of US college students. *BMC Psychology*, *9*(1), 95. https://doi.org/10.1186/s40359-021-00598-3

Li, B., Luo, S., Qin, X., & Pan, L. (2021). Improving GAN with inverse cumulative distribution function for tabular data synthesis. *Neurocomputing*, *456*, 373–383.

Li, L., Shen, C., Wu, A. C., & Li, X. (2011). Propensity score-based sensitivity analysis method for uncontrolled confounding. *American Journal of Epidemiology*, *174*(3), 345–353.

Liaw, A., Wiener, M., et al. (2002). Classification and regression by randomForest. *R News*, *2*(3), 18–22.

Lumley, T. (2020). Package "survey." *Available at the Following Link: Https://Cran. R-Project. Org.*

McCaffrey, D. F., Ridgeway, G., & Morral, A. R. (2004). Propensity Score Estimation With Boosted Regression for Evaluating Causal Effects in Observational Studies. *Psychological Methods*, *9*(4), 403–425. https://doi.org/10.1037/1082-989x.9.4.403

McLewis, C. C. (2021). Higher Education: Handbook of Theory and Research, Volume 36. *Higher Education: Handbook of Theory and Research*, 105–160. https://doi.org/10.1007/978-3-030-44007-7/_6

Molock, S. D., & Parchem, B. (2022). The impact of COVID-19 on college students from communities of color. *Journal of American College Health*, *70*(8), 2399–2405. https://doi.org/10.1080/07448481.2020.1865380

Morris, T. P., White, I. R., & Crowther, M. J. (2018). Using simulation studies to evaluate statistical methods. *Statistics in Medicine*, *38*(11), 2074–2102. https://doi.org/10.1002/sim.8086

Murnane, R. J., & Willett, J. B. (2010). *Methods matter: Improving causal inference in educational and social science research*. Oxford University Press.

Naughton, M. R. (2021). Cracks to Craters: College Advising During COVID-19. *AERA Open*, *7*, 23328584211018715. https://doi.org/10.1177/23328584211018715

Neyman, J. (1923). On the application of probability theory to agricultural experiments. Essay on principles. *Ann. Agricultural Sciences*, 1–51.

Nurshatayeva, A., Page, L. C., White, C. C., & Gehlbach, H. (2021). Are Artificially Intelligent Conversational Chatbots Uniformly Effective in Reducing Summer Melt? Evidence from a Randomized Controlled Trial. *Research in Higher Education*, *62*(3), 392–402. https://doi.org/10.1007/s11162-021-09633-z

Oakes, J., & Rogers, J. (2007). Radical change through radical means: learning power. *Journal of Educational Change*, *8*(3), 193–206. https://doi.org/10.1007/s10833-007-9031-0

Oreopoulos, P. (2020). Promises and Limitations of Nudging in Education. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.3695419

Oreopoulos, P. (2021). What Limits College Success? A Review and Further Analysis of Holzer and Baum's Making College Work. *Journal of Economic Literature*, *59*(2), 546–573. https://doi.org/10.1257/jel.20191614

Ornelas, A., & Solorzano, D. G. (2004). A Critical Race Analysis of Latina/o and African American Advanced Placement Enrollment in Public High Schools. *The High School Journal*, *87*(3), 15–26. https://doi.org/10.1353/hsj.2004.0003

Oromaner, M., & Oakes, J. (1986). Keeping Track: How Schools Structure Inequality. *Contemporary Sociology*, *15*(1), 93. https://doi.org/10.2307/2070941

Page, L. C., Castleman, B. L., & Meyer, K. (2019). Customized Nudging to Improve FAFSA Completion and Income Verification. *Educational Evaluation and Policy Analysis*, *42*(1), 3–21. https://doi.org/10.3102/0162373719876916

Page, L. C., & Gehlbach, H. (2017). How an Artificially Intelligent Virtual Assistant Helps Students Navigate the Road to College. *AERA Open*, *3*(4), 2332858417749220. https://doi.org/10.1177/2332858417749220

Page, L. C., Sacerdote, B. I., Goldrick-Rab, S., & Castleman, B. L. (2022). Financial Aid Nudges: A National Experiment With Informational Interventions. *Educational Evaluation and Policy Analysis*, 016237372211114. https://doi.org/10.3102/01623737221111403

Page, L. C., & Scott-Clayton, J. (2016). Improving college access in the United States: Barriers and policy responses. *Economics of Education Review*, *51*, 4–22. https://doi.org/10.1016/j.econedurev.2016.02.009

Page, L., Lee, J., & Gehlbach, H. (2020). Conditions under which college students can be responsive to nudging. *EdWorkingPaper*. https://doi.org/10.26300/vjfs-kv29

Pan, W., & Bai, H. (2018). Propensity score methods for causal inference: an overview. *Behaviormetrika*,

45(2), 317–334. https://doi.org/10.1007/s41237-018-0058-8

Pang, B., Nijkamp, E., & Wu, Y. N. (2019). Deep Learning With TensorFlow: A Review. *Journal of Educational and Behavioral Statistics*, *45*(2), 227–248. https://doi.org/10.3102/1076998619872761

Peters, A., Hothorn, T., & Hothorn, M. T. (2009). Package "ipred." *R Package*, 2009.

Phillips, M., & Reber, S. (2022). Does virtual advising increase college enrollment? Evidence from a random-assignment college access field experiment. *American Economic Journal: Economic Policy*, *14*(3), 198–234.

Powell, M. G., Hull, D. M., & Beaujean, A. A. (2020). Propensity Score Matching for Education Data: Worked Examples. *The Journal of Experimental Education*, *88*(1), 145–164. https://doi.org/10.1080/00220973.2018.1541850

Reardon, S. F., & Stuart, E. A. (2019). Education Research in a New Data Environment: Special Issue Introduction. *Journal of Research on Educational Effectiveness*, *12*(4), 567–569. https://doi.org/10.1080/19345747.2019.1685339

Reifeis, S. A., & Hudgens, M. G. (2022). On variance of the treatment effect in the treated when estimated by inverse probability weighting. *American Journal of Epidemiology*, *191*(6), 1092–1097.

Richardson, J. T. (2011). Eta squared and partial eta squared as measures of effect size in educational research. *Educational Research Review*, *6*(2), 135–147.

Ripley, B., Venables, B., Bates, D. M., Hornik, K., Gebhardt, A., Firth, D., & Ripley, M. B. (2013). Package "mass." *Cran r*, *538*, 113–120.

Robins, J. M., Hernan, M. A., & Brumback, B. (2000). Marginal structural models and causal inference in epidemiology. *Epidemiology*, 550–560.

Rosenbaum, P. R. (1987). Model-Based Direct Adjustment. *Journal of the American Statistical Association*,

82(398), 387–394. https://doi.org/10.1080/01621459.1987.10478441

Rosenbaum, P. R. (2010). *Design of Observational Studies*. Springer New York. https://doi.org/10.1007/978-1-4419-1213-8

Rosenbaum, P. R., & Rubin, D. B. (1981). The Central Role of the Propensity Score in Observational Studies for Causal Effects. *Biometrika*. https://doi.org/10.21236/ada114514

Rosenbaum, P. R., & Rubin, D. B. (1984). Reducing Bias in Observational Studies Using Subclassification on the Propensity Score. *Journal of the American Statistical Association*, *79*(387), 516. https://doi.org/10.2307/2288398

Rosenbaum, P. R., & Rubin, D. B. (2022). Propensity Scores in the Design of Observational Studies for Causal Effects. *Biometrika*.

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, *65*(6), 386.

Rubin, D. B. (1976). Inference and missing data. *Biometrika*, *63*(3), 581–592.

Rubin, D. B. (2004). Teaching Statistical Inference for Causal Effects in Experiments and Observational Studies. *Journal of Educational and Behavioral Statistics*, *29*(3), 343—367. https://doi.org/10.3102/10769986029003343

Rubin, D. B. (2005). Causal inference using potential outcomes: Design, modeling, decisions. *Journal of the American Statistical Association*, *100*(469), 322–331.

Setoguchi, S., Schneeweiss, S., Brookhart, M. A., Glynn, R. J., & Cook, E. F. (2008). Evaluating uses of data mining techniques in propensity score estimation: a simulation study. *Pharmacoepidemiology and Drug Safety*, *17*(6), 546–555. https://doi.org/10.1002/pds.1555

Song, X., & Coleman, T. S. (2020). Using Administrative Big Data to Solve Problems in Social Science.pdf.

University of Pennsylvania Population Center Working Paper (PSC/PARC). https://repository.upenn.edu/psc/_publications/58/

Soria, K. M., Horgos, B., Chirikov, I., & Jones-White, D. (2020). *First-Generation Students' Experiences During the COVID-19 Pandemic*. Student Experience in the Research University (SERU) Consortium. https://conservancy.umn.edu/handle/11299/214934

Stuart, E. A. (2010). Matching Methods for Causal Inference: A Review and a Look Forward. *Statistical Science*, *25*(1), 1–21. https://doi.org/10.1214/09-sts313

Stuart, E. A. (2023). What is a propensity score? Applications and extensions of balancing score methods. *Observational Studies*, *9*(1), 113–117. https://doi.org/10.1353/obs.2023.0011

Suk, Y., Kang, H., & Kim, J.-S. (2021). Random Forests Approach for Causal Inference with Clustered Observational Data. *Multivariate Behavioral Research*, *56*(6), 829–852. https://doi.org/10.1080/00273171.2020.1808437

Team, R. D. C. (2009). A language and environment for statistical computing. *Http://Www. R-Project. Org*.

Therneau, T., Atkinson, B., Ripley, B., & Ripley, M. B. (2015). Package "rpart." *Available Online: Cran. Ma. Ic. Ac. Uk/Web/Packages/Rpart/Rpart. Pdf (Accessed on 20 April 2016)*.

Weberpals, J., Becker, T., Davies, J., Schmich, F., Rüttinger, D., Theis, F. J., & Bauer-Mehren, A. (2021). Deep Learning-based Propensity Scores for Confounding Control in Comparative Effectiveness Research: A Large-scale, Real-world Data Study. *Epidemiology*, *32*(3), 378–388. https://doi.org/10.1097/ede.0000000000001338

Webster-Clark, M., Stürmer, T., Wang, T., Man, K., Marinac-Dabic, D., Rothman, K. J., Ellis, A. R., Gokhale, M., Lunt, M., Girman, C., et al. (2021). Using propensity scores to estimate effects of treatment initiation decisions: State of the science. *Statistics in Medicine*, *40*(7), 1718–1735.

Westreich, D., Lessler, J., & Funk, M. J. (2010). Propensity score estimation: neural networks, support vector machines, decision trees (CART), and meta-classifiers as alternatives to logistic regression. *Journal of Clinical Epidemiology*, *63*(8), 826—833. http://www.sciencedirect.com/science/article/pii/S0895435610001022

Whata, A., & Chimedza, C. (2022). Evaluating Uses of Deep Learning Methods for Causal Inference. *IEEE Access*, *10*, 2813–2827. https://doi.org/10.1109/access.2021.3140189

Wyss, R., Ellis, A. R., Brookhart, M. A., Girman, C. J., Funk, M. J., LoCasale, R., & Stürmer, T. (2014). The Role of Prediction Modeling in Propensity Score Estimation: An Evaluation of Logistic Regression, bCART, and the Covariate-Balancing Propensity Score. *American Journal of Epidemiology*, *180*(6), 645–655. https://doi.org/10.1093/aje/kwu181

Wyss, R., Schneeweiss, S., Van Der Laan, M., Lendle, S. D., Ju, C., & Franklin, J. M. (2018). Using super learner prediction modeling to improve high-dimensional propensity score estimation. *Epidemiology*, *29*(1), 96–106.