Distribution Agreement

Zijian Wang April 9th, 2025

Minimax Structured Neural Tangent Kernel in Estimating Average Treatment Effect Confounded by Image Covariate

by

Zijian Wang

David Hirshberg, Ph.D.
Advisor

Department of Quantitative Theory and Method

David Hirshberg, Ph.D.
Advisor

Abhishek Ananth, Ph.D.
Committee Member

Kevin McAlister, Ph.D.
Committee Member

2025

Minimax Structured Neural Tangent Kernel in Estimating Average Treatment Effect Confounded by Image Covariate

By

Zijian Wang

David Hirshberg, Ph.D.
Advisor

An abstract of
A thesis submitted to the Faculty of the Emory College of Arts and Sciences
of Emory University in partial fulfillment
of the requirements of the degree of
Bachelor of Science with Honors

Department of Quantitative Theory and Method

2025

Abstract

Minimax Structured Neural Tangent Kernel in Estimating Average Treatment Effect Confounded by Image Covariate
By Zijian Wang

Estimating the average treatment effect (ATE) in observational studies is challenging, particularly when confounding arises from high-dimensional image co-variates. Traditional inverse probability weighting (IPW) methods could fall under the issue of the reliance on knowledge of propensity scores and their high variability in estimation caused by extreme propensity values. Thus, a minimax optimization framework is proposed that minimizes the maximum bias in treatment effect estimation. This thesis aims to propose a *Minimax* structured Neural Tangent Kernel to minimize the maximal of the bias. To simulate real-world conditions where direct patient data is limited, three semi-synthetic data generation frameworks are introduced—ranging from simple image brightness measures to more complex labeling and filtering techniques—to mimic treatment assignments and outcomes based on lung X-ray images. Empirical evaluations using those semi-synthetic data demonstrate that these advanced techniques yield estimates closely aligned with the true ATE, highlighting their promise for robust causal inference in complex, image-driven settings.

Minimax Structured Neural Tangent Kernel in Estimating Average Treatment Effect Confounded by Image Covariate

by

Zijian Wang

David Hirshberg, Ph.D.
Advisor

A thesis submitted to the Faculty of the Emory College of Arts and Sciences
of Emory University in partial fulfillment
of the requirements of the degree of
Bachelor of Science with Honors

Department of Quantitative Theory and Method

2025

Acknowledgments

My thank to Dr. David Hirshberg as my thesis chair and who guided me through the entire project and with more advanced knowledge. Additional thank to Dr. Kevin McAlister and Dr. Abhishek Ananth as my committee members with valuable advice to my project.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

In many observation studies, the goal is to estimate **Average Treatment Effect** (ATE) τ of a binary treatment assignment $W \in \{0, 1\}$. However, some confounding co-variates X, whether observable or unobservable variables that make the treatment correlate with the potential outcomes, presenting in those studies sometimes make our estimation biased. And the issue would be extremely challenging to handle once we encounter image as a high-dimensional confounding.

One standard way of addressing the problem is using **Inverse Probability Weighting** (IPW) that gives the inverse of propensity score, $e(X_i) = P(W_i = 1|X_i)$, for each observation as the balancing weights, which would yield unbiased estimation through such co-variate balancing scheme. As investigated previously by Hirshberg [1], such IPW estimator could be formalized that under the assumption of Ignorability and Overlap, and the ATE can be identified as:

$$\tau_{ATE} = E[Y_i(1) - Y_i(0)] = E[\frac{W_i Y_i}{e(X_i)} - \frac{(1 - W_i)Y_i}{1 - e(X_i)}] \tag{1.1}$$

with the corresponding Inverse Propensity Weights could be defined as:

$$\gamma_{IPW}(X_i) = \frac{W_i}{e(X_i)} - \frac{1 - W_i}{1 - e(X_i)}$$

However, this theoretical guarantee relies on the knowledge of the true propensity score $e(X)$, which is typically unknown in practice and must be estimated from the data. When the estimated propensity score $\hat{e}(X)$ approaches extreme values near 0 or 1, the resulting inverse weights become unstable.

Mathematically, to obtain a better understanding of the estimation error or bias of the IPW estimator, we consider the **conditional mean function** noted as $f(W, X)$, and $f(1, X)$ and $f(0, X)$ correspond to the expected outcomes under treatment and control, respectively. The true Average Treatment Effect (ATE) is then given by

$$\tau = \mathbb{E}[f(1, X) - f(0, X)], \text{ where } f(W, X) = \mathbb{E}[Y|X, W] \tag{1.2}$$

The ultimate goal of inverse propensity weighting is to construct a set of weights $\gamma(X)$ such that the weighted average of observed outcomes $f(W, X)$ approximates the true contrast $f(1, X) - f(0, X)$. That is, we ideally would result in:

$$\mathbb{E}\left[\gamma(X) f(W, X)\right] \approx \mathbb{E}[f(1, X) - f(0, X)] \tag{1.3}$$

However, when the propensity score is estimated and the weights are imperfect, this equality would not hold and introduce bias in our estimation. To quantify such an **imbalance**, we define the worst-case scenario (the maximum of the bias) over **a class of functions $\mathcal{F}$** to express conditional means as:

$$\text{imbalance}_{\mathcal{F}}(\gamma) := \max_{f \in \mathcal{F}} \left| \frac{1}{n} \sum_{i=1}^{n} \gamma_i \, f(W_i, X_i) - \frac{1}{n} \sum_{i=1}^{n} [f(1, X_i) - f(0, X_i)] \right| \tag{1.4}$$

This imbalance formulation reflects the discrepancy between the weighted observed outcomes and the underlying treatment effect contrast, and a possible optimization on the balancing weights that could possibly minimize the maximum of such imbalance would be useful here, which forms the basis of a **Minimax approach** to optimize the balancing weights γ over a function class $\mathcal{F}$ that in maths can be expressed as:

$$\gamma^{\star} = \arg\min_{\gamma} \left\{ \text{imbalance}_{\mathcal{F}}(\gamma) + \frac{\sigma^2}{n^2} \|\gamma\|^2 \right\} \tag{1.5}$$

where σ as a finite bound of the conditional variance $Var[Y|X]$ and sample size n [2].

The key modeling decision lies in the choice of $\mathcal{F}$. In low-dimensional settings, $\mathcal{F}$ may be taken as a set of linear functions or low-order polynomials. But in high-dimensional confounding, especially with images, such simplistic models are insufficient to characterize the complex outcome relationships. In this thesis, we propose to define $\mathcal{F}$ as a function in a reproducing kernel Hilbert space (RKHS), and in particular use the **Neural Tangent Kernel** (NTK) to represent complex functions over image co-variates. The proposed approach would possess much more flexibility without relying on structural information of co-variate from the data generating process. We also aim to examine and compare different statistical approaches to address confounding problem appearing in estimating ATE where we have the freedom of manipulating the function used to estimate the balancing weights γ.

1.1 Empirical Problem

The confounding we would like to address is present in many forms, and we would specifically zoom in a case confounded by image co-variate that brings about more complexity in real-life settings. For instance, in the field of real estate, the ”conceptual construction” picture (including the landscape, material used, overall structure, and neighborhood) would inattentively affect the price and selling outcome, same for hotel online reservation system. In medical research, the image co-variates could be impactful to the decision making in assigning treatment. In respiratory conditions, people with white or hazy shadows appearing in their lung imaging would be recognized as certain pneumonia, which impacting both treatment and outcome; similarly, some orthopedic surgeries would potentially depend on what the X-ray or MRI illustrates and suggests.

Thus, in the empirical stage of examining our approaches, we planned to apply our ideal methodologies to publicly available lung X-ray images data[3]. The treatment assigned to a patient with respiratory disease, and therefore the outcome, might depend on lung imaging. However, due to the limitation of disclosure of data, we are unable to obtain the information from actual patience. So we simulated the treatment and potential outcome data based on the labeling attached to the original lung imaging data with semi-synthetic data generation, that is to make changes building off of the original data with some simulations, which we'll explore further in the following chapter.

1.2 Overview of Chapters

- In Chapter 2, we will formalize our semi-synthetic data generation frameworks to provide a scenario of mimicking both our research setting and real world potential situations.

- In Chapter 3, we will use the standard application of Inverse Probability Weighting (IPW) estimator as baseline estimates and how we can modify the propensity score estimation to achieve a procedure that aligns with the reality closer.

- In Chapter 4, we will explore the proposed *Mini-Max* Approach and start with a simple linear model of estimating the inverse propensity weights, which attempts to achieve a balance of the co-variate for treated and untreated groups through minimizing the maximum of imbalance in co-variates.

- In Chapter 5, we will replace the linear function to be a kernel function that a more proper version of high-dimensional estimation in the our image setting, so that we are able to reformulate the imbalance problem solved with the "Minimax" Approach above and balance a kernelized representation of co-variates.

- In Chapter 6, we will ultimately reach our proposed estimator, a Mini-Max structured Neural Tangent Kernel, in replacing the position of estimating function of balancing weights. Besides applying IPW, we will further estimate with Augmented Inverse Probability Weighting approach as our final estimators once those weights are estimated.

- In Chapter 7, we will summarize a categorization of different estimators mentioned previously as either *Oracle Estimator*, that it uses the information of the co-variates from the data generating process, or *Pixel-Based Estimators*, that relies on no the structural knowledge or assumptions of the co-variates, and make further discussions with respect to our specific setting.

- In Chapter 8, we would utilize the lung imaging data that is publicly available and the semi-synthetic data we've generated. With those data, we are able to examine and evaluate the approaches we've studied in previous chapters and justify their applicability.

- In Chapter 9, a more comprehensive discussion and conclusion will be driven to summarize all the discoveries. Critics and potential future working direction would also be included for continuing refining the study.

Chapter 2

Semi-Synthetic Data Generation

As previously mentioned, due to the limitation of data disclosure, we are unable to acquire the treatment and outcome data from the patients, which could be a common challenge considering the research ethics. Thus, a semi-synthetic data generating framework would be applicable in our setting to obtain the treatment and outcome data.

In this generation process, we have considered three different ways of simulations, from the easiest scenario to the most complex one that will be explained in the following sections.

2.1 Framework 1: A Simple One

In this simple framework, the only co-variate X_i is the average brightness. We obtain the image pixel values in gray-scale as a 224 by 224 matrix, denote as P_{ij}. Then the average brightness, B_i, and itself as our co-variate in this framework, would become:

$$B_i = \frac{1}{224 * 224} \sum_{i,j} P_{ij} = X_i$$

The propensity score, $e(X_i) = E[W = 1|X]$, is obtained using the average brightness, X_i and a scaling factor α:

$$e(X_i) = \frac{1}{1 + e^{-\alpha(X_i - c)}}$$

With the propensity score defined, the treatment assignment follows:

$$W_i \sim Bernoulli(e(X_i))$$

The potential outcome is defined as:

$$Y_i(1) = 1 + e(X_i), \quad Y_i(0) = 0 + e(X_i)$$

where:

- $Y_i(1)$ represents the outcome is the unit would have received given the treatment $W_i = 1$
- $Y_i(0)$ represents the outcome is the unit would not have received given the treatment $W_i = 0$

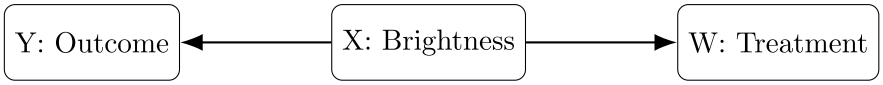
And our observed outcomes Y_i are defined as:

$$Y_i = W_i Y_i(1) + (1 - W_i) Y_i(0)$$

We have set $Y_i(1)$ to be 1 plus propensity score $e(X)$ and $Y_i(0)$ to be 0 plus propensity score $e(X)$ as our counterfactual outcomes in this case. 0 and 1 here would signify a level of recovery where 1 is recovered 0 is not. When defining the outcomes, we bring the confounding co-variate of images, X_i, to the framework by making both treatment assignment and outcome depend on it. This approach is highlighted with its simplicity that the counterfactual average treatment effect (ATE) is simply 1 acquired. Therefore, in the idealized case, we could simplify our analysis and justify the estimations from the base case:

$$ATE = E[Y_i(1) - Y_i(0)] = 1$$

In this framework's definition, the confounding structure is characterized by the following Directed Acyclic Graph:



This structure ensures that our framework has designated to use the average brightness as a confounder that impacts both treatment and outcome.

2.2 Framework 2: Labeling

Provided by the source of data, the labels of the lung imaging reveal the type of pneumonia or normality. With that, we assume that the treatment is *Antibacterial*, and the labels, denoted by L_i, act as an important co-variate influencing both treatment assignment and potential outcomes. This framework refines the previous one by introducing both image brightness and labels as confounding variables, aiming to mimic a more realistic clinical setting.

Let X_i, the co-variate, be defined as $X_i = [L_i, B_i]$, where B_i denote the average brightness of the i-th image and $L_i \in \{\text{NORMAL}, \text{BACTERIA}, \text{VIRUS}\}$ represent its label. The probability of treatment assignment (propensity score) incorporates both brightness and label:

$$\text{logit}(e(X_i)) = \beta_0 + \beta_1 B_i + \beta_2 \cdot I(L_i = \text{BACTERIA}) + \beta_3 \cdot I(L_i = \text{VIRUS})$$

Then, we are able to obtain the propensity score by taking the inverse logit of the right-hand side of the equation above:

$$e(X_i) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 B_i + \beta_2 \cdot I(L_i = \text{BACTERIA}) + \beta_3 \cdot I(L_i = \text{VIRUS}))}}$$

The actual treatment assignment also follows:

$$W_i \sim \text{Bernoulli}(e(X_i))$$

The potential outcomes are designed to reflect the differential impact of antibacterial treatment across the three labels. Specifically, the untreated $Y_i(0)$ and treated $Y_i(1)$ potential outcomes are:

$$Y_i(0) = \theta(L_i) + e(X_i)$$

$$Y_i(1) = Y_i(0) + \tau(L_i) = \theta(L_i) + e(X_i) + \tau(L_i)$$

where $\theta(L_i)$ encodes the baseline outcome based on label:

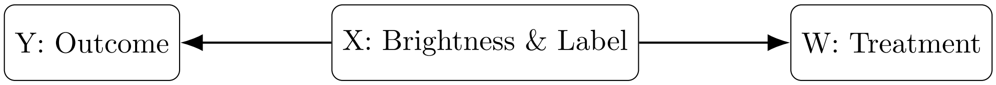
$$\theta(L_i) = \begin{cases} 0, & \text{if } L_i = \text{NORMAL} \\ -1, & \text{if } L_i = \text{BACTERIA or VIRUS} \end{cases}$$

and $\tau(L_i)$ represents the label-specific treatment effect:

$$\tau(L_i) = \begin{cases} 0, & \text{if } L_i = \text{NORMAL} \\ 1, & \text{if } L_i = \text{BACTERIA} \\ -1, & \text{if } L_i = \text{VIRUS} \end{cases}$$

In the normal case, using antibacterial or not as treatment will not be improving or decreasing *level of recovery*, so that we set the effect to be 0 for both treated and controlled cases. For the bacteria-infected lungs, if the person is treated with the anti-bacterial, the treatment will have a positive effect, *level of recovery* would be 1, for this group, while their baseline status are set to be -1. Lastly for the virus-infected lungs, if they are being treated with anti-bacterial mistakenly, the virus in lungs would not be cured by such treatment so that the disease could be worsened to a imaginary -2 of *level of recovery*, demonstrating a negative effect on this group, while the untreated virus-infected people would still stay at their baseline level of -1.

In this framework's definition, the confounding structure is characterized by the following Directed Acyclic Graph:



This structure with heterogeneous outcomes defined complicates the framework to use both brightness and label of each individual together as confounder that impacts both treatment and outcome. We see that this framework is built off of the previous confounding structure and aims to mimic the more complex real-world scenario of treatment-outcome relation by introducing the actual characteristics of the units, the labels, so that treatment would potentially generate various outcome with the extra complexity.

2.3 Framework 3: Image Filtering

In both of the previous frameworks, we only applied average image brightness, X_i, as the confounding co-variate. However, average brightness is vague in describing the whole image if we are using an image resolution of $H \times W = 224 \times 224$. Therefore, a better representation of the image is to apply the *Image Filtering*.

We could first define a filtering matrix as:

$$F = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \tag{2.1}$$

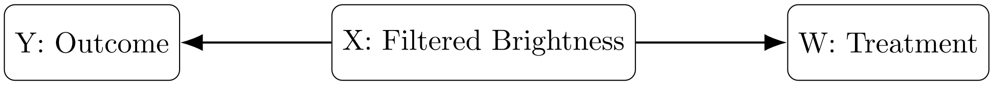
This filter matrix would give high weights to the pixel in the center of each selected 3 by 3 region of interest but low weights to surrounding pixels, which makes it highlight the area of sharp intensity change while suppressing smooth regions. Using this filter matrix, we would obtain each pixel in the filtered image with:

$$I'_{i,j} = \sum_{u=-1}^{1} \sum_{v=-1}^{1} I_{i+u,j+v} \cdot F_{u+2,v+2} \tag{2.2}$$

where I is the original image, I' is the filtered image, and $F_{u+2,v+2}$ is the filter matrix entry $(u+2, v+2)$. After obtaining the pixel values, we could reconstruct the filtered image and aggregate those new pixel values:

$$X_i' = \frac{1}{H' \cdot W'} \sum_{i=1}^{H'} \sum_{j=1}^{W'} (I_{ij}')^2 \tag{2.3}$$

We would then use the *filtered brightness* to pass into our original framework 1 of treatment and outcome assignment. With the change of co-variate, we can formulate this using the following DAG:



It ensures that our new *filtered brightness* as a confounder that impacts both treatment and outcome. This aggregated version of filtering brightness, differing itself from the simple average brightness, could extract the information of edges and textures of the images, which would add complexities in further statistical applications when we apply it to semi-synthetic data generating process. The information encoded in the aggregation would also mimic the emphasize of some particular medical features, like the clearness of lung's boundaries, making the filtered brightness a better input in generating our data given our specific context.

Chapter 3

Inverse Probability Weighting

3.1 IPW I: Standard IPW with True Propensity Scores

Inverse Probability Weighting (IPW) has been widely used as a justifiable way to yield unbiased estimators in the field of causal inference, and its earliest application could be traced back to 1983 when Paul Rosenbaum and Donald Rubin first studied and formalized the concept of propensity score, the conditional probability that each observation assigned to be treated given certain co-variates, in observational studies [4]. Re-weighting each observation by the inverse of its propensity score, $e(X_i) = P(W_i = 1|X_i)$, has been justified to yield unbiased estimation through balancing co-variates, making treated and untreated groups comparable.

Mathematically, our first examined estimator of ATE with the true propensity score, $\hat{\tau}_{IPW1}$ could be formulated as:

$$\hat{\tau}_{IPW1} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{W_iY_i}{e(X_i)} - \frac{(1-W_i)Y_i}{1-e(X_i)}\right) \tag{3.1}$$

where W_i as treatments, Y_i as outcomes, and $e(X_i)$ as the true propensity score of co-variate X_i. Following previous discussion, the true inverse propensity weights used here are defined as:

$$\gamma_{IPW1} = \frac{W_i}{e(X_i)} - \frac{(1 - W_i)}{1 - e(X_i)} \tag{3.2}$$

However, one challenge of this approach is that we generally have no information of the propensity weights, $e(X_i) = P(W_i = 1|X_i)$. Therefore, researchers would choose to estimate $e(X_i)$ through many ways, turning the estimation a two-staged problem: first, estimating $e(X_i)$ and second, computing the IPW estimator. In the following sections, we would explore two ways of estimating the propensity score. One is to use logistic regression to predict the probability of each unit being treated with certain co-variate depending on situations. This approach is commonly used in tackling the challenge of unknown propensity score. The other way is to manipulate the "brightness" by breaking down each image into pixels and to examine which pixel value contributes more to the decision of treatment assignment- a pixel-level weighted brightness. Specifically we fit a *Lasso Regression* and re-weight each pixel by its "contribution". Using the weighted brightness itself as the co-variate or as a part of the co-variate, we could then apply the logistic regression from the second estimator for propensity score estimation.

3.2 IPW II: Logistic Regression Estimation of Propensity Scores

The most direct and commonly used method for estimating propensity scores is logistic regression. In this approach, a logistic model is trained to predict the probability of treatment assignment using certain co-variates.

Given the co-variates X_i, the logistic regression model is specified as:

$$\hat{P}(W_i = 1|X_i) = \frac{e^{(\beta_0 + \beta_1 X_i)}}{1 + e^{(\beta_0 + \beta_1 X_i)}} \tag{3.3}$$

where X_i represents the co-variate, and β_0, β_1 are coefficients learned from the data. Using this estimated probability $\hat{e}(X_i) = \hat{P}(W_i = 1|X_i)$, we then apply the IPW formula to estimate the treatment effect:

$$\hat{\tau}_{IPW2} = \frac{1}{n} \sum_{i=1}^{n} \left(\frac{W_i Y_i}{\hat{e}(X_i)} - \frac{(1 - W_i) Y_i}{1 - \hat{e}(X_i)} \right) \tag{3.4}$$

And the estimated Inverse Propensity Weights are defined by:

$$\gamma_{IPW2} = \frac{W_i}{\hat{e}(X_i)} - \frac{(1 - W_i)}{1 - \hat{e}(X_i)} \tag{3.5}$$

The advantage of logistic regression is its simplicity and interpretability. However, this estimator explicitly assumes a logistic function as the correct function for treatment assignment, making it an *Oracle Estimator*. And because of such assumption, it relies completely on an aggregated information of each image, which is unachievable in reality. This motivates us to explore a more refined estimation strategy to use pixel-level weighting.

3.3 IPW III: Weighted Pixel by Lasso Regression

Considering the previous oracle estimator, we would investigate on an approach that does not rely on the co-variate structural information but is completely based on observed data. Thus, a special detour is to capture the relationship between individual pixel intensities and treatment assignment. We applied a regularized regression approach using Lasso (Least Absolute Shrinkage and Selection Operator). Lasso regression is particularly useful in high-dimensional settings because it performs feature selection by shrinking some unnecessary parameters to zero, identifying the most relevant pixels for predicting treatment assignment with parameters benefit from such a behavior.

Formally, we fit a Lasso regression model where the treatment assignment W_i is predicted using the vector of pixel value $X_{i,j}$:

$$W_i = \alpha + \sum_{j=1}^{p} \beta_j P_{i,j} + \epsilon_i, \text{ with } L_1 \text{ penalty: } \sum_{j=1}^{p} |\beta_j| \leq \lambda \tag{3.6}$$

where p is the number of pixels in each image, and λ is a tuning parameter that controls the degree of regularization.

Once the Lasso model is trained, the learned coefficients $\hat{\beta}_j$ provide a weighting scheme for pixel importance. We then define a weighted brightness measure:

$$B_i^* = \sum_{j=1}^{p} \hat{\beta}_j P_{i,j} \tag{3.7}$$

Finally, either B_i^* itself becomes the whole co-variate X_i^* or in combination with label information L_i as an aggregated co-variate, we use logistic regression on X_i^* to estimate the propensity scores and apply IPW as before:

$$\hat{\tau}_{IPW3} = \frac{1}{n} \sum_{i=1}^{n} \left(\frac{W_i Y_i}{\hat{e}(X_i^*)} - \frac{(1 - W_i) Y_i}{1 - \hat{e}(X_i^*)} \right) \tag{3.8}$$

And the estimated Inverse Propensity Weights using this pixel-reinforced co-variate is:

$$\gamma_{IPW2} = \frac{W_i}{\hat{e}(X_i^*)} - \frac{(1 - W_i)}{1 - \hat{e}(X_i^*)} \tag{3.9}$$

where

$$X_i^* = \begin{cases} B_i^* & \text{if } X_i = B_i \\ [L_i, B_i^*] & \text{if } X_i = [L_i, B_i] \end{cases}$$

The Lasso-based approach offers several advantages over standard logistic regression. First off, it is not an *Oracle Estimator*, as it only relies on the observed co-variate information of the images to work without further assumptions about co-variate structures, which matches closer to the structure of most practical applications. This analysis on the pixel value also makes it more flexible for our IPW estimation.

Chapter 4

A Minimax Approach

As introduced in Chapter 1, the imbalance term arises when the estimated weights γ_i fail to perfectly reweight the observed outcomes $f(W_i, X_i)$ to recover the treatment contrast as a counterfactual term $f(1, X_i) - f(0, X_i)$, particularly in complex or high-dimensional settings like images. Formally, this imbalance can be defined over a class of conditional mean functions $\mathcal{F}$ as Equation 1.4 specified. It captures the worst-case discrepancy between the weighted average of observed conditional expectations and the true treatment contrast. The *minimax* strategy aims to find weights γ that minimize this maximal imbalance (as Equation 1.5 showed). Since there is a great freedom in selecting the function $f \in \mathcal{F}$ to represent the conditional means, we could start from the basic linear relationship in the following elaborations in this chapter.

4.1 Linear Conditional Mean Functions

We begin by assuming that the conditional mean function $f(W, X) \in \mathcal{F}$ is linear in a feature representation of X with some basis function $\psi(X) \in \mathbb{R}^d$ that is used to span the space of the functions f. Specifically, we assume a linearly combined form:

$$f(W_i, X_i) = (1 - W_i)\beta_0^\top \psi(X_i) + W_i \beta_1^\top \psi(X_i) \tag{4.1}$$

where

$$\psi(X_i) = \begin{pmatrix} 1 \\ X_i \\ X_i^2 \end{pmatrix}$$

and $\beta_0, \beta_1 \in \mathbb{R}^d$ are coefficients corresponding to the control and treatment groups, respectively. Similarly, we could evaluate the bias induced by weighting the linear conditional mean function as follows:

$$\begin{aligned} \text{bias} &= \frac{1}{n}\sum_{i=1}^{n}\left[\gamma_i f(W_i, X_i) - (f(1, X_i) - f(0, X_i))\right] \\ &= \frac{1}{n}\sum_{i=1}^{n}\left[\gamma_i((1-W_i)\beta_0^\top \psi(X_i) + W_i\beta_1^\top \psi(X_i)) - (\beta_1 - \beta_0)^\top \psi(X_i)\right] \end{aligned} \tag{4.2}$$

whose value is upper bounded by the worst-scenario imbalance over the designated linear function class $\mathcal{F}$:

$$\text{bias} \leq \text{imbalance}_{\mathcal{F}}(\gamma)$$

This can be further reorganized as a matrix-multiplication format for convenience in calculation:

$$\text{bias} = \left(\frac{1}{n}\sum_{i=1}^{n}\gamma_i \begin{pmatrix} \psi(X_i)(1-W_i) \\ \psi(X_i)W_i \end{pmatrix} - \frac{1}{n}\sum_{i=1}^{n}\begin{pmatrix} -\psi(X_i) \\ \psi(X_i) \end{pmatrix}\right)^\top \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix} \tag{4.3}$$

Applying the Cauchy–Schwarz inequality we could upper bound the bias, which is also the worst-case imbalance, as:

$$\text{bias} \leq \left\| \frac{1}{n}\sum_{i=1}^{n}\gamma_i \begin{pmatrix} \psi(X_i)(1-W_i) \\ \psi(X_i)W_i \end{pmatrix} - \frac{1}{n}\sum_{i=1}^{n}\begin{pmatrix} -\psi(X_i) \\ \psi(X_i) \end{pmatrix} \right\| \cdot \left\| \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix} \right\| \tag{4.4}$$

With the freedom of choosing conditional mean function f, if we assume the norm of the parameter vector is bounded by a parameterized bias budget B (e.g.,

$\|\beta\|_2 = \left\| \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix} \right\|_2 \leq B$) for $B = p^k$ where p is the number of parameters in the model we used to represent the outcome and $k \in [-1, 1]$ as a tuning exponent, then we could write the bias upper bound as:

$$\text{bias} \leq \left\| \frac{1}{n} \sum_{i=1}^{n} \gamma_i \begin{pmatrix} \psi(X_i)(1 - W_i) \\ \psi(X_i) W_i \end{pmatrix} - \frac{1}{n} \sum_{i=1}^{n} \begin{pmatrix} -\psi(X_i) \\ \psi(X_i) \end{pmatrix} \right\| \cdot B = \|A\gamma - b\|_2 \cdot B \quad (4.5)$$

where:

$$A = \frac{1}{n} \begin{pmatrix} \psi(X_1)(1 - W_1) & \cdots & \psi(X_n)(1 - W_n) \\ \psi(X_1) W_1 & \cdots & \psi(X_n) W_n \end{pmatrix}, \quad b = \frac{1}{n} \sum_{i=1}^{n} \begin{pmatrix} -\psi(X_i) \\ \psi(X_i) \end{pmatrix}$$

And minimizing the worst-case squared bias would be equivalent to the following quadratic expression:

$$\min_{\gamma} \text{bias}^2 \leq \min_{\gamma} \|A\gamma - b\|_2^2 \cdot B^2 = \min_{\gamma} B^2 \cdot (\gamma^\top A^\top A \gamma - 2\gamma^\top A^\top b + b^\top b) \quad (4.6)$$

Solving the quadratic form above, we would end up with the following optimized solution:

$$\hat{\gamma} = \arg\min_{\gamma} \|A\gamma - b\|_2^2 \cdot B^2 = (A^\top A)^{-1} A^\top b \quad (4.7)$$

Following the formulation by Hirshberg and Wager [2] and by Hirshberg [1], we can also add the regularization term written as $\lambda = \frac{\sigma^2}{n^2}$ for σ as a finite bound of the conditional variance $Var[Y|X]$ and sample size n. So we can re-write the optimization problem as:

$$\hat{\gamma}_\lambda = \arg\min_{\gamma} \left\{ \|A\gamma - b\|_2^2 \cdot B^2 + \frac{\sigma^2}{n^2} \|\gamma\|^2 \right\} \quad (4.8)$$

Or if we would want to present the imbalance in a normalized way, we could reformulate the expression to be:

$$\hat{\gamma}_\lambda = \arg\min_\gamma \left\{ \|A\gamma - b\|_2^2 + \frac{\sigma^2}{n^2 B^2} \|\gamma\|^2 \right\} = (A^\top A + \frac{\sigma^2}{n^2 B^2} I)^{-1} A^\top b \tag{4.9}$$

where we incorporate the bias budget $B = p^k$ and make our tuning parameter λ as a parameterized function with respect to the budget:

$$\lambda(B) = \frac{\sigma^2}{n^2 B^2} \text{ for } B = p^k \tag{4.10}$$

With those weights calculated, we could then apply it to the observed outcomes for balancing and ATE estimation. This minimax framework in general offers a more flexible solution, where we have the freedom to choose different representations of conditional mean functions f. Also, we could simply tune k to adjust for the penalty term to better adapt to the complexity of the function class, ensuring that as the model becomes more flexible (i.e., as p increases), the regularization is appropriately weakened to achieve a favorable bias-variance tradeoff within the minimax framework and ultimately our estimation result.

Chapter 5

RBF Kernelized Minimax Approach

In the previous chapter, we introduced a minimax approach to inverse probability weighting (IPW), which minimizes the worst-case imbalance across a function class $\mathcal{F}$. And we assumed $\mathcal{F}$ to consist of linear functions over some feature basis functions $\psi(X)$. While this linear formulation yields tractable solutions and insights, it may be hard to capture some complex, nonlinear dependencies.

To address this limitation, we extend the minimax framework to a function in Reproducing Kernel Hilbert Space (RKHS). Specifically, we consider function classes $\mathcal{F}$ in the RKHS defined by a positive semi-definite kernel function $K(x, x')$, and specifically, we focus on the Radial Basis Function (RBF) kernel.

5.1 RBF Kernel and Minimax Approach

Given a kernel function K, the associated RKHS $\mathcal{H}_K$ is the space of all functions f that can be expressed as $f(x) = \sum_i \alpha_i K(x, x')$ for some coefficients $\alpha_j \in \mathbb{R}$. The minimax imbalance in this kernelized method becomes:

$$\text{imbalance}_{\mathcal{H}_K}(\gamma) := \max_{f \in \mathcal{H}_K} \left| \frac{1}{n} \sum_{i=1}^{n} \gamma(X_i)\, f(W_i, X_i) - \frac{1}{n} \sum_{i=1}^{n} [f(1, X_i) - f(0, X_i)] \right| \tag{5.1}$$

We here select to use the *Radial Basis Function (RBF) Kernel* to defined the RKHS by:

$$K(x, x') = \exp\left(-\frac{\|x - x'\|^2}{2\sigma^2} \right) \tag{5.2}$$

where $\sigma > 0$ is the bandwidth parameter that controls the smoothness of the function class. To incorporate treatment assignment W_i into the kernel function, we define an extended feature vector:

$$Z_i = \begin{bmatrix} \alpha W_i \\ \beta X_i \end{bmatrix} \tag{5.3}$$

with scaling constants α, β to balance the influence of treatment status and covariates. We then compute the full kernel matrix K as the replacement of $f(W_i, X_i)$ with each entry at position i, j defined as

$$K_{ij} = \exp\left(-\frac{\|Z_i - Z_j\|^2}{2\sigma^2} \right) \tag{5.4}$$

To construct the treatment group difference in replacing in the second counterfactual term in the minimax framework 5.1, we define K_{diff} as:

$$K_0 = \sum_{j:W_j=0} K(Z_i, Z_j), \quad K_1 = \sum_{j:W_j=1} K(Z_i, Z_j), \quad K_{\text{diff}} = K_1 - K_0 \tag{5.5}$$

The optimal weights γ minimizes the following regularized problem under similar derivation we've shown in the previous chapter:

$$\hat{\gamma} = (K + \lambda(B) \cdot I)^{-1} K_{\text{diff}} \tag{5.6}$$

where $\lambda(B) = \frac{\sigma^2}{n^2 B^2}$ is a regularization parameter to stabilize the solution as we've discussed in the last chapter.

This kernelized minimax formulation retains the core idea of minimizing worst-case imbalance, but generalizes the function class $\mathcal{F}$ from linear to nonlinear spaces. With the RBF kernel, we can flexibly capture smooth variations in the outcome model, making this method robust to complex co-variate structures. This is particularly important when estimating treatment effects is high-dimensional.

Chapter 6

Neural Tangent Kernel Mini-Max

While the RBF kernel offers a powerful, nonlinear generalization of the minimax imbalance framework, it lacks the structural adaptability as an *oracle estimator*. In this chapter, we introduce the Neural Tangent Kernel (NTK) as a tool to extend the minimax weighting approach to deep learning models, which is the ultimate solution we proposed to be in the place of the function f in the minimax framework.

6.1 Neural Network Structure

The NTK used to replace the function defines the kernel as:

$$K_{NTK}(x, x') = \nabla_\theta f(x)^\top \nabla_\theta f(x'), \tag{6.1}$$

where $f(x)$ is the output of a neural network and $\nabla_\theta f(x)$ is the gradient with respect to its parameters.

So, to implement NTK Kernel matrices, we first need to define a simple CNN architecture, whose gradients are important for constructing NTK, for outcome prediction:

$$f(x) = \text{Conv2D} \to \text{ReLU} \to \text{Batch Normalization} \to \text{Pooling} \to \text{Dense} \to \text{Output} \tag{6.2}$$

We separately train two versions of this model: one on treated samples ($W = 1$) and one on control samples ($W = 0$). For each input image, we compute the gradient of individual co-variate$\nabla_\theta f(X_i)$ from the respective model.

6.2 NTK-Based Minimax

Analogous to the RBF case, we define the imbalance as:

$$\text{imbalance}_{\mathcal{H}_{NTK}}(\gamma) := \max_{f \in \mathcal{H}_{NTK}} \left| \frac{1}{n} \sum_{i=1}^{n} \gamma(X_i)\, f(W_i, X_i) - \frac{1}{n} \sum_{i=1}^{n} [f(1, X_i) - f(0, X_i)] \right| \tag{6.3}$$

where $\mathcal{H}_{NTK}$ is the RKHS induced by the NTK, containing all the functions f that could be expressed as linear combinations of the kernel evaluations of the NTK.

Since we would want to measure a treatment-specific counterfactual structure in replacing the original function, we can define our kernel matrix as follows:

$$K_{NTK}\left(\begin{pmatrix} W \\ X \end{pmatrix}, \begin{pmatrix} W' \\ X' \end{pmatrix} \right) = \begin{cases} \langle \nabla_\theta f_1(X), \nabla_\theta f_1(X') \rangle & \text{if } W = W' = 1 \\ \langle \nabla_\theta f_0(X), \nabla_\theta f_0(X') \rangle & \text{if } W = W' = 0 \\ 0 & \text{otherwise} \end{cases} \tag{6.4}$$

Similar to the previous structure in RBF Kernel, we define treatment-specific

NTK-weighted similarities:

$$K_{\text{diff},0}(i,j) = \begin{cases} f_0(X_i)^T f_0(X_j), & \text{if } W_i = 0 \\ 0, & \text{otherwise} \end{cases}$$

$$K_{\text{diff},1}(i,j) = \begin{cases} f_1(X_i)^T f_1(X_j), & \text{if } W_i = 1 \\ 0, & \text{otherwise} \end{cases}$$

$$K_{\text{diff}} = K_{\text{diff},1} - K_{\text{diff},0}$$

Then the optimal balancing weights γ will become:

$$\hat{\gamma} = (K_{\text{NTK}} + \lambda(B) \cdot I)^{-1} K_{\text{diff}} \tag{6.5}$$

where $\lambda(B) = \frac{\sigma^2}{n^2 B^2}$ as the regularization parameter.

The NTK provides a powerful and theoretically grounded framework for extending minimax imbalance to settings where co-variates are image data. It preserves the interpretability of kernel methods while incorporating the expressive capacity of deep neural networks, leading to more informed balancing weights γ. These weights can then be used in IPW or Augmented IPW (AIPW) estimators to reduce bias in the presence of high-dimensional confounding.

6.3 The AIPW Estimator

Having derived balancing weights $\hat{\gamma}_i$ from the Neural Tangent Kernel (NTK) framework and trained group-specific outcome models $\hat{f}_0, \hat{f}_1$, we also invest to test out the *Augmented Inverse Probability Weighting (AIPW)* estimator, which blends both model-based predictions and weight-based corrections into a single coherent estimator. A crucial advantage of the AIPW estimator is its consistency if either the model

$\hat{f}_w(X)$ is correctly specified or the weights $\hat{\gamma}_i$ are valid. This dual pathway to unbiasedness allows researchers to hedge against model misspecification, which makes AIPW particularly useful in high-dimensional or complex settings like image confounding.

The AIPW estimator is defined as:

$$\hat{\tau}_{\text{AIPW}} = \frac{1}{n}\sum_{i=1}^{n}\left\{\hat{f}_1(X_i) - \hat{f}_0(X_i)\right\} + \frac{1}{n}\sum_{i=1}^{n}\hat{\gamma}_i \cdot \left\{Y_i - \hat{f}_{W_i}(X_i)\right\}, \tag{6.6}$$

where:

- $\hat{f}_w(X_i)$: outcome prediction from the treatment-specific CNN model,
- $\hat{\gamma}_i$: NTK-based minimax balancing weights,
- $Y_i - \hat{f}_{W_i}(X_i)$: the residual that quantifies deviation from the model.

The first term is purely model-based. The second term applies bias correction using the weighted residuals, improving robustness even if the outcome model is imperfect.

6.4 Non-Parametric Variance Estimation

The $\hat{\tau}_{\text{AIPW}}$ estimator's variance could be estimated using the following formulation:

$$\hat{V} = \frac{1}{n}\sum_{i=1}^{n}\hat{\gamma}_i^2 \cdot \left(Y_i - \hat{f}_{W_i}(X_i)\right)^2. \tag{6.7}$$

where $\hat{\gamma}(X_i)$ denotes the balancing weight assigned to the i-th observation determined by the non-parametric estimation frameworks, Y_i is the observed outcome for observation i, $\hat{f}(X_i)$ is a consistent estimator of the conditional mean outcome using Neural Network model [1].

A critical advantage of this variance estimate accommodates the non-linear and high-dimensional nature of the data without relying on rigid parametric assumptions.

Across all datasets, the variance estimator facilitates the construction of meaningful confidence intervals, especially when traditional bootstrap methods or sampling distribution may computationally prohibitive. Also, such non-parametric variance would not rely on any other components besides the balancing weights, the observed outcomes, and model predicted outcomes with co-variates. Those advantages further leverage the flexibility of such non-parametric approach in our estimation problems.

6.5 Implementation

In the implementation and empirical testing stage of the estimator, we split the data by half randomly into *training data* and *evaluating data*. The first "training data" is used to train the simple neural network defined above in 6.2; the second half "evaluating data" is to be passed into neural network model for gradient computation and NTK construction. Using this split during implementation would 1) avoid overfitting to the specific data and improve the generalizability of the NTK, 2) reduce bias in estimation by avoid using the same dataset for both training and evaluation (also known as "Double Dipping"), and we will be empirically see how such implementation performs in Chapter 8.

Chapter 7

Oracle and Non-Oracle Estimators

It has been dabbled that some of our estimators mentioned in the previous chapters are *Oracle Estimators*, and others are *Non-Oracle Estimators.* So in this chapter we would classify all estimators introduced into those two categories. This classification is based on whether prior knowledge is assumed about the function class representing the co-variate structure $f(X)$ or the data generating information (e.g. the treatment assignment).

7.1 Oracle Estimators

Oracle Estimators are idealized approaches that assume specific prior knowledge on the class of functions $f(X)$ used to model the conditional mean or propensity score structure from our data generating process. The *Oracle Estimators* from our study include:

- **IPW I: Inverse Probability Weighting with *True* propensity score**

  This estimator assumes exact knowledge of the true propensity score function $e(X)$. It could be serving as an idealized benchmark

- **IPW II: Inverse Probability Weighting with propensity score estimated by logistic regression**

  This estimator is explicitly built on a known summary of co-variate, brightness, with the parametric logistic model as a prior structural information of data.

- **Linear Minimax: Inverse Probability Weighting with balancing weights estimated via Linear functions used in Minimax Structure**

  The linear minimax estimator presumes that the true underlying structure of the conditional means lies within a known linear function class spanned by selected basis functions $\psi(X)$.

- **RBF Minimax: Inverse Probability Weighting with balancing weights estimated via RBF Kernel used in Minimax Structure**

  The use of the Radial Basis Function kernel assumes a RKHS constructed that contains the underlying conditional mean function.

7.2 Non-Oracle Estimators

Non-Oracle Estimators, by contrast, do not rely on explicit knowledge of the appropriate functional form for modeling co-variates or treatment assignment mechanisms. Instead, these estimators utilize flexible models trained directly on raw co-variate inputs. In our study, these include:

- **IPW III: Inverse Probability Weighting with Lasso-reweighted pixel values**

  This estimator directly models the propensity score using the manipulation of observed pixel data from the co-variate via Lasso regression, making no explicit structural assumptions on the functional form of $f(X)$.

- **IPW NTK Minimax: Inverse Probability Weighting with balancing weights estimated via NTK used in Minimax**

  This estimator employs the Neural Tangent Kernel, derived from the gradients of neural networks trained directly on raw pixel data, without explicitly assuming a specific function class.

- **AIPW NTK Minimax: Augmented Inverse Probability Weighting with NTK used in Minimax**

  Similar to IPW NTK Minimax, this estimator does not rely on explicit prior knowledge about the functional form.

We would expect those oracle estimators, as a golden standard, to perform generally better than non-oracle estimators with the assumption of the co-variate structure beforehand, and such a distinction between oracle and non-oracle estimators helps clarify how explicit assumptions about the function class influence estimation bias, variance, and practical applicability in high-dimensional and complex co-variate settings.

Chapter 8

Empirical Performances

In this chapter, we evaluate the empirical performance of different estimation techniques applied to the lung X-ray images. Our assessment focuses on the following methodologies:

1. IPW I
2. IPW II
3. IPW III
4. Linear Minimax
5. RBF Kernel Minimax
6. IPW NTK Minimax
7. AIPW NTK Minimax

In the evaluation stage, Data 1 represents the first and simplest semi-synthetic framework of assigning treatments and potential outcomes; Data 2 represents the second framework where we complicate outcomes by their labels; Data 3 represents the last framework where we applied image filtering to capture more details in pixels instead of directly aggregating mean brightness. Since we applied 3 different data simulation mechanisms, the results are separately presented in Table 8.1 for framework 1, Table 8.2 for framework 2, and Table 8.3 for framework 3. The visualizations of balancing weights estimation, sampling distribution, and tuning process could be found at Appendix A.

8.1 Evaluation Techniques

The entire dataset we obtained consist 5216 images, and we are able to perform random draws to form sampling distribution by changing seeds and examine the coverage. Specifically, in each estimation process, we draw 200 samples from the population image dataset and perform estimation each time to ultimately obtain a sampling distribution. Then we obtain a *normally approximated* coverage using this estimated sampling distribution standard error $\hat{se}$ around the true parameter for better quantitative evaluation. For the estimators that involve a tuning parameter k (refer to the expression 4.9), we primarily evaluate the *bias*, *RMSE*, and *Coverage* by selecting several specific $k \in [-1, 1]$ that controls the regularizing strength and pick the top 2 well-performed candidates.

The visualizations in the Appendix A includes the following components:

- For IPW I, II, and III, it includes the inverse probability weights and sampling distribution
- For Minimax based approaches, it includes 6 plots for each data generating process:
  - 2 preferred choices of balancing weights after tuning the regularization term by k.

- 2 estimated sampling distributions with $\tau \pm 1.96\hat{se}$ interval estimate plotted.
- 1 plot for k selection in the tuning process of bias/rmse and rmse curves plotted with 1-1 corresponding relationship between bias/rmse and coverage.
- a coverage plot for confirmation.

The exact k and λ value selected will be mentioned also in the Appendix B.

8.2 Results

Method	$\hat{\tau}_0$	**Sample** σ	**Coverage**	$E_n[\hat{\tau}_i]$
Truth	1	NA	NA	NA
IPW I	0.863	0.197	0.93	1.043
IPW II	0.898	0.283	0.99	1.065
IPW III	0.765	0.072	0.71	0.893
IPW w/ Linear Mini-Max	0.982	0.017	0.94	0.997
IPW w/ RBF Mini-Max	1.001	0.007	0.93	1.001
IPW w/ NTK Mini-Max	1.070	0.022	0.32	1.052
AIPW w/ NTK Mini-Max	1.002	0.021	0.93	0.999

Table 8.1: Semi-Synthetic Framework 1

Method	$\hat{\tau}_0$	**Sample** σ	**Coverage**	$E_n[\hat{\tau}_i]$
Truth	0.227	NA	NA	NA
IPW I	0.362	0.137	0.95	0.254
IPW II	0.337	0.116	0.96	0.242
IPW III	0.303	0.057	0.68	0.308
IPW w/ Linear Mini-Max	0.273	0.062	0.96	0.230
IPW w/ RBF Mini-Max	0.292	0.074	0.94	0.237
IPW w/ NTK Mini-Max	0.284	0.078	0.93	0.263
AIPW w/ NTK Mini-Max	0.323	0.061	0.43	0.356

Table 8.2: Semi-Synthetic Framework 2

Method	$\hat{\tau}_0$	**Sample** σ	**Coverage**	$E_n[\hat{\tau}_i]$
Truth	1	NA	NA	NA
IPW I	1.006	0.195	0.94	1.008
IPW II	0.997	0.091	0.95	1.023
IPW III	0.812	0.073	0.44	0.846
IPW w/ Linear Mini-Max	0.951	0.051	0.97	0.959
IPW w/ RBF Mini-Max	0.985	0.022	0.95	0.994
IPW w/ NTK Mini-Max	1.072	0.041	0.05	1.149
AIPW w/ NTK Mini-Max	1.123	0.004	0	1.121

Table 8.3: Semi-Synthetic Framework 3

Chapter 9

Discussion and Conclusion

Previously we've explored methodologies to estimate the Average Treatment Effect (ATE) in observational studies complicated by high-dimensional image co-variates. This chapter would summarize key findings and interpret based on the simulation results in Tables 8.1, 8.2, and 8.3 as well as in Visualizations from Appendix A and B.

9.1 Interpretation of Findings

9.1.1 Table Results

Our results from Chapter 8 could reveal several critical insights:

- The *oracle estimators*, including IPW I, II, IPW w/ Linear Minimax, and IPW w/ RBF Mini-Max, generally perform well in three semi-synthetic data frameworks due to the prior knowledge and assumptions about co-variate structures. Among those estimators, IPW w/ Linear Minimax and IPW w/ RBF Minimax are with impressively small bias and variance but still with a decent coverage. Those oracle estimators could be serving as a golden standard of estimation, and their results could also inform us the significance of assumptions about co-variate structure in observational studies before estimation.

- From the tables presented in the last chapter, those *non-oracle estimators*, including IPW III, IPW w/ NTK Minimax, and AIPW w/ NTK Mini-Max, compared to those oracle estimators performs slightly worse due to their pure reliance on the image pixel values during estimation. Generally speaking, IPW III performs well across the three frameworks with a medium level of achievements of coverage, variance, and bias. We do observe a huge fluctuation of performances of our proposed approaches using NTK Minimax framework: the IPW estimator with NTK-based Minimax, compared to other estimators, performs badly in the first (simple binary outcome) and third (filtered image as co-variate) empirical data generating framework by resulting a small coverage, while it outperforms many estimators in the second data (both label and brightness as an aggregated co-variate). The AIPW estimator under the NTK also fluctuates in performance when it obtains the highest coverage score under testing with the first data but gradually decreases in this measure as the data generating mechanism becomes more complex. It ultimately obtains 0 coverage with bias existing but tiny variance.

9.1.2 Visual Results

Plots from appendix could reveal some more inferences:

- *IPW I and II* Those "baseline-kind" estimators, being oracle, predicts the inverse propensity weights aligning with the true weight curves well under the testing of all data, which also results a good coverage ultimately.

- *IPW III* Due to the fact that it is not cheating with assuming co-variate structural information, it outputs some weights that are not aligning well enough with the true curve. But since it captures the general predicted shape, it is able to recover a medium coverage at most times.

- *Linear Minimax* The Linear Minimax generally performs well in rendering weights to capture the essence of the true curves. But due to the limitation in of the model's linear structure, the weights would not describe some of the variation in the true curve well. Despite the imperfectness, it normally results decent coverage and balance between bias and variance in all three data. A special choice for tuning parameter in data 3 ends up with huge variance when we enlarge the weights magnitude, where we could still obtain a good coverage.

- *RBF Minimax* The RBF Minimax, by including more functions it is able to model, adds more variations in predicting balancing weights. With proper regularizing strength, the RBF Minimax normally ends up with good coverage and bias-variance balance.

- *IPW w/ NTK Minimax*: The NTK weights, when used with IPW, could more or less capture the essence of the true weight curve. However, its coverage fluctuates for the best tuning parameter choice, and sometimes we are choosing small weights by regularizing more to obtain a better coverage (in data 2).

- *AIPW w/ NTK Minimax*: The NTK weights' issue becomes more obvious when used in the AIPW estimator. Basically across all three data generating frameworks, the weights are all generally small. And observing the parameter tuning curve, we see that the Bias/RMSE curves are mostly flat to be close to one, indicating that the Bias almost takes up the entire RMSE.

9.2 Discussion

Our proposed estimator, IPW or AIPW using Neural Tangent Kernel, admittedly doesn't perform well as we have expected. One thing to discuss based on the findings is the zero coverage observed from the AIPW based NTK Minimax estimator. Although we applied AIPW estimator due to its doubly robustness in practice, we still encounter an inferior estimation result. This could be indicating that both our balancing weights and our outcome predictive model are not idealized to work well. Nevertheless, we do obtain a good coverage when using small weights by regularizing harshly ($\lambda = 3600$ and $B \approx 0$) in the first data, this is probably because of the well-performed predictive model from our first AIPW term thankful to the doubly robustness of the estimator. However, if the first AIPW term is not predicting the outcomes well, the weights we obtained with different specific tuning parameters would still not make proper corrections for the bias, demonstrated from the results in data 2 and 3 (as the visualization of inverse probability weights of this estimator shown).

When discussing about our uncommon tuning parameters, there is a chance that we missed the actual regularization term that could bring us to an ideal, properly tuned solution. Since some IPW and most AIPW estimators receive small weights, the harshly regularized tiny variation would not render a preferable coverage. A small regularization term might have not been tested in our scenario that could render a better coverage. And we could look into a wider but also finer range of tuning parameters, or use a better parameter tuning algorithm than our simple grid search, to find out that specific sweet spot. But we are still unable to figure out the reason for such a behavior of those regularization terms specifically for the Neural Network based minimax frameworks at the current stage of our study.

All the discussion above could be brought back to our Neural Network Modeling that used for constructing kernel matrices and making predictions. If we could better tune the model or use some existing models, the output from those Neural Network based estimators could be potentially better. Since the schedule of this study is limited to months, there was little time to be spent on tuning the Neural Network Model or using pre-trained models as a better framework for outcome prediction, which could lead to a bad estimate of our final NTK based Mini-Max estimators.

9.3 Future Work

One future direction is to focus on refining the kernel formulations and tuning the Neural Network used in NTK mechanism. Additionally, integrating regularization techniques specifically tailored to high-dimensional image data could help stabilize estimation. Last but not least, we would ultimately extending these methods to real-world, non-synthetic medical datasets one day, where we know the actual treatment assigned and outcomes to eventually test out the performance of those approaches in the real situation.

Appendix A

Plots of Simulation Results

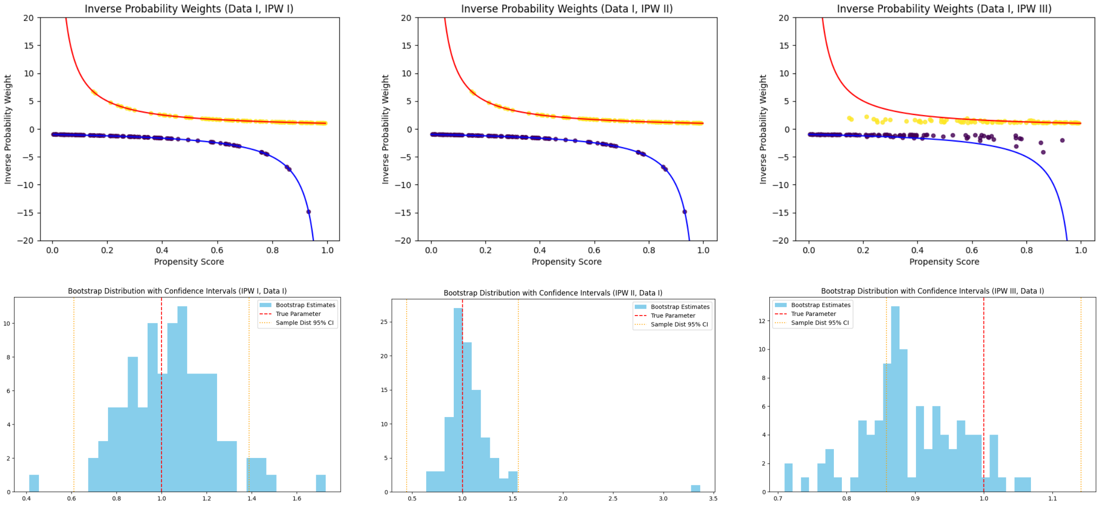


Figure A.1: Semi-Synthetic Framework 1 - IPW I, II, III

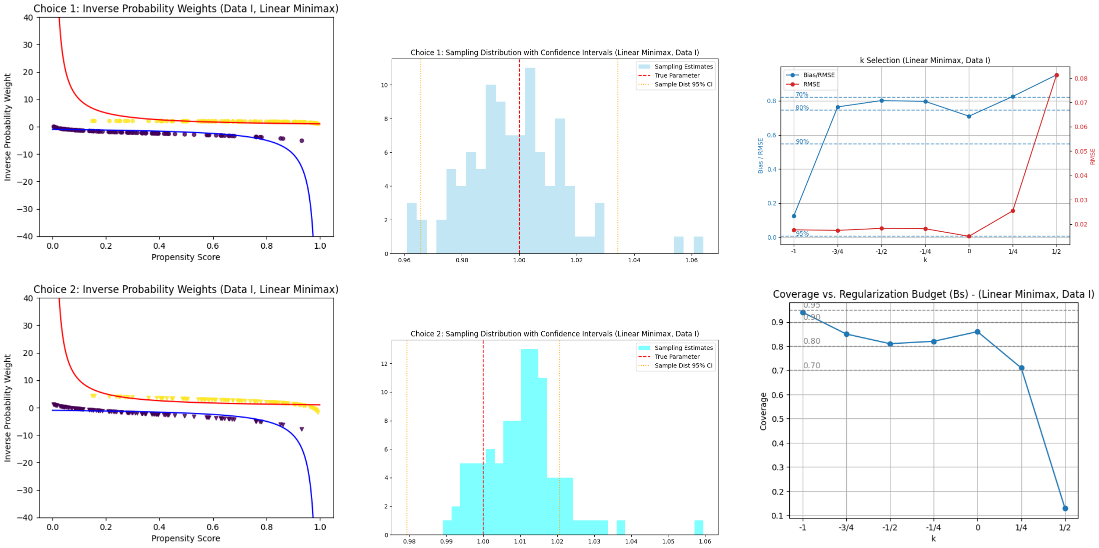


Figure A.2: Semi-Synthetic Framework 1 - IPW w/ Linear Minimax

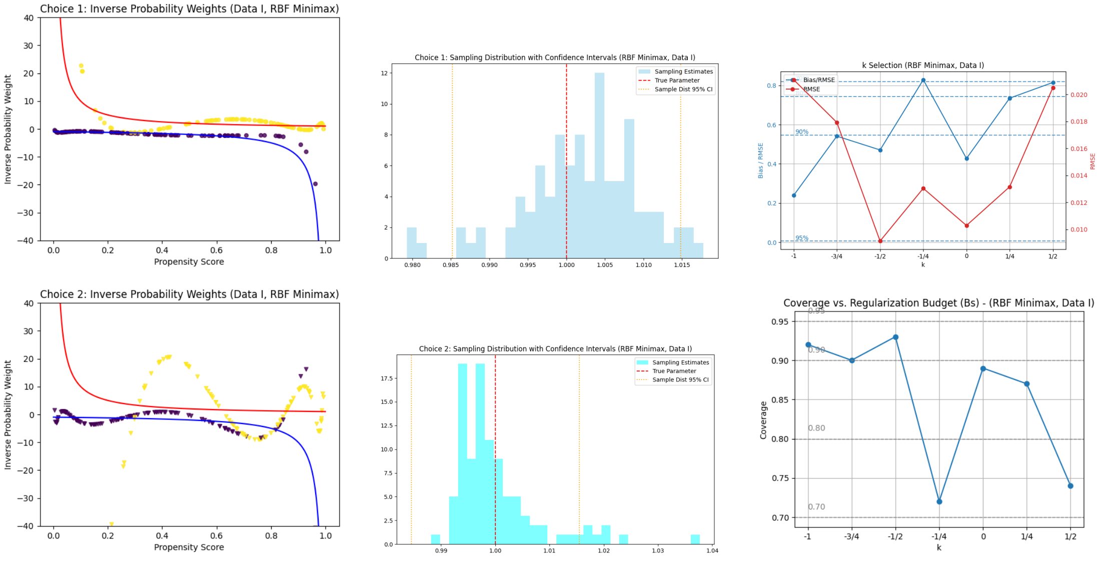


Figure A.3: Semi-Synthetic Framework 1 - IPW w/ RBF Minimax

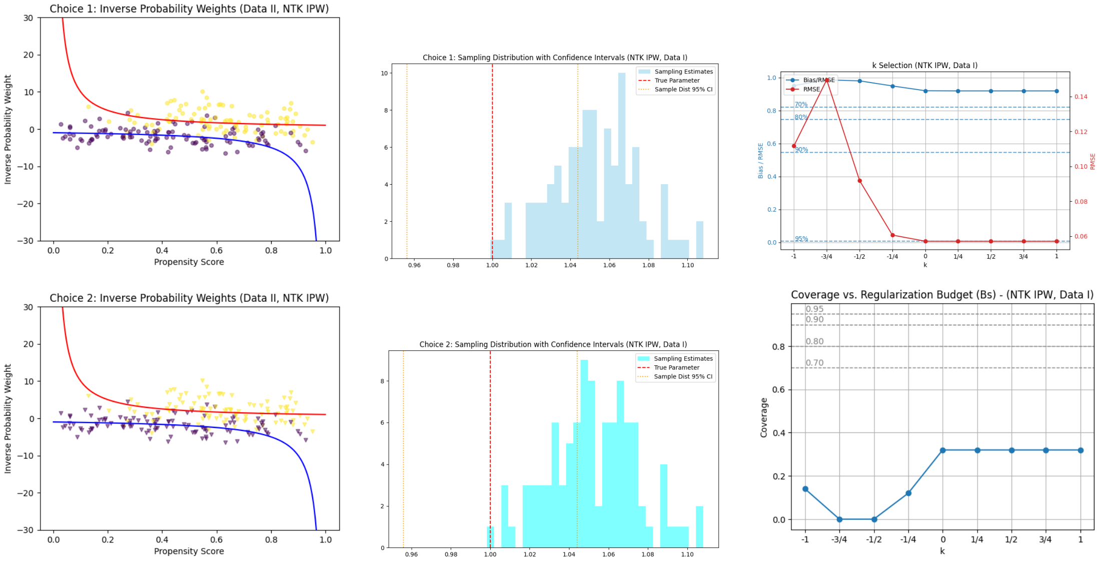


Figure A.4: Semi-Synthetic Framework 1 - IPW w/ NTK Minimax

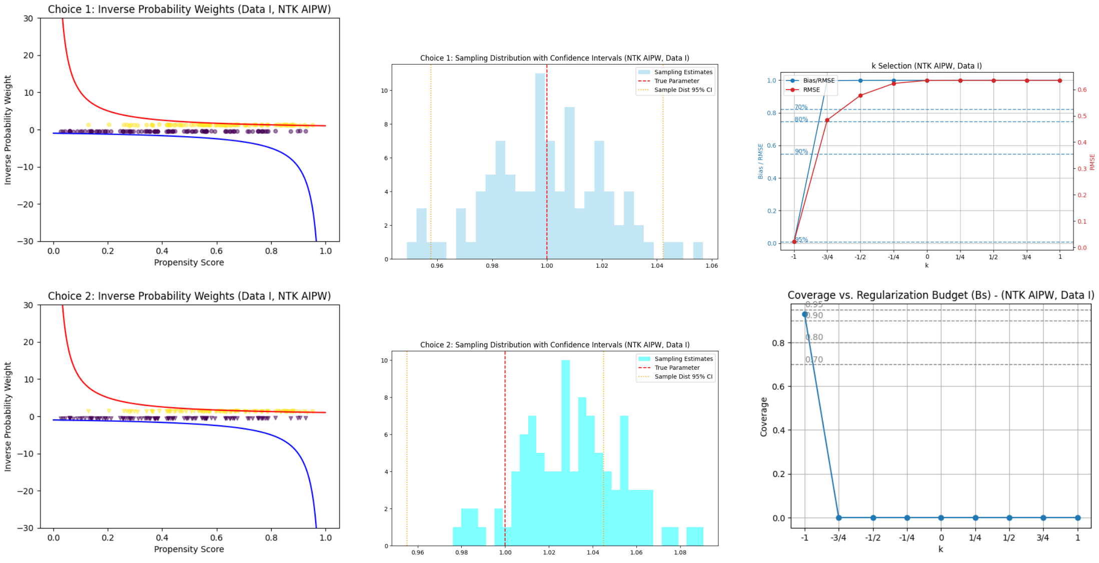


Figure A.5: Semi-Synthetic Framework 1 - AIPW w/ NTK Minimax

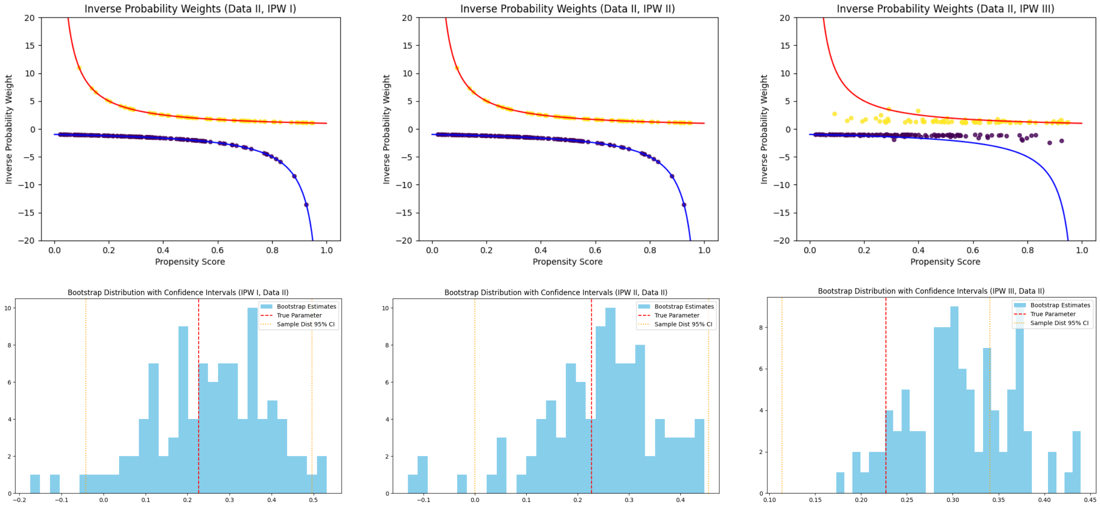


Figure A.6: Semi-Synthetic Framework 2 - IPW I, II, III

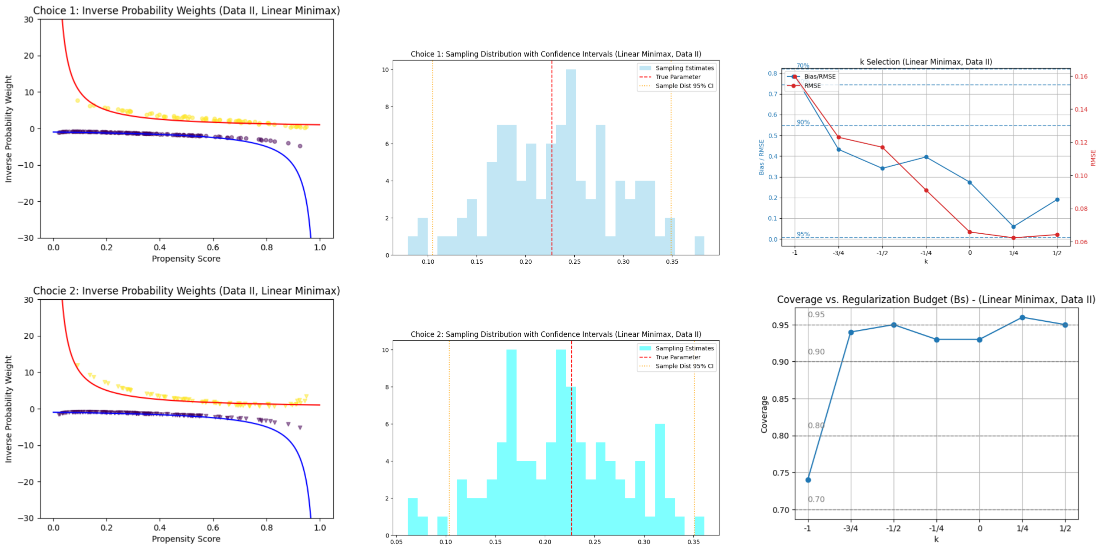


Figure A.7: Semi-Synthetic Framework 2 - IPW w/ Linear Minimax

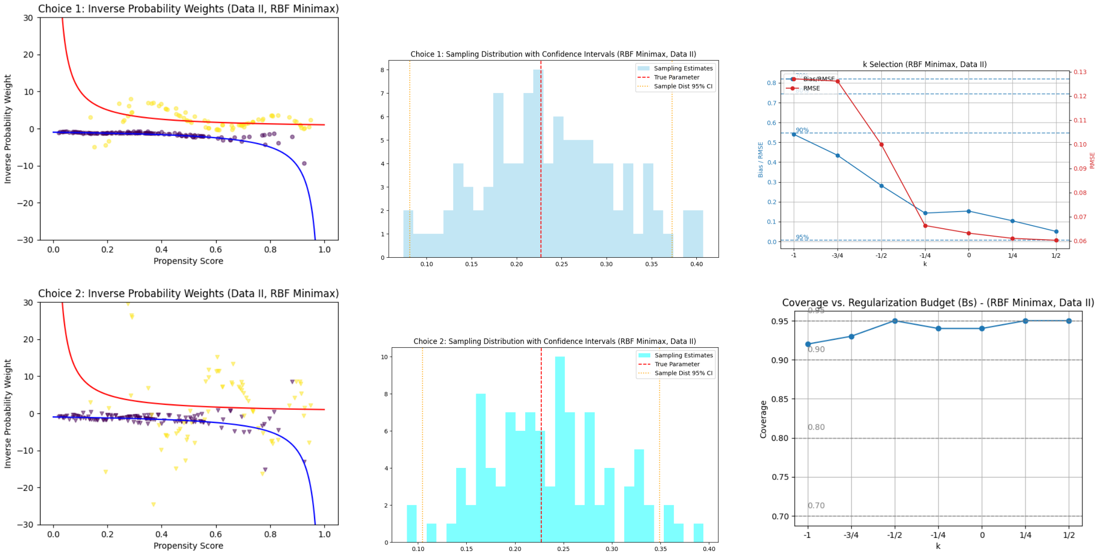


Figure A.8: Semi-Synthetic Framework 2 - IPW w/ RBF Minimax

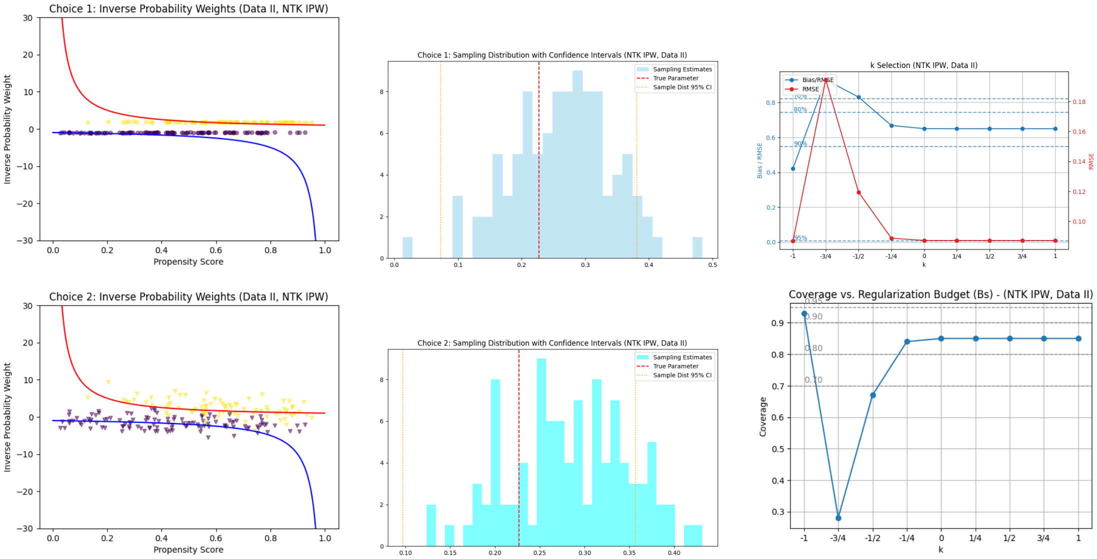


Figure A.9: Semi-Synthetic Framework 2 - IPW w/ NTK Minimax

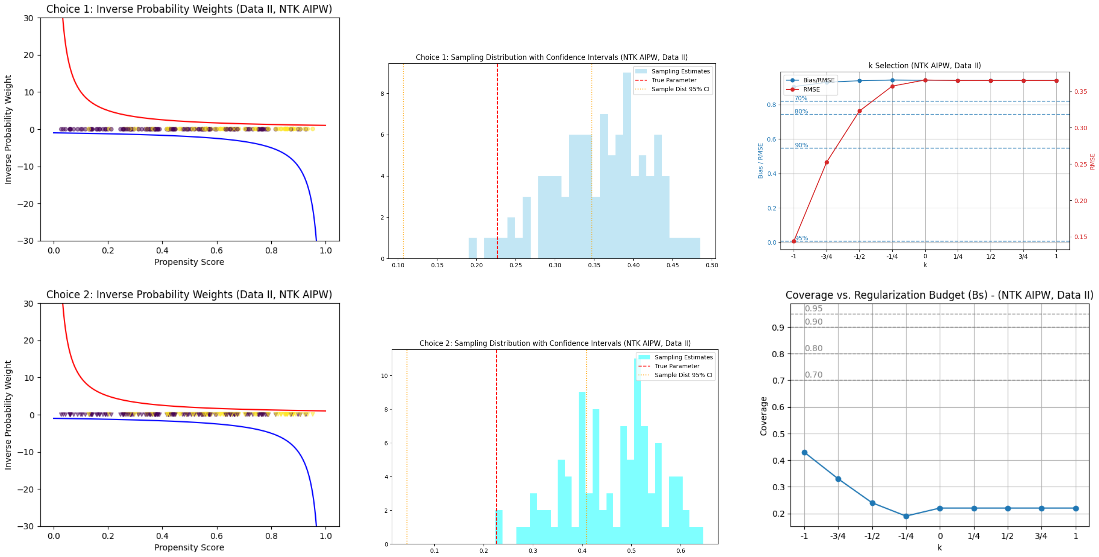


Figure A.10: Semi-Synthetic Framework 2 - AIPW w/ NTK Minimax

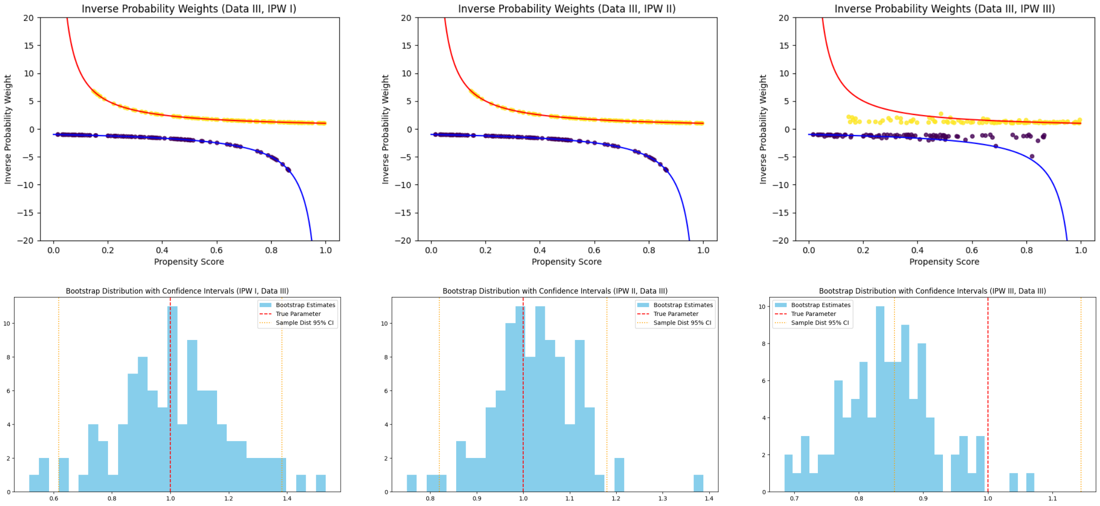


Figure A.11: Semi-Synthetic Framework 3 - IPW I, II, III

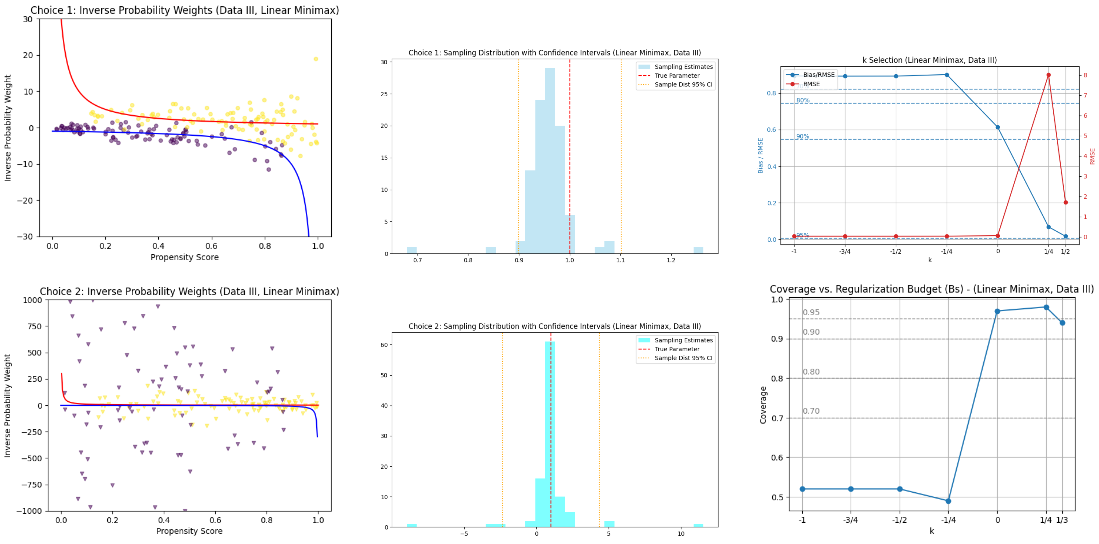


Figure A.12: Semi-Synthetic Framework 3 - IPW w/ Linear Minimax

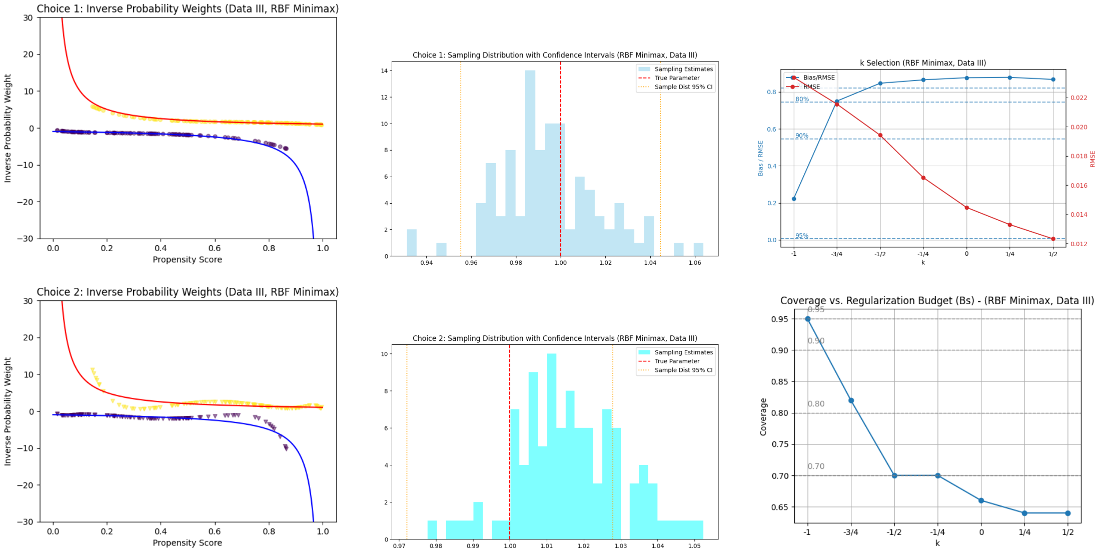


Figure A.13: Semi-Synthetic Framework 3 - IPW w/ RBF Minimax

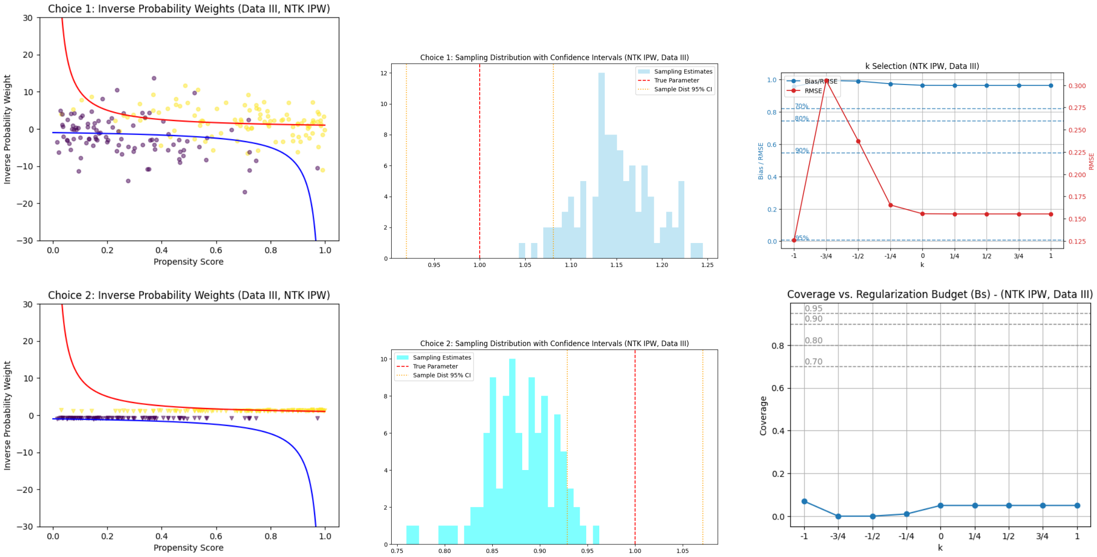


Figure A.14: Semi-Synthetic Framework 3 - IPW w/ NTK Minimax

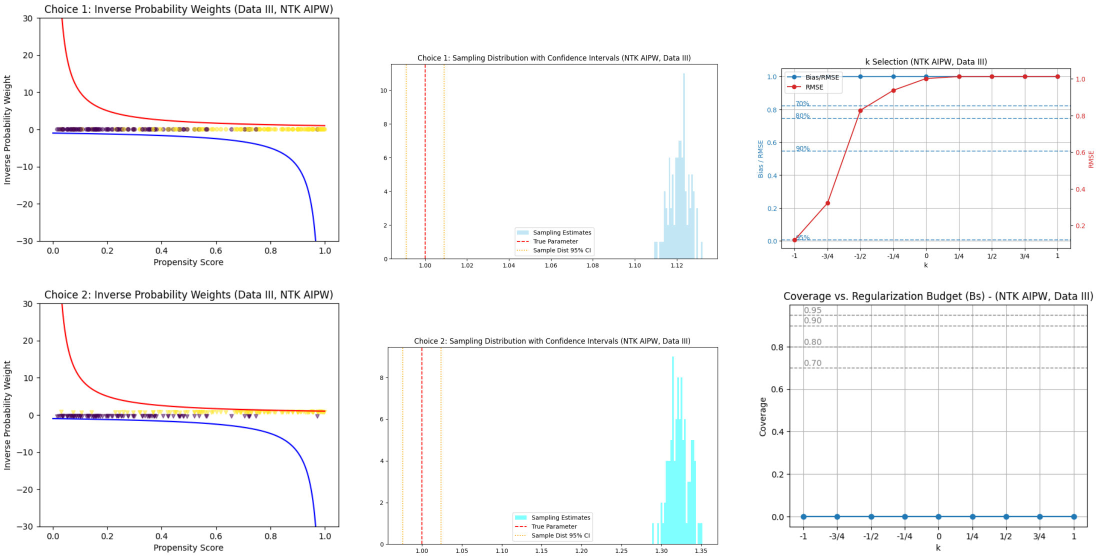


Figure A.15: Semi-Synthetic Framework 3 - AIPW w/ NTK Minimax

Appendix B

Parameter Tuning Selection

	$\hat{\tau}_0$	**Sample** σ	**Coverage**	$E_n[\hat{\tau}_i]$	$\lambda(B)$	B
Truth	1					
Choice 1	0.982	0.017	0.94	0.997	1	0.005
Choice 2	1.012	0.010	0.86	1.010	2.5e-5	1

Table B.1: Parameter Tuning: Data 1, IPW w/ Linear Minimax

	$\hat{\tau}_0$	**Sample** σ	**Coverage**	$E_n[\hat{\tau}_i]$	$\lambda(B)$	B
Truth	1					
Choice 1	1.001	0.007	0.93	1.001	0.005	0.0707
Choice 2	0.993	0.007	0.93	1.000	2.5e-5	1

Table B.2: Parameter Tuning: Data 1, IPW w/ RBF Minimax

	$\hat{\tau}_0$	**Sample** σ	**Coverage**	$E_n[\hat{\tau}_i]$	$\lambda(B)$	B
Truth	1					
Choice 1	1.070	0.022	0.32	1.052	2.5e-5	1
Choice 2	1.070	0.022	0.32	1.052	6.9e-13	6000

Table B.3: Parameter Tuning: Data 1, IPW w/ NTK Minimax

	$\hat{\tau}_0$	**Sample** σ	**Coverage**	$E_n[\hat{\tau}_i]$	$\lambda(B)$	B
Truth	1					
Choice 1	1.002	0.021	0.93	0.999	3600	0.00016
Choice 2	1.034	0.022	0.73	1.030	3000	0.00146

Table B.4: Parameter Tuning: Data 1, AIPW w/ NTK Minimax

	$\hat{\tau}_0$	**Sample** σ	**Coverage**	$E_n[\hat{\tau}_i]$	$\lambda(B)$	B
Truth	0.227					
Choice 1	0.273	0.062	0.96	0.230	1.76e-6	3.76
Choice 2	0.254	0.063	0.95	0.214	1.25e-7	14.14

Table B.5: Parameter Tuning: Data 2, IPW w/ Linear Minimax

	$\hat{\tau}_0$	**Sample** σ	**Coverage**	$E_n[\hat{\tau}_i]$	$\lambda(B)$	B
Truth	0.227					
Choice 1	0.292	0.074	0.94	0.237	0.00035	0.26
Choice 2	0.277	0.062	0.94	0.233	1.76e-6	3.76

Table B.6: Parameter Tuning: Data 2, IPW w/ RBF Minimax

	$\hat{\tau}_0$	**Sample** σ	**Coverage**	$E_n[\hat{\tau}_i]$	$\lambda(B)$	B
Truth	0.227					
Choice 1	0.284	0.078	0.93	0.263	900	0.00016
Choice 2	0.231	0.066	0.85	0.283	2.5e-5	1

Table B.7: Parameter Tuning: Data 2, IPW w/ NTK Minimax

	$\hat{\tau}_0$	**Sample** σ	**Coverage**	$E_n[\hat{\tau}_i]$	$\lambda(B)$	B
Truth	0.227					
Choice 1	0.323	0.061	0.43	0.356	90,000	0.00016
Choice 2	0.323	0.093	0.33	0.461	1161	0.00146

Table B.8: Parameter Tuning: Data 2, AIPW w/ NTK Minimax

	$\hat{\tau}_0$	**Sample** σ	**Coverage**	$E_n[\hat{\tau}_i]$	$\lambda(B)$	B
Truth	1					
Choice 1	0.951	0.051	0.97	0.959	2.5e-5	1
Choice 2	1.109	1.709	0.94	0.971	7.31e-7	5.84

Table B.9: Parameter Tuning: Data 3, IPW w/ Linear Minimax

	$\hat{\tau}_0$	**Sample** σ	**Coverage**	$E_n[\hat{\tau}_i]$	$\lambda(B)$	B
Truth	1					
Choice 1	0.985	0.022	0.95	0.994	1	0.005
Choice 2	1.004	0.014	0.82	1.016	0.0707	0.018

Table B.10: Parameter Tuning: Data 3, IPW w/ RBF Minimax

	$\hat{\tau}_0$	**Sample** σ	**Coverage**	$E_n[\hat{\tau}_i]$	$\lambda(B)$	B
Truth	1					
Choice 1	1.072	0.041	0.05	1.149	3.22e-7	8.80
Choice 2	0.929	0.036	0.07	0.879	900	0.00016

Table B.11: Parameter Tuning: Data 3, IPW w/ NTK Minimax

	$\hat{\tau}_0$	**Sample** σ	**Coverage**	$E_n[\hat{\tau}_i]$	$\lambda(B)$	B
Truth	1					
Choice 1	1.123	0.004	0	1.121	202500	0.00016
Choice 2	1.312	0.012	0	1.322	2614	0.00146

Table B.12: Parameter Tuning: Data 3, AIPW w/ NTK Minimax

Bibliography

[1] Eli Ben-Michael, Avi Feller, David A. Hirshberg, and José R. Zubizarreta. The balancing act in causal inference, 2021. URL `https://arxiv.org/abs/2110.14831`.

[2] David A. Hirshberg and Stefan Wager. Augmented minimax linear estimation, 2020. URL `https://arxiv.org/abs/1712.00038`.

[3] Daniel S Kermany, Michael Goldbaum, Wenjia Cai, M Anthony Lewis, Huimin Xia, and Kang Zhang. Identifying medical diagnoses and treatable diseases by image-based deep learning. *Cell*, 172(5):1122–1131.e9, 2018.

[4] PAUL R. ROSENBAUM and DONALD B. RUBIN. The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1): 41–55, 04 1983. ISSN 0006-3444. doi: 10.1093/biomet/70.1.41. URL `https://doi.org/10.1093/biomet/70.1.41`.