Causal Inference for Traffic Simulations

Tal Kraicer
DDS Faculty
talkraicer@campus.technion.ac.il

Dan Shlomo Mizrahi
DDS Faculty
mizrahid@campus.technion.ac.il

Abstract

In this project, we extensively explore causal inference methods applied to traffic simulations, addressing traffic light control management. Utilizing the SUMO (Simulator of Urban MObility) framework, we generated experimental (simulated an observational study) and testing datasets (from which we calculated the real ATE) to evaluate the effects of various traffic light policies on junction performance. Our study highlights the importance of understanding CI assumptions and methods while raising a few future research directions. The findings underscore the need for further research and the development of more survey studies of causal inference in transportation engineering worlds. For source code and additional resources, visit our GitHub repository [1].

1 Introduction and Prior Work

Researchers have recently tried using data science and machine learning techniques for transportation engineering. Such research includes traffic light scheduling using Reinforcement Learning ([16], [13]), variable speed limit using Machine Learning ([10], [8]), and ramp metering using Machine Learning [7]. This creates numerous opportunities to enhance the current state of transportation, yet it also presents several challenges. One key challenge is causal inference, which involves identifying and estimating the effect of an intervention, despite observing only some outcomes per subject (such as a junction, road, network, or city in the context of transportation) instead of the whole traffic system.

As Dorie et al. suggest [6], researchers may be unaware of the best causal inference estimation method when in need; we believe it is still the case for transportation researchers and engineers. To solve this problem, the authors created a novel competition on simulated data and reported the results and conclusions. This work uses a similar pipeline to create a guide from which transportation engineers could benefit. Because of the scope of this work, we couldn't hold a competition, as the authors did in [6]. Instead, we compared some state-of-the-art causal inference methods, all are black-box methods (without tailing the solution specifically for the data)[see Sec. 4], creating a similar open competition in this field remains an idea for future work [see Sec. 7].

Using a widely used (among transportation engineers) micro-traffic-simulator named SUMO (Simulator of Urban MObility) [2], we generated 2 datasets:

1. An experimental dataset that was generated according to Sec. 2 in order to asses the ability of different methods from Sec. 4.

2. A testing dataset that was generated using the same pipeline from Sec. 2, but under $do(T = 0, 1, 2)$. In other words, the same traffic conditions but using different treatments. The ability to generate such a dataset is unfeasible in real-world experiments, but using the simulator we can measure these results as ground truth, and leverage them to estimate the performance of causal inference methods Sec. 4.

The simulation settings are very simple and described in detail in Sec. 3. The causal inference question that the causal models aim to answer is: *What is the ATE of applying a Traffic Light policy on a junction?*

After simulating appropriate data, we continued to explore CI models for estimation see Sec. 4. These have roused questions and interesting insights discussed about in Sec. 5 and Sec. 7.

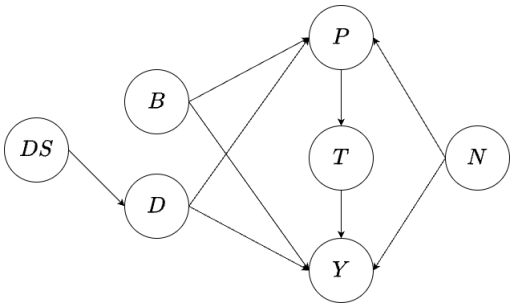
We encourage other researchers to use our generated dataset (which can be found in the GitHub repository[1]).

[1] `https://github.com/Dan551Mizrahi/CausalInferenceProject`

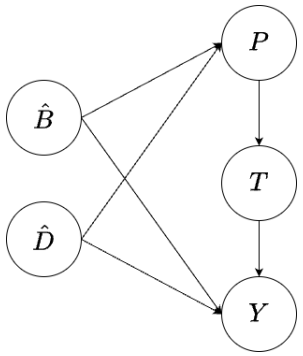
2 Causal Description of The Problem

The experiment settings are as follows: we collect data from fixed structured junctions. Denote by N the structure of a junction (described in Sec. 3.1), which differ from one another in their respective *demand sizes*, DS, distribution of demands for each route, D, and *drivers behaviors* distribution, B. These varieties may create heavy traffic conditions that result in a large *time delay* for each vehicle, or almost zero *time delay* for low values of DS. Using sum as an aggregation method, we get a measurement of delay for the whole junction, denote it by *Prior*, P (see Sec. 3 for a detailed explanation on each). Then, a transportation engineer decides whether and how to change the traffic light policy in the junction using a simple ruled-based procedure (see Sec. 3.3). This will be the treatment, T, which can get three values 0, 1, or 2 (see Sec. 3.3). After applying the treatment, we can measure the new delays in the junction in a different setting (because time has gone by the time we applied it), and we will refer to it as Y. The time passing by can lead to changes in the realization of D (i.e., changes in the arriving process of the drivers) and changes in the realization of B (i.e., changes in each driver's behavior), but we will assume that for each junction the distribution stays the same. The way that we generated data for this experiment is described in Sec. 3. Now, we wish to know the causal effect of our treatment T on Y.

We will borrow the terminology of Dorie et al. paper [6] and differentiate between "oracle" and "nonoracle" variables, meaning whether it is reasonable to assume that a variable would be available to a researcher with real-world data. From the "perspective of the oracle", the experiment can be described in Pearl's causal graph encoding [11] using the following causal graph:



Essentially, we assume that N, D, B, and T are the only things affecting the delay; that is reasonable because, given them, we can simulate (and measure its delay, as in Sec. 3) accurately the entire system, but it might not hold in real life (e.g. time of the year could be a factor, raining day, sunny day, ...). We are also assuming that the demand does not affect behavior and vice versa; this might not hold in real life. The driver who likes to drive fast might prefer a road with no police or a freeway, but in certain real-life situations, drivers can't really decide how to drive (e.g., there is only one way). In our simulated data, both of these assumptions hold. Our experiments confounders are N, B, and D. Note that N, the road system structure is constant throughout this work [see Sec. 3], so it is not an interesting backdoor in this work (it could be very interesting to examine its effect when it is not constant). B and D are not "nonoracle" variables, but we explained in 3.4 why they can be approximated in real-life settings, and we denote their approximations by $\hat{B}$ and $\hat{D}$. So, from the researcher's perspective, the appropriate causal graph is:



2.1 Choosing ATE Over ATT

Unlike Dorie et al., who looked at estimations of the ATT, we decided early on to look at the ATE. We had several considerations in mind:

- When dealing with transportation systems, we want to estimate the possible effect of a treatment on all the junctions we have, not just those we treated.
- When presented to a policymaker unfamiliar with causal inference (e.g., someone in your local town hall), we believe it is easier to understand ATE and justify a decision with it.

2.1.1 Addressing one line in the ATE matrix

Note that because we have 3 treatments, we are dealing with a matrix of ATEs:

$$\begin{bmatrix} 0 & \mathbb{E}[Y_1 - Y_0] & \mathbb{E}[Y_2 - Y_0] \\ \mathbb{E}[Y_0 - Y_1] & 0 & \mathbb{E}[Y_2 - Y_1] \\ \mathbb{E}[Y_0 - Y_2] & \mathbb{E}[Y_1 - Y_2] & 0 \end{bmatrix}$$

But, that is a symmetric matrix, and from linearity of expectation, it holds that:

$$\mathbb{E}[Y_2 - Y_1] = \mathbb{E}[Y_2 - Y_0 + Y_0 - Y_1] = \\ \mathbb{E}[Y_2 - Y_0] - \mathbb{E}[Y_1 - Y_0]$$

So it means we only need to estimate $\mathbb{E}[Y_2 - Y_0]$ and $\mathbb{E}[Y_1 - Y_0]$ to determine the values of all of the matrix.

The rest of our experimental causal assumptions are described in 3 and 4.

3 Experiment Setup

In our experiment, a traffic engineer observes a junction that uses T=0 from Sec. 3.3 for a while and calculates a *Prior*. The *Prior* is the sum of the *totalDelay* of all the incoming vehicles. We define the *totalDelay* as the *departDelay* + *timeLoss* [1] that describes the time the car waited to enter the system + the time it lost compared to driving alone in the route.

Given the prior, the expert determines (stochastically) which treatment to apply, according to Sec. 3.3. We then reset our simulation, measure the experiment's features, as described in Sec. 3.4, as well as the new *totalDelay*, and report it as our Y. This meant to simulate different real-world conditions (for example, different month, other drivers), because, in the real world, we cannot test our treatment on the same conditions as were in the prior, but we can assume that these random variables are generated from the same distribution.

3.1 Road Network

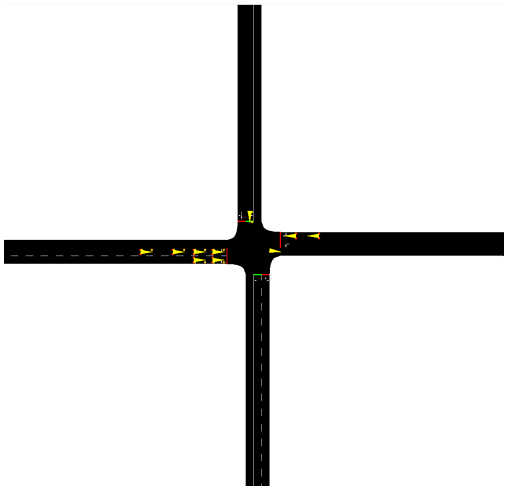


Figure 1: Road network

Our simulation framework models a four-way junction as illustrated in Fig. 1, and in Sec. 2, where we referred to the structure of it as N. Each incoming direction includes two lanes: the right one is dedicated to vehicles aiming to turn right or continue straight, and the left one is for left turns. The outgoing roads include only one lane. The maximal allowed speed is 50 (km/h), stimulating an urban road. We also experimented with a simplified version of the road network featuring a single lane in each direction. However, traffic lights negatively impacted these simulations, as vehicles at the front of each lane occasionally attempted left turns, thereby blocking all incoming traffic in that direction.

3.2 Traffic Demand and Randomization

Since the road network is constant, the difference between our experiments lies in the realization of the demand and behavior of the drivers. For each experiment, the following variables are sampled:

- Demand Size (DS): the number of vehicles per hour that arrive in expectation from each direction. The DS increases and then decreases every 10 minutes time-period (TP) (demonstrated in Fig. 2), simulating a rush hour scenario when in the middle of the simulation there is a demand peak. The initial DS, $DS[0]$, is sampled from a $U[120, 240]$ distribution. Other DS are determined using a multiplication of $DS[0]$ with a scalar. The scalars are constantly chosen, ranging from 1 to 2, with a 0.2 gap between them.

- Incoming Direction Factor (IDF): For each TP, every incoming direction is assigned with a factor (sampled from $U[0.8, 1.2]$ distribution). Note that $DS[TP] \cdot IDF[TP]$ is the expected number of vehicles incoming from a direction per hour in a TP.

- Outgoing Direction Factor (ODF): For each period, every outgoing direction is assigned with an outgoing probability (sampled independently for each incoming direction). The ODF is a 3-dimensional probability vector (sums up to 1). Note that $DS[TP] \cdot IDF[TP] \cdot ODF[TP]$ is the expected number of vehicles driving from incoming direction to outgoing direction per hour in a TP.

- Speed Factor (SF): as explained in [4], the speed factor in sumo defines the speed at which the individual aims to drive in comparison to the maximal allowed speed. The speed factor is randomized for each incoming direction, sampled from a normal distribution $N(\mu, \sigma)$ where $\mu \sim U[0.9, 1.1], \sigma \sim U[0, 0.2]$.

All of the randomized variables mentioned above are sampled using one seed value, such that a seed defines a unique experiment. Using that fact, we can test the performance of different treatments under the same conditions, as described in Sec. 5.1

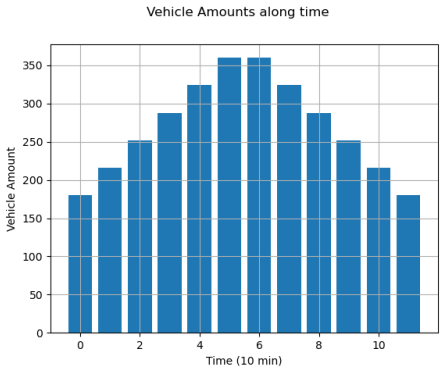


Figure 2: The DS change over time

3.3 Treatment Assignment

In our work, we examined 3 kinds of treatments:

- T=0 - Priority: As described in [3], priority junction is the SUMO default junction configuration, where each vehicle gives a right-of-way to vehicles approaching from its right direction. We refer to this treatment as no treatment, considering that it doesn't include any special intervention from the traffic expert.

- T=1 - Uniform Traffic Light System: As described in [5], we defined for T=1 the simplest version of traffic lights with the default timing that SUMO determined for each phase in the traffic light system, as described in Fig. 3. The timing is 33 seconds for Phase 1, 3 seconds for Phase 2, and 6 seconds for Phase 3.

- T=2 - Delay Based Traffic Light System: The same phases from T=1, but the duration of each phase is smartly determined to reduce the delay it causes to the road network, estimated using smart sensors spread across the junction. The duration for Phase 1 is chosen between 5 and 50 seconds, the duration of Phase 2 is fixed at 3 seconds, and the duration of Phase 3 is chosen between 5 and 50 seconds.

As mentioned above in Sec. 3, we use the prior P to determine the treatment. To maintain *overlap*, our treatment is chosen stochastically. Our traffic expert is implemented using the sigmoid function: $\sigma(x) := \frac{1}{1+e^{-x}}$. We then define the probability to make a treatment - $p := \sigma(c_1 * Prior/c_2 - c_3)$. The constant - $c_1 = 55$ is based on [9] as an estimator for the value of time per vehicle per hour. The constant c_2 includes the normalization of the $Prior$ to hours, and the cost of building a smart traffic light system based on [14], divided by the number of relevant hours where our simulation behavior makes sense (about 10 hours a day), resulting in about $250,000$. The constant c_3 is chosen empirically to be 3, represents that for a junction with $Prior \approx 0$ the probability will be $\sim \sigma(-3) \approx 0$, and for applying a policy the $Prior$ must be around 3 times larger than the cost of building a smart traffic light.

We have chosen the probability to make T=2 as p^2, as deciding twice to make a treatment (traffic light and upgrading to smart one), resulting in a probability of:

$$P(T=t) = \begin{cases} 1-p & \text{if } t=0, \\ p-p^2 & \text{if } t=1, \\ p^2 & \text{if } t=2. \end{cases}$$

The behavior of the treatment assignment function is illustrated in Fig. 4. For small values of p, the probability of $T=0$ is almost 1, and as p increases the probability of $T=2$ increases. We can see that the probability of $T=1$ reaches its maximal value of 0.25 at $p=0.5$, which illustrates that installing a naive traffic light leads to bad results under most priors.

3.4 Extracted Features

We aim to measure all of the confounders in the simulation based on the causal graph presented in Sec. 2. Here, we describe how we estimate the variables:

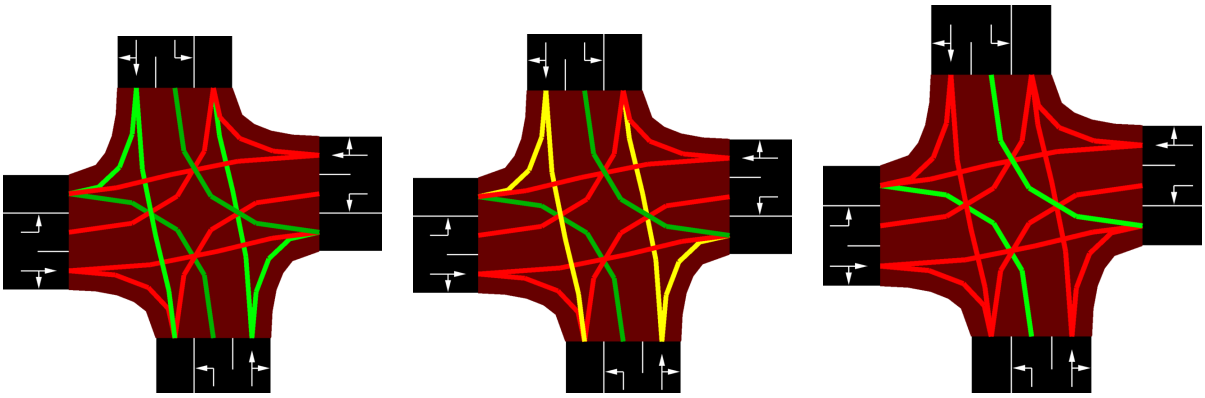
- $\hat{B}$: A 8-dimensional vector representing the mean and std of the SF described in Sec. 3.2 for each incoming direction. We assumed that the SF of each vehicle is "nonoracle" data because, in today's world, several navigation applications estimate it (Waze is an example), so we use those to calculate empirical mean and std.

- $\hat{D}$: A 4-dimensional vector representing the total demand for each incoming direction. We measured the exact number, assuming that we may use a simple loop detector (a very basic sensor used in traffic domains to count the number of vehicles) at a distance from the junction to count the number of incoming vehicles avoiding the post-treatment effects of the treatment.

An example of the extracted data can be found in Table 3. The vectors are separated into different columns, where each input has its incoming direction (East, West, North, or South).

4 Causal Methods

Note: Due to the large number of graphs included in this section and in Section 5, as well as space limitations, we have decided to place most of the graphs in the Appendix. We apologize for any inconvenience this may cause in the reading flow. If any of the graphs



(a) Phase 1 (b) Phase 2 (c) Phase 3

Figure 3: *3 traffic light phases from the simulation. After them, the same 3 phases happen in the other directions and then repeat. Phase 1 allows driving forward, right, and turning left only when clear. Phase 2 allows left turns and finishes the forward and right movements. Phase 3 allows only left turns.*

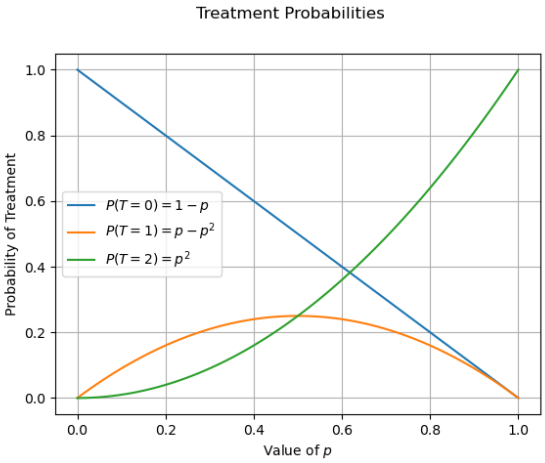


Figure 4: Treatment Assignment Probabilities

is unclear (too small), please go to the GitHub repository, all the graphs here are there.

We used only black box causal inference models, unlike Dorie et al. [6]. We used 9 different models described below with varying sets of parameters. All of them are taken from IBM's `causallib` Python library [12]. This allowed us to examine a variety of causal inference models easily, and we highly recommend any future researchers use our code as a pipeline for comparing causal inference models, especially in the areas of transportation engineering.

4.1 Causal Assumptions

The experiment we created follows several curtail assumptions:

- *Strong ignorability* or *ignorability* with *overlap* (*common support*) - *Ignorability* which means that $Y_0, Y_1 \perp T|P, \hat{D}, \hat{B}$ this is important for the estimation ability of an ATE [6], in our case it approximately holds, as we mentioned $\hat{D}$, and $\hat{B}$ are approximations of the confunders, but are different. As we explained in Sec. 3.3, we created overlap by inherently determining a probability function that for it: $0 < \mathbb{P}(T = i|X) < 1$ for any $i \in 0, 1, 2$ and X.

- *Covariants are not affected by treatment* - As we explained before, we created an isolated road system for it; we can assume it, $\hat{B}, \hat{D}$, are not affected by the treatment. As we mentioned, it may not be the case in real life, but it is true for our data.

- *Consistency* Our data is wholly simulated; thus, the outcome we would expect given data is exactly as it will be.

- *Stable Unit Treatment Value Assumption (SUTVA)* - In our simulation, every junction is isolated and unaffected by any other junction. Some can say, rightfully, that this does not hold in the real world because every junction is part of a larger road system, but we can imagine an experiment that takes place in far enough junctions so they have to affect one another (e.g., different states).

- *Alignment* - This idea is that "The only covariants that have the potential to cause bias in our estimation of treatment effects are those that play a role in both the assignment mechanism and the response surface." [6] This holds since we didn't introduce any other type of variables (D and B both affect the assignment mechanism and the response surface).

4.2 Preprocessing and Scaling

You can take a look at a few lines of the data in 3. Now, looking at the histogram of the full training dataset (see 5), a data set that for every junction, we have exactly one treatment and outcome, we can notice that:

1. Most of the data has a uniform shape-like distribution (makes sense given the simulation construction), except the prior, which has a fast decaying distribution.
2. Each feature has a different scale of values.
3. Interestingly enough, the prior's and Y's distributions are not so similar (note that for the prior, the x-axis scale is exponential). We can visually see the effect of the different treatments on the delays, and that is a great verification.

Given the above, as good data scientists, we wanted to scale the data somehow. Eventually, we used the classic `MinMaxScaler`, but from what we saw, it didn't matter for the results. We also decided after some thought that we should not scale the prior variant because its direct connection to the outcome, it yielded slightly better results.

4.3 Short Review of the Methods

We used 9 methods including (by order of appearance in graph):

1. Inverse Probability Weighting (IPW) - Built to estimate ATE, it uses every individual inverse *propensity score*, the probability that she belongs to some treatment group to weight her sample. To estimate the propensity scores, we used `Logistic Regression` with $l1$ penalty and `Gradient Boosting` (See below for the evaluation of our propensity scores).
2. Matching - To estimate the effect of the treatment, we match every treated individual with someone similar to him who is not treated (it could be more, but unfortunately, we did not have the time to examine alternatives other than 1NN). We used Matching with Euclidean distance and Mahalanobis distance.
3. Targeted Maximum Likelihood Estimation (TMLE) [15] - A complex doubly robust model, essentially the idea is to focus the estimator on the treatment effect mainly. We used to version of it found in [12] repository (including one that creates polynomial features [`TMLE_complex`]).
4. Propensity Matching - We use propensity scores to calculate our neighbors.
5. Standardization (Direct Outcome Prediction Model) [S-Learner] - A "standard" *standardization* model is simply a S-Learner, meaning it gets the treatment as an additional feature. We used a S-Learner with linear regression and one with a gradient-boosting regressor.

Surprisingly, in our first tries, all the propensity models we tried to train did not converge (solver did not converge). So we had to tune the parameters a little bit, and we found that gradient boosting is much easier to train than logistic regression (surprise), not because of the time,but because it gets good results easily.

How did we know if a propensity score model is good? By the AUC of it, we are dealing with multiclass treatment so we checked both AUC ovo (one vs one) and AUC ovr (one vs rest).

The best we got was `AUC ovo: 0.82, AUC ovr: 0.89` by logistic regression.

4.4 Degree of overlap

One quick way to understand the data's overlap is as shown in 6 and 7. We want to see how the propensity scores are distributed over the treatments. As you can see, it's pretty poorly distributed. So, we created another dataset, a subsample that has better overlap.

5 Results

We want to evaluate which causal inference method estimates the ATE most accurately compared to the simulator's ground truth using some traffic light control policies and a large set of scenarios and causal inference methods.

5.1 Ground Truth

Table 1 presents the ground truth ATEs calculated using the simulator. Each experiment was held under do $T = 0, 1, 2$, so we could calculate the exact effect of each treatment. The confidence interval (CI) was calculated using bootstrap, where we sampled a subset of the same size as the number of experiments we held (with replacement), and calculated the ATEs over the sample, then using percentile-based CI. We used 0.975 and 0.025 percentiles for 95% confidence interval.

	T=0	T=1	T=2
T=0	0.0 ± 0.0	-42207.2 ± 5640.6	-58498.7 ± 5761.2
T=1	42207.2 ± 5640.6	0.0 ± 0.0	-16291.5 ± 242.1
T=2	58498.7 ± 5761.2	16291.5 ± 242.1	0.0 ± 0.0

Table 1: ATE calculation and half 95% CI length of the ground truth. The value is the column treatment's Y minus the row treatment's Y

As we can see in Table 1, $T = 1$ is significantly better than $T = 0$, and $T = 2$ is slightly better than $T = 1$. These results can be used to choose on simulator-based data which treatment to use, but as mentioned above we will use them as ground truth to test wether causal inference methods are able to reproduce them.

5.2 Naive approach - RCT

As a naive approach, we tried to simply calculate the ATEs based on the data produced using the pipeline presented in Sec. 2 using a simple average function over each treatment. Assuming that the number of experiments we held (3000) is large enough, and under the assumption of randomized controlled trial, such estimator should predict the ground-truth ATEs accurately.

	T=0	T=1	T=2
T=0	0.0	30724.0	35438.2
T=1	-30724.0	0.0	4714.3
T=2	-35438.2	-4714.3	0.0

Table 2: ATE calculation under the assumption of RCT. the value is the column treatment's Y minus the row treatment's Y.

As you can see in Table 2, in comparison to Table 1, the results are way off. Moreover, the sign of the values is exactly opposed to the one in the ground truth, yielding opposite conclusions. Therfore we must use a smarter approach to estimate the ATEs.

5.3 Methods Performance

We will now address all the results in C. From a quick first sight, you can see that the results are not as good as those of Dorie et al. [6] results; we couldn't fit a good model to our needs. However, we should mention a few interesting things that have arisen from these results.

5.3.1 How we measured error of a model?

Dorie et al. measure the models with a variety of factors leading with $l1$ error and $RMSE$. Although we did cover some $l1$ (see Sec. 5.3.4) error, it does not fit correctly to our data. Think about the following scenarios: Our model predicts ATE of 110 instead of 100, and the second scenario is that the model predicts ATE of 1010 instead of 1000. Under $l1$, or $l2$ penalties, the model will get the same penalty in both cases, but it is clear that in scenario one, it should get a much more significant penalty. So, we will discuss *Relative Error*, $e(Y, \hat{Y}) = \left(\frac{|Y-\hat{Y}|}{|Y|}\right)$,which we find more suitable for this experiment.

5.3.2 Three sub-datasets

We work with three datasets to see how the results vary (all the experiments we did, we repeated on all of them):

1. Full dataset - the 3000 records we mentioned before. Every junction appears exactly once, and we only have access to this treatment during training.

2. 100 first rows - the first 100 rows of the full dataset.

3. Clipped dataset - As discussed in Section 4.4, we took a subsample of the dataset according to the data's propensity scores to create a bigger overlap.

5.3.3 Estimating ATE

From the graphs in the first part of Sec. C, we can learn that, first of all, this is not a trivial task because an S-Learner with linear regression performed poorly on it. Almost all the other models (except $TMLE$) successfully estimated the ATE between $T = 0$ and $T = 1$. Interestingly, the double robust model failed here. For $T = 0$ and $T = 2$, models have a harder time; this probably has a direct link to 7, the overlap is not optimal for them. Most models performed better on the smaller dataset.

Now recall that in Sec. 2, we showed that we only need to estimate these two, but for better understanding, we wanted to see how the models perceived the potential outcomes, estimating the ATE of $T = 1$ and $T = 2$ and they completely failed, this is very reasonable given that its, not their purpose, but somewhat disappointing, in Sec. 5.3.4 we will see a different result that cheered us.

Propensity Matching was the most consistent method among our methods.

5.3.4 Individual Predication Errors ($l1$ and *Relative Error*)

We wanted to assess how good our models are so we computed for each junction the effect of the policy individually (for the methods we could). Then we use the metrics we talked about earlier ($l1$ and *Relative Error*) the results appear in Sec. C. The best results overall are achieved when training on the clipped by propensity dataset, and the methods that shine are Matching methods, and S-Learner with GB.

Notice two things: unlike before, the best performance of the models is actually on $T = 1$ and $T = 2$, and while $l1$ are very high, relative errors stay relatively low; the estimations are the same magnitude as the actual effect.

5.3.5 Limitations the results

Unlike the participants in Dorie et al.'s competition [6] we are not casual inference experts so we could not deal with a lot of complicated methods of causal inference or DIY methods. Due to the scope of this work, we couldn't tune the suitable parameters for each model, nor could we even try a lot of models. We also didn't do a bootstrap for this part as the training is too exhaustive (timewise).

6 Challenges

The work included many challenges because the scope of our work included both designing the causal problem (Sec. 2), creating a suitable dataset to answer the causal question (Sec. 3), and solving the causal question itself using a variety of causal inference methods (Sec. 4, Sec. 5).

As for the causal problem, some of the key challenges were:

- Addressing a meaningful transportation engineering challenge.
- Understanding the variables we would like to measure, distinguish between pre-treatment and post-treatment variables, and understand the relation between the variables. Because of the way we generate our data, we have to understand which variables are distributions and which are realizations.
- Modeling the prior knowledge we have before applying a treatment ($Prior$) in a simple yet meaningful way.

As for the data generation, the main difficulties we faced were:

- Creating randomized results yet reproducible for testing under do $T = 0, 1, 2$.
- Implementing a modular script that allows checking of many kinds of transportation networks built in SUMO.
- Designing a road network on which makes sense under some conditions to append a traffic light system.
- Finding demand sizes range that introduce the benefits of all of the treatments, without the use of any real-world data (haven't found).
- Creating a treatment assignment policy that maintains *overlap* and applies a reasonable treatment as a traffic engineer might have done.

As for solving the causal problem using methods:

- Most CI methods are suited for binary treatment. We had to make changes to address this issue.
- Work with limited time and resources.
- We encounter a lack of suitable evaluation methods.

7 Discussion and Future Work

Even though we faced many significant challenges as we elaborated on Sec. 6, we managed to create a full simulation pipeline that resulted in an interesting dataset (Table 3) that maintains overlap (Sec. 4). As we seen in Sec. 5, the naive RCT failed and led to opposite results. We then used several CI methods to estimate the ATE. The results as we explored in Sec. 5 mainly raise questions, but we managed to learn three things: when trimmed by propensity models tend to do better, matching models are powerful tools despite their simplicity, and you should probably explore more Relative Error.

The road network includes only one junction and doesn't measure the long-term effect. We only tested on a closed environment without measuring the effects on a larger road network where routing applications such as Waze may be used, allowing vehicles to change their behavior in an unmeasurable way with relation to our treatment.

As extensions, many other versions of the problem can be tested. Some of them are:

- Checking more treatment assignment functions.
- Checking smarter treatments such as machine learning-based treatments ([16]).
- Checking a different set of traffic light phases.
- Using more complicated causal inference methods, and DIY methods.
- Changing the behavior of the DS change over time.

8 LLM chatbots usage

Special thanks to Gemini (`https://gemini.google.com/`) and ChatGPT (`https://openai.com/index/chatgpt/`) for the help throughout the research: code writing, latex handling, and phrasing.

References

[1] Sumo delay explanation. `https://sumo.dlr.de/docs/Simulation/Output/TripInfo.html`.

[2] SUMO documentation. `https://sumo.dlr.de/docs/index.html`.

[3] Sumo priority explanation. https://sumo.dlr.de/docs/Networks/SUMO_Road_Networks.html.

[4] Sumo speed factor documentation. https://sumo.dlr.de/docs/Simulation/VehicleSpeed.html#edgelane_speed_and_speedfactor.

[5] Sumo traffic light system explanation. https://sumo.dlr.de/docs/Simulation/Traffic_Lights.html.

[6] Vincent Dorie, Jennifer Hill, Uri Shalit, Marc Scott, and Dan Cervone. Automated versus do-it-yourself methods for causal inference: Lessons learned from a data analysis competition. *Statistical Science*, pages 43–68, 2019.

[7] Saeed Ghanbartehrani, Anahita Sanandaji, Zahra Mokhtari, and Kimia Tajik. A novel ramp metering approach based on machine learning and historical data. *Machine Learning and Knowledge Extraction*, 2(4):379–396, 2020.

[8] Martin Gregurić, Krešimir Kušić, and Edouard Ivanjko. Impact of deep reinforcement learning on variable speed limit strategies in connected vehicles environments. *Engineering Applications of Artificial Intelligence*, 112:104850, 2022.

[9] Texas A&M Transportation Institute. Value of delay time for use in mobility monitoring efforts, 2017. Accessed: 2024-11-05.

[10] Krešimir Kušić, Ivana Dusparic, Maxime Guériau, Martin Gregurić, and Edouard Ivanjko. Extended variable speed limit control using multi-agent reinforcement learning. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–8, 2020.

[11] Judea Pearl. *Causality*. Cambridge university press, 2009.

[12] Yishai Shimoni, Ehud Karavani, Sivan Ravid, Peter Bak, Tan Hung Ng, Sharon Hensley Alford, Denise Meade, and Yaara Goldschmidt. An evaluation toolkit to guide model selection and cohort definition in causal inference. *arXiv preprint arXiv:1906.00442*, 2019.

[13] Seungah Son and Juhee Jin. Applying reinforcement learning to optimize traffic light cycles, 2024.

[14] The Pricer. How much does a traffic light cost?, 2024. Accessed: 2024-11-05.

[15] Mark J Van Der Laan and Daniel Rubin. Targeted maximum likelihood learning. *The international journal of biostatistics*, 2(1), 2006.

[16] Bin Wang, Zhengkun He, Jinfang Sheng, and Yu Chen. Deep reinforcement learning for traffic light timing optimization. *Processes*, 10(11), 2022.

Appendices

A Data Example

Y	T	d_W	d_S	d_N	d_E	b_std_W	b_std_S	b_std_N	b_std_E	b_mean_W	b_mean_S	b_mean_N	b_mean_E	Prior
34898.61	1	349	424	379	400	0.061298	0.046485	0.076154	0.033245	0.922521	0.990401	0.985858	1.03025	6355.47
46908.66	2	661	605	623	598	0.144297	0.028254	0.053774	0.077906	1.04941	0.905537	0.922665	0.985786	165952
7265.94	0	382	405	373	367	0.066335	0.023232	0.064352	0.131573	1.083979	1.087753	0.92067	1.02515	6210.39

Table 3: Example of data generated from the simulator as inputs for the causal methods

B Preprocessing

B.1 Data Statistics (short EDA)

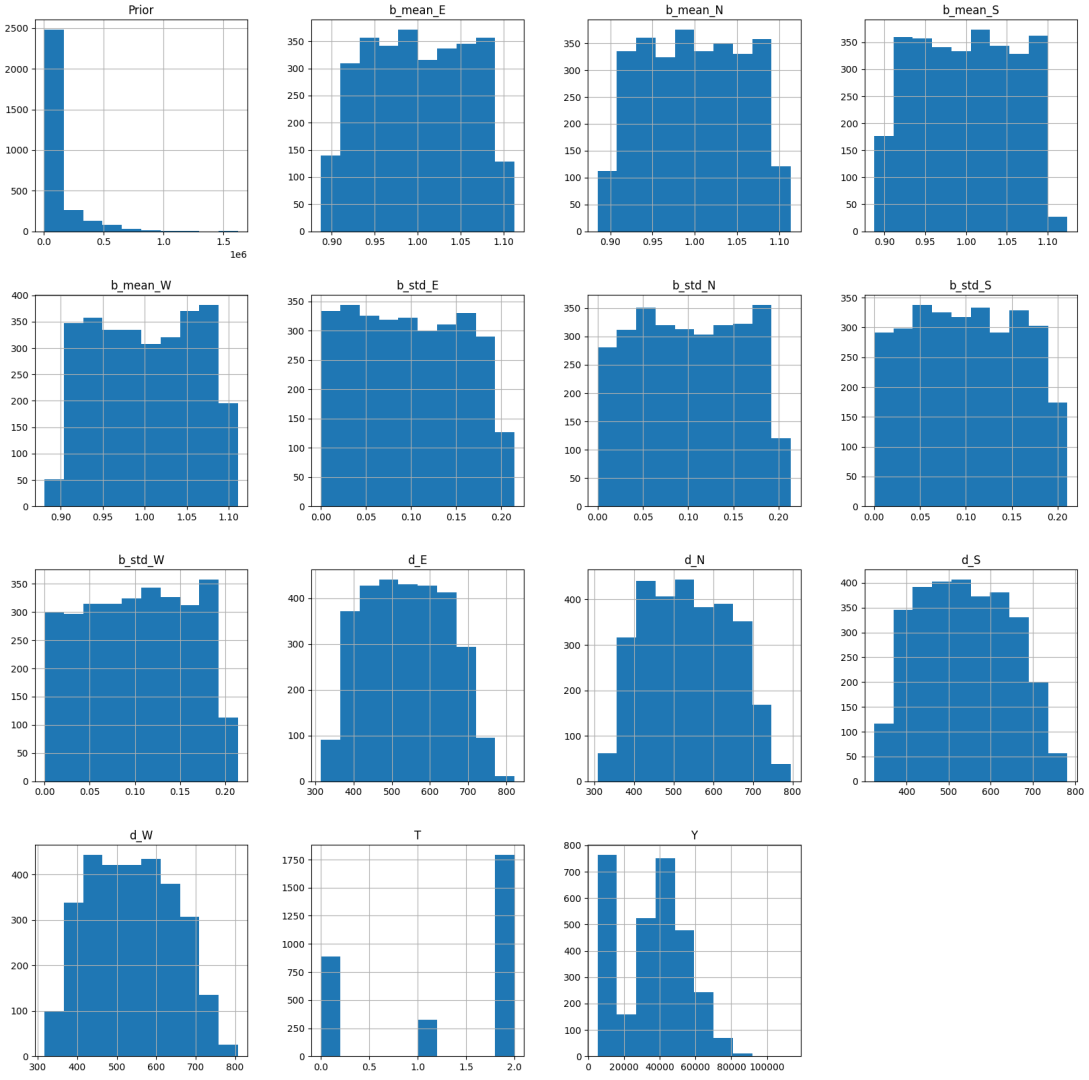
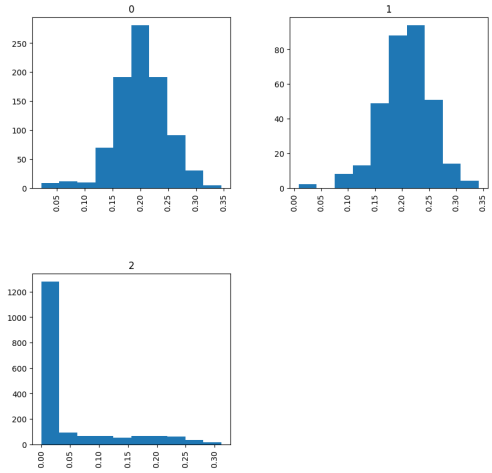
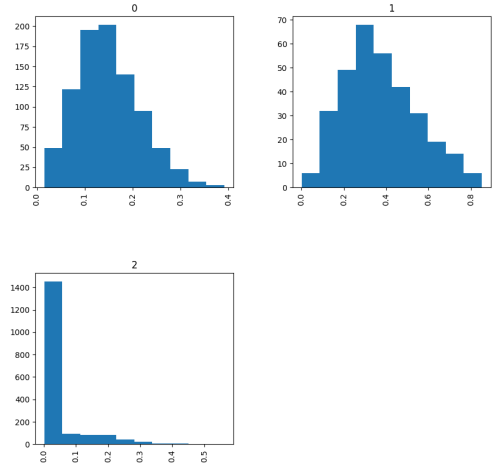


Figure 5: Initial histograms describing the data, 3000 junctions. Interesting difference between the Prior P and the outcome Y.

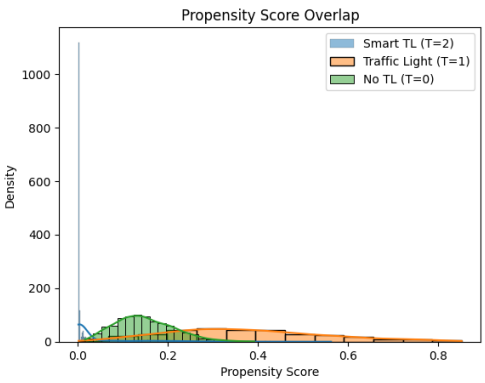


(a) Logistic Regression propensity scores by Treatment

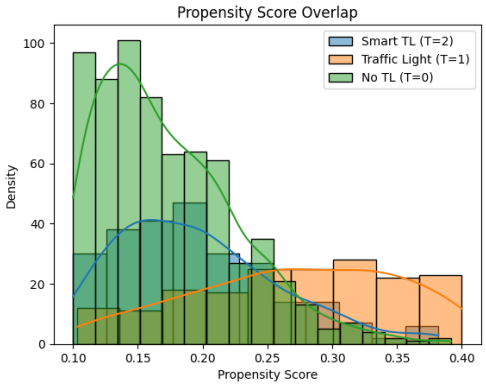


(b) Gradient Boosting propensity scores by Treatment

Figure 6: *A visual representation of the data overlap, as you can see for $T = 2$ it seems there is less overlap.*



(a)The full version of the graph.



(b) A clipped version.

Figure 7: *Another visual representation of an overlap estimation. Both graphs were created using gradient-boosting propensity scores. The graph on the right side is a clipped version between propensities in the interval* $[0.1, 0.4]$*; we will use this dataset later.*

C Results

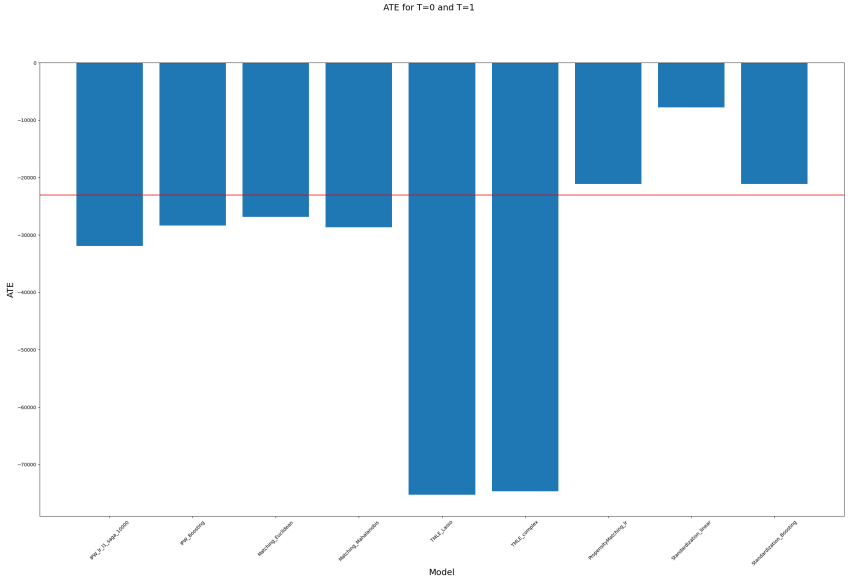


Figure 8: *ATE estimations by all of the models between* $T = 0$ *and* $T = 1$ *using the full dataset for training. Red is the GT* $\mathbb{E}[Y_0 - Y_1]$.

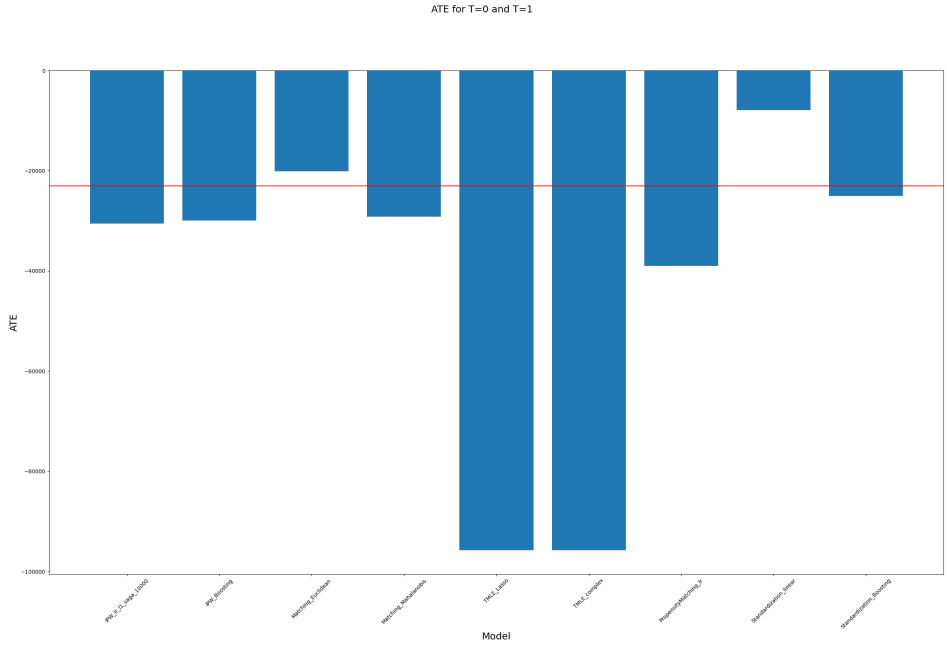


Figure 9: *ATE estimations by all of the models between* $T = 0$ *and* $T = 1$ *using the first* 100 *training examples in the dataset. Red is the GT* $\mathbb{E}[Y_0 - Y_1]$.

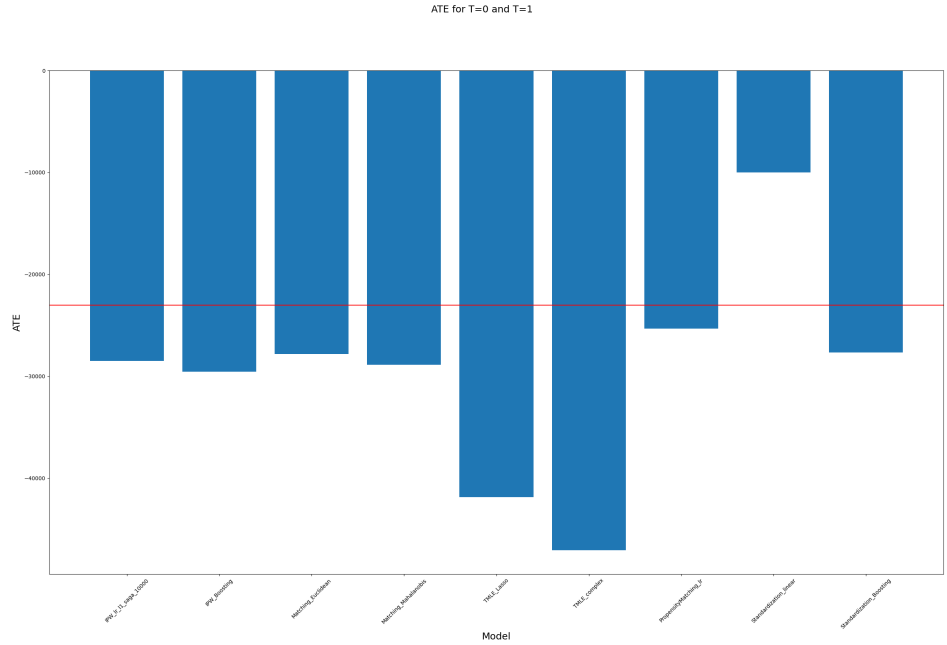


Figure 10: *ATE estimations by all of the models between* $T = 0$ *and* $T = 1$ *using the clipped by propensity dataset for training. Red is the GT* $\mathbb{E}[Y_0 - Y_1]$.

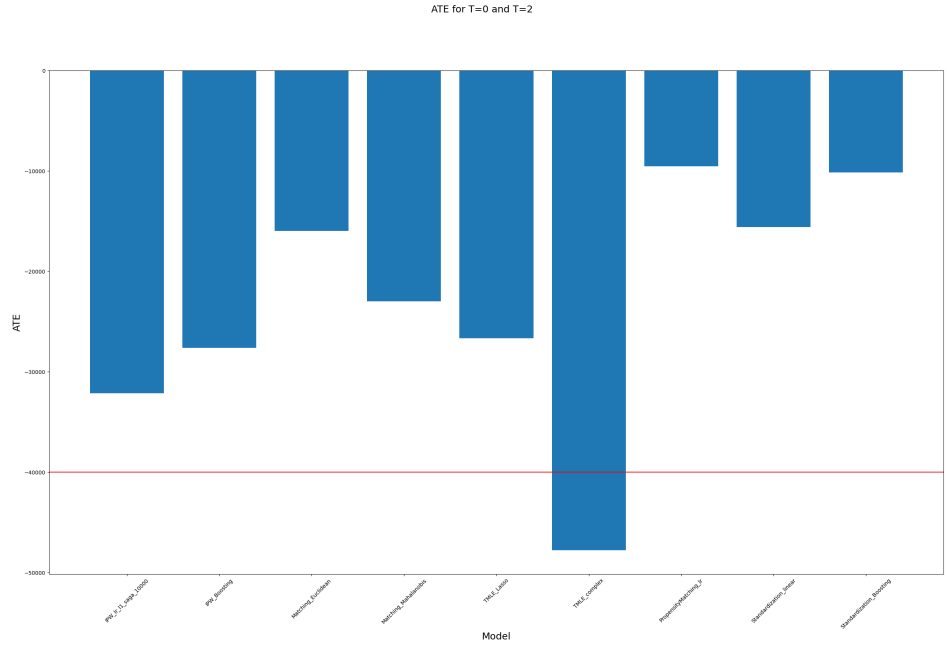


Figure 11: *ATE estimations by all of the models between* $T = 0$ *and* $T = 2$ *using the full dataset for training. Red is the GT* $\mathbb{E}[Y_0 - Y_2]$.

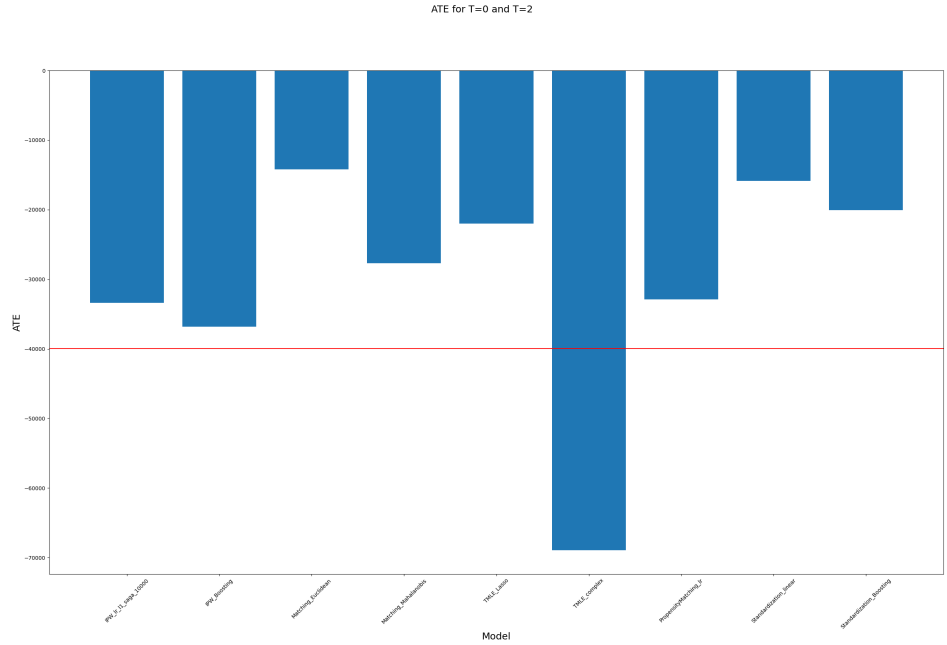


Figure 12: *ATE estimations by all of the models between* $T = 0$ *and* $T = 2$ *using the first* 100 *training examples in the dataset. Red is the GT* $\mathbb{E}[Y_0 - Y_2]$.

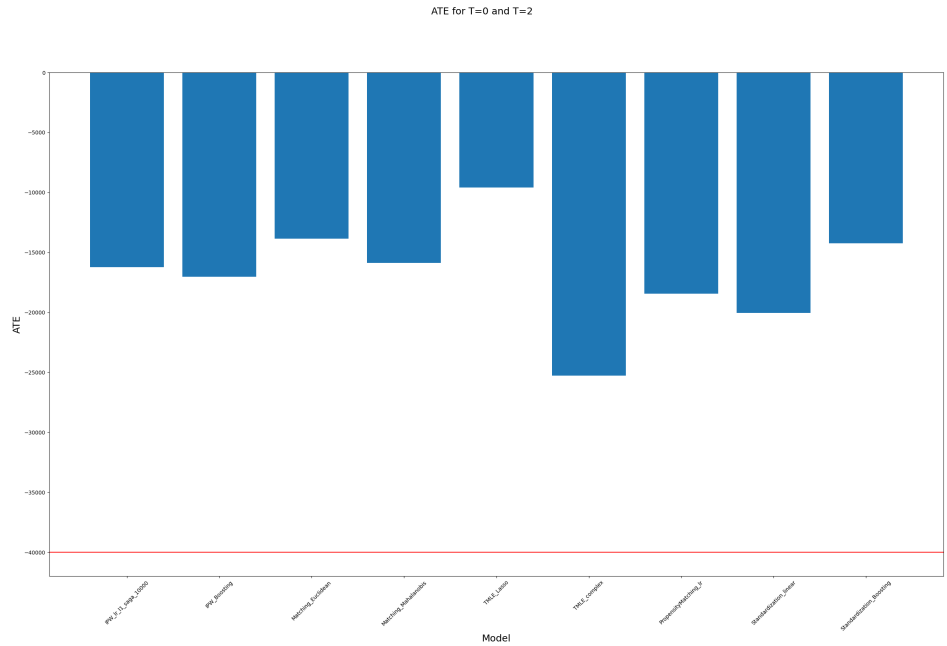


Figure 13: *ATE estimations by all of the models between* $T = 0$ *and* $T = 2$ *using the clipped by propensity dataset for training. Red is the GT* $\mathbb{E}[Y_0 - Y_2]$.

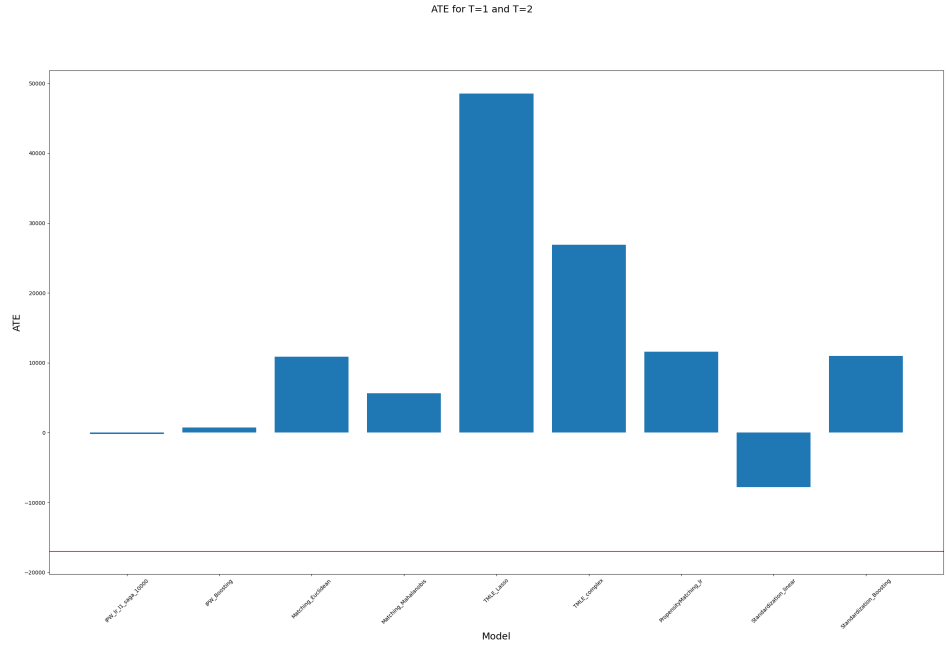


Figure 14: *ATE estimations by all of the models between* $T = 1$ *and* $T = 2$ *using the full dataset for training. Red is the GT* $\mathbb{E}[Y_1 - Y_2]$.

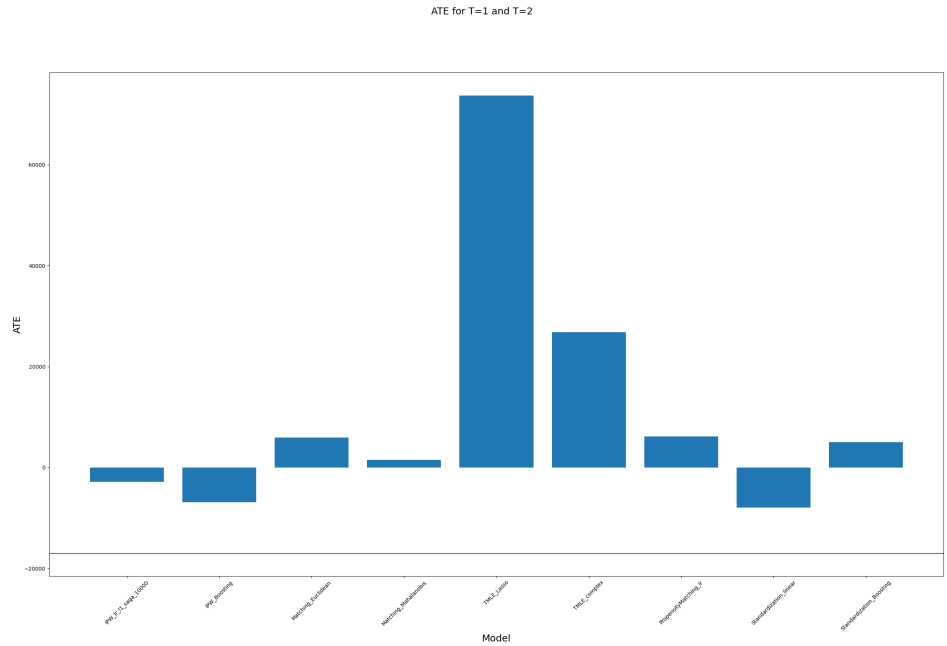


Figure 15: *ATE estimations by all of the models between* $T = 1$ *and* $T = 2$ *using the first* 100 *training examples in the dataset. Red is the GT* $\mathbb{E}[Y_1 - Y_2]$.

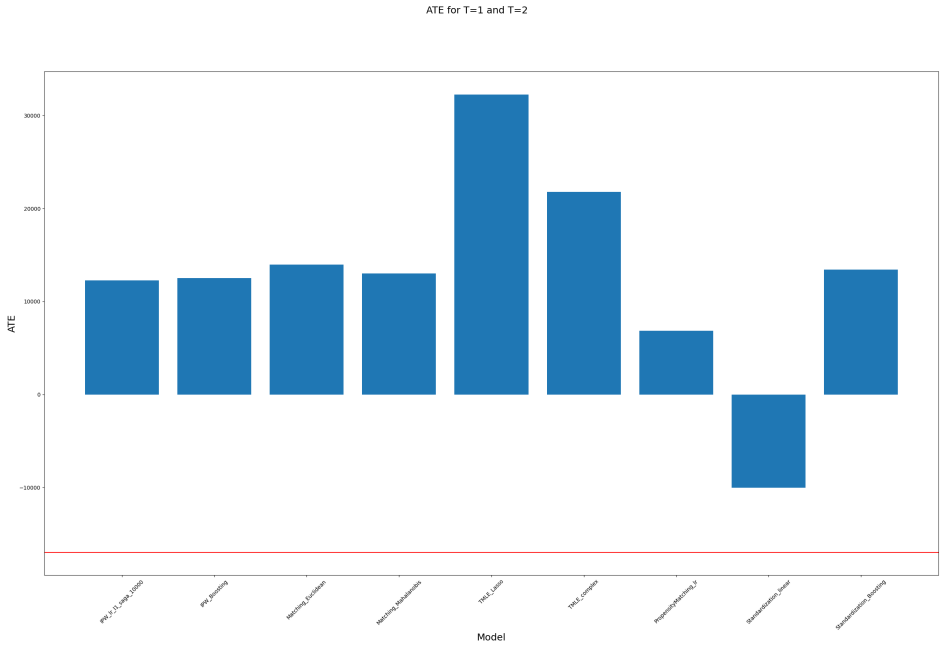


Figure 16: *ATE estimations by all of the models between* $T = 1$ *and* $T = 2$ *using the clipped by propensity dataset for training. Red is the GT* $\mathbb{E}[Y_1 - Y_2]$.

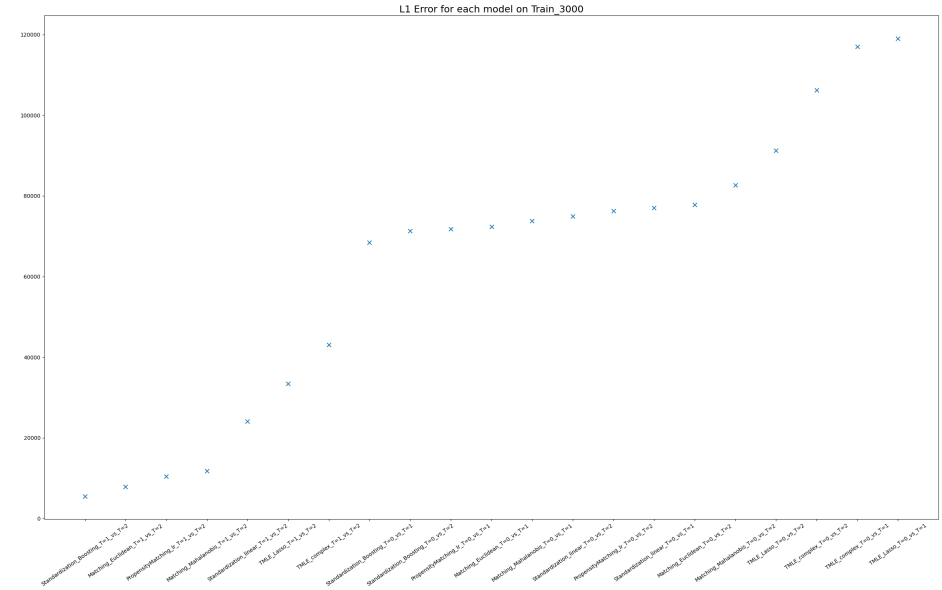


Figure 17: *Mean* $l1$ *error on all of the junctions, we predict for each junction with the model and then compare the induced effect to the real one* (l_1). *Here, only models can predict each object in an observational study. This graph concerns models that train on the full training dataset.*

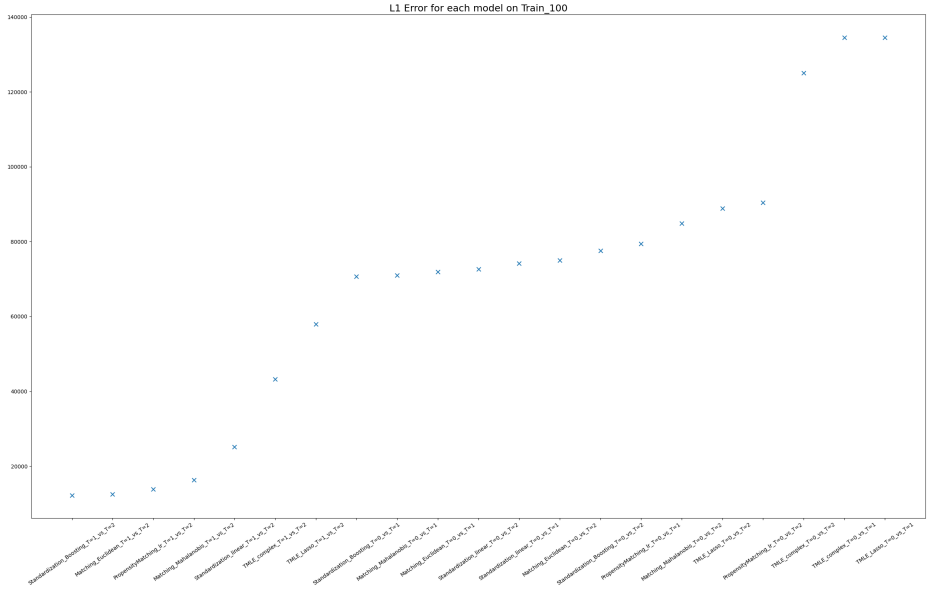


Figure 18: *Mean* $l1$ *error on all of the junctions, we predict for each junction with the model and then compare the induced effect to the real one* (l_1)*. Here, only models can predict each object in an observational study. This graph concerns models that train using the dataset's first* 100 *training examples.*

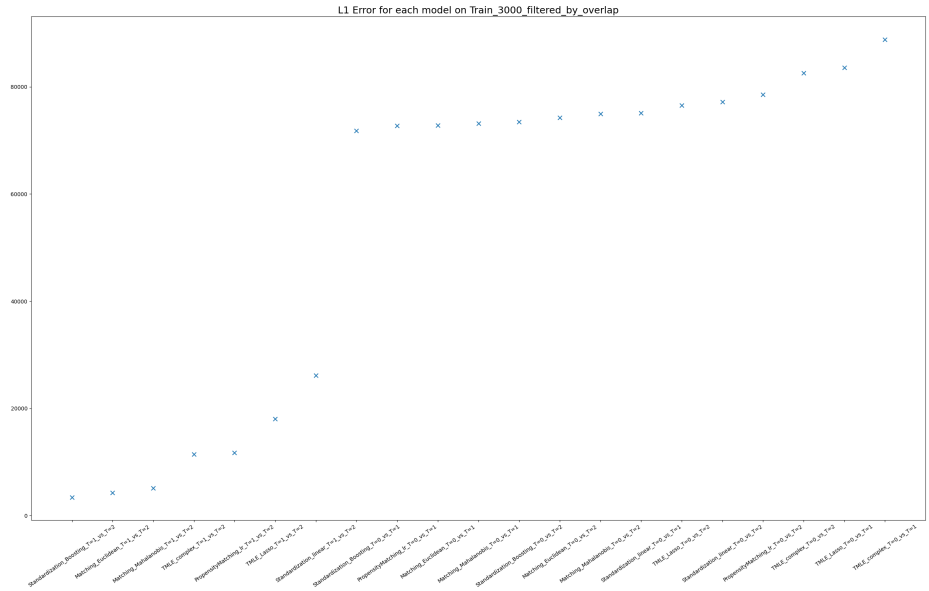


Figure 19: *Mean* $l1$ *error on all of the junctions, we predict for each junction with the model and then compare the induced effect to the real one* (l_1)*. Here, only models can predict each object in an observational study. This graph concerns models that train using the clipped by propensity dataset for training.*

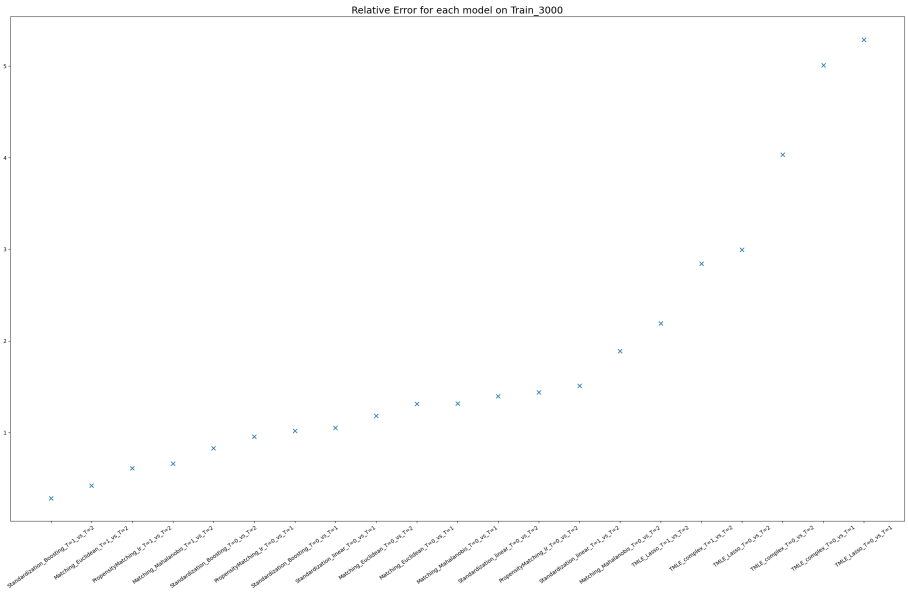


Figure 20: *Mean relative error on all of the junctions, we predict for each junction with the model and then compare the induced effect to the real one* $\left(\frac{|Y-\hat{Y}|}{|Y|}\right)$*. Here, only models can predict each object in an observational study. This graph concerns models that train on the full training dataset.*

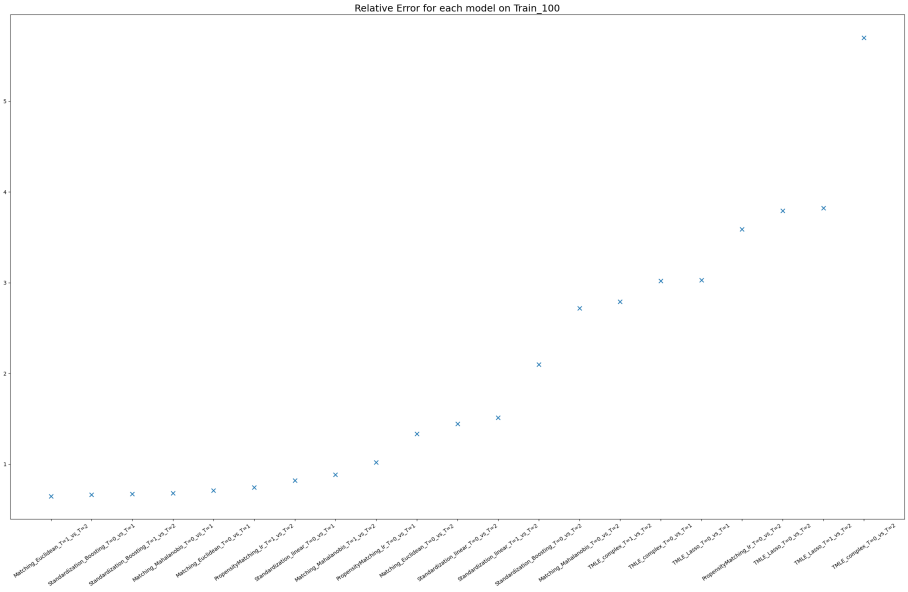


Figure 21: *Mean relative error on all of the junctions, we predict for each junction with the model and then compare the induced effect to the real one* $\left(\frac{|Y-\hat{Y}|}{|Y|}\right)$*. Here, only models can predict each object in an observational study. This graph concerns models that train using the dataset's first* 100 *training examples.*

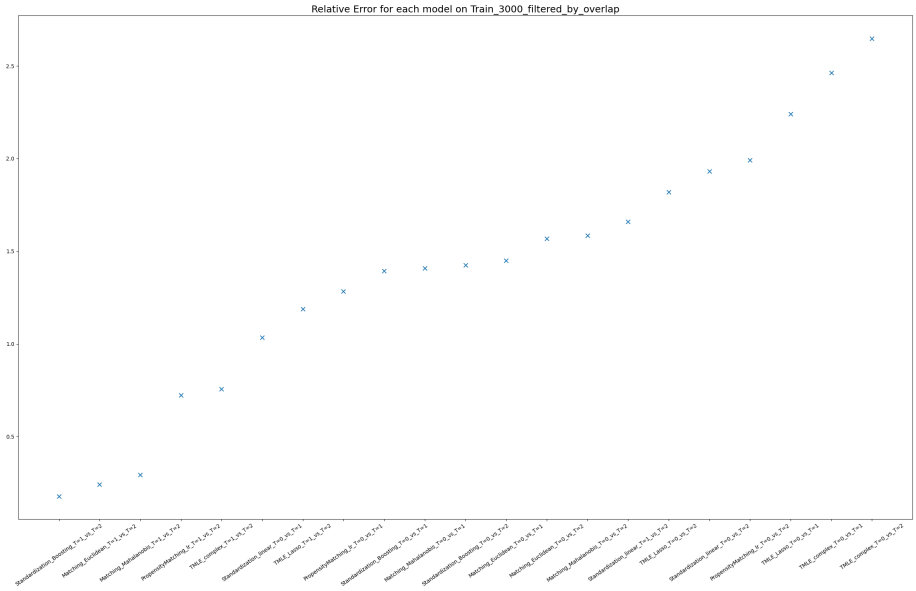


Figure 22: *Mean relative error on all of the junctions, we predict for each junction with the model and then compare the induced effect to the real one* $\left(\frac{|Y-\hat{Y}|}{|Y|}\right)$*. Here, only models can predict each object in an observational study. This graph concerns models that train using the clipped by propensity dataset for training.*