Journal of Machine Learning Research 22 (2021) 1-41 Submitted 10/19; Revised 12/20; Published 1/21

FLAME: A Fast Large-scale Almost Matching Exactly Approach to Causal Inference

Tianyu Wang TIANYU@CS.DUKE.EDU
Marco Morucci MARCO.MORUCCI@DUKE.EDU
M. Usaid Awan MUHAMMAD.AWAN@DUKE.EDU
Yameng Liu YAMENG.LIU@DUKE.EDU
Sudeepa Roy SUDEEPA@CS.DUKE.EDU
Cynthia Rudin CYNTHIA@CS.DUKE.EDU
Alexander Volfovsky ALEXANDER.VOLFOVSKY@DUKE.EDU
Duke University

Editor: Russ Greiner

Abstract

A classical problem in causal inference is that of *matching*, where treatment units need to be matched to control units based on covariate information. In this work, we propose a method that computes high quality almost-exact matches for high-dimensional categorical datasets. This method, called *FLAME (Fast Large-scale Almost Matching Exactly)*, learns a distance metric for matching using a hold-out training data set. In order to perform matching efficiently for large datasets, FLAME leverages techniques that are natural for query processing in the area of database management, and two implementations of FLAME are provided: the first uses SQL queries and the second uses bit-vector techniques. The algorithm starts by constructing matches of the highest quality (exact matches on all covariates), and successively eliminates variables in order to match exactly on as many variables as possible, while still maintaining interpretable high-quality matches and balance between treatment and control groups. We leverage these high quality matches to estimate conditional average treatment effects (CATEs). Our experiments show that FLAME scales to huge datasets with millions of observations where existing state-of-the-art methods fail, and that it achieves significantly better performance than other matching methods.

Keywords: observational studies, distance metric learning, heterogeneous treatment effects, algorithms, databases

1. Introduction

Questions of robust causal inference, beyond simple correlations or model-based predictions, are practically unavoidable in health, medicine, or social studies. Causal inference goes beyond simpler correlation, association, or model-based predictive analysis as it attempts to estimate the causal effects of a certain intervention.

Much of the available data in the clinical and social sciences is observational. In such situations individuals may be selected in a way that treatment assignments depend on outcomes: people who benefit more from pain relievers tend to take them more often,

individuals who are likely to succeed in higher education are more likely to enroll in it, and so forth. Estimating causal effects in an observational setting becomes a problem of representing the available data as if it were collected from a randomized experiment in which individuals are assigned to treatment independently of their potential outcomes.

A natural approach to observational studies is matching treated and control units such that underlying background covariates are balanced (Chapin, 1947; Greenwood, 1945). Under the assumption of unconfoundedness, such matches allow the practitioner to estimate causal effects. If we can match units exactly (or almost exactly) on raw covariate values, the practitioner can further interpret causal estimates within matched groups as conditional average treatment effects. Exact matching increases the interpretability and usefulness of causal analyses in several ways: It is a tool for granular causal analysis that can provide crucial information on who benefits from treatment most, where resources should be spent for future treatments, and why some individuals are treated while others were not. It can provide explanations for treatment effect estimates in a way that pure modeling methods (that do not use matching) cannot. It helps determine what type of additional data must be collected to control for confounding. Interpretability is important: a recent empirical study (Dorie et al., 2019) suggests that CATEs estimated by manually calibrated methods can be better than CATEs estimated by black box models, even though the latter can achieve good performance if they are implemented with careful consideration of the specific application (Hitsch and Misra, 2018). These statements are in agreement with other general observations that interpretable models do not necessarily lead to sacrifices in accuracy because they allow humans to troubleshoot more effectively (Rudin, 2019).

In this work, we propose an approach to matching under the potential outcomes framework with a binary treatment, for datasets with a possibly large number of discrete covariates. Our method (FLAME – Fast, Large-scale, Almost Matching Exactly) creates matches that are *almost-exact*, meaning that it tries to match treatment and control units exactly on important covariates. The main benefits of FLAME are:

- It *learns* a weighted Hamming distance for matching based on a hold-out training set (rather than using a pre-specified distance). In particular, it successively drops covariates, but always retains enough covariates for high quality CATE estimation and balance between treatment and control groups.

- It lends itself naturally to large datasets, even those that are too large to fit in memory. FLAME has two implementations. One implementation uses bit vectors, and is extremely fast for data that has been preprocessed and fits in memory. The other implementation leverages database management systems (e.g., PostgreSQL, 2016), and in particular, highly optimized built-in SQL *group-by* operators that can operate directly on the database, and can handle data sets that are too large to fit in memory. The use of database systems makes the matching algorithm suitable for parallel executions. The database implementation is specialized to be efficient, and only a few lines of carefully constructed SQL code are needed to perform matching.

FLAME improves over current coarsened exact matching, mixed integer programming matching, and network flow methods in that it does not introduce a distance metric on the covariate space *a priori*, instead, it *learns* the distance metric. For this reason, it does

not suffer when irrelevant variables are introduced. It further improves over regression, propensity and their variable selection methods by not forcing a model form on either the treatment or the outcome, instead it matches on covariates directly (nonparametrically). Unlike other nonparametric methods, like black box machine learning approaches, FLAME is interpretable.

By successively dropping covariates to permit matches, FLAME increases bias in order to make predictions of conditional average treatment effect. We can calculate FLAME's bias directly in some cases, and its bias depends directly on how important the dropped covariates are to predicting the output. If only the irrelevant covariates are dropped, the estimates are unbiased. In that case, FLAME's estimates and the gold standard estimates of exact matches are identical.

We discuss FLAME's relationship to prior work (Section 2) and introduce FLAME's framework (Section 3). We present FLAME's algorithm and implementation (Section 4), give theoretical suggestions of FLAME's statistical bias (Section 5), and provide experiments (Section 6).

2. Relationship to Prior Work

Early approaches to matching considered exact matching on covariates but quickly ran into issues of insufficient sample size when the number of covariates was even moderately large. In high dimensions, there simply are not enough treated and control units with exactly the same values of all the covariates. In the 1970's and 1980's, a large literature on different dimension reduction approaches to matching was developed (e.g., Rubin, 1973a,b, 1976; Cochran and Rubin, 1973) with the extreme being work on propensity score matching, which was later extended to work on penalized regression approaches that leverage propensity (Schneeweiss et al., 2009; Rassen and Schneeweiss, 2012; Belloni et al., 2014; Farrell, 2015). The problem with propensity score methods (and other parametric methods) is that they require a proper specification of a model for the treatment assignment probability, which can never be provided in practice. With doubly robust methods, analysts can specify both a model for the outcome and for the propensity score, with only one of the two having to be correct in order to obtain unbiased estimates; however, there is no reason to presume that either of these models would always be specified correctly in practice. These methods can be augmented with variable selection procedures at either or both steps of the estimation procedure, which can improve estimation but leads to further questions of uncertainty quantification (Brookhart et al., 2006; Hahn, 2004).

Further, in these dimension reduction techniques (Schneeweiss et al., 2009), along with more modern neural network-based dimension-reduction techniques, the reduced feature spaces cease to be interpretable with respect to the original input variables, and the matches are no longer meaningful. While these methods are uninterpretable, it has recently been shown (Farrell et al., 2018) that neural networks can achieve asymptotically consistent treatment effect estimates.

An important type of technique, coarsened exact matching (CEM), creates matches that attempt to preserve more covariate information. It bypasses the need to fit complicated propensity score models by coarsening or discretizing covariates in such a way that the newly constructed covariates allow for exact matching (Iacus et al., 2011a,b). This approach is

appealing when there are many continuous covariates that are naturally amenable to binning (Cattaneo and Farrell, 2011). However, when most or all of the covariates are categorical, which is the case we consider in this work, *a priori* coarsening becomes impossible without introducing a calculus on all of the covariates. This can be problematic in high dimensions, and tends to have the same problem as nearest neighbor techniques, which are well-known to perform poorly in high dimensions because they use manually chosen distance metrics. Further, when categorical variables have two levels, coarsening is equivalent to variable selection. In this setting, coarsened exact matching would lead to the same matches as high dimensional propensity score techniques with variable selection.

A similar problem exists with what is called optimal matching in the literature (Rosenbaum, 2016). A distance metric over variables is defined manually, which introduces a calculus on the covariates. That distance metric is used as input to a network flow problem which optimizes match quality. Despite the optimality of the solution network flow problem, the quality of the matches is questionable since, again, it relies on a manually defined distance measure. Network flow problems also cannot directly handle constraints; the user needs to manually manipulate the algorithm in order to obtain desired balance constraints (Zubizarreta, 2012).

The problems with network flow optimization highlight an important concern about non-exact matching methods generally, which is that they implicitly approximate the solution of a hard combinatorial optimization problem. It is possible that high quality match assignments exist, but the standard approximate methods of constructing matches, such as network flow optimization, did not find them. To handle the problem of finding suboptimal match assignments, some newer matching schemes use mixed integer programming, which is a flexible framework that can accommodate linear balance constraints (Zubizarreta, 2012; Zubizarreta et al., 2014; Zubizarreta and Keele, 2017; Resa and Zubizarreta, 2016; Morucci et al., 2018; Noor-E-Alam and Rudin, 2015). However, these methods have two major disadvantages: first they cannot scale to large problems; second, they may be trying to match units on covariates that are not important in any way. In the simulations later, we show how these issues with past work heavily affect practical performance of these methods, whereas FLAME, which learns interpretable distances on the covariates, does not seem to have these problems.

We do not recommend using FLAME on continuous covariates unless there is some substantive knowledge that allows the user to bin the covariates in a way that would not hinder downstream causal estimation. An example of such knowledge is that the outcome and covariates vary smoothly with the continuous variable and that the bins are sufficiently small to model that variation. While our work considers large scale problems with mainly discrete or categorical covariates, its extensions have adapted its main ideas to handle continuous covariates by learning either adaptive stretch metrics (Parikh et al., 2018; Parikh et al., 2019) or adaptive hyperboxes (AHB) (Morucci et al., 2020), both of which use distance metrics that are different than the adaptive Hamming distance that FLAME uses. FLAME can be used alongside these methods, for instance, to eliminate a large number of irrelevant categorical variables to attain a smaller dataset on which MALTS or AHB can operate on both continuous and categorical variables.

3. The FLAME Framework

Suppose $\mathcal{S} = [X, Y, T]$ is the *data table* with n units. X is $n \times d$, Y and T are $n \times 1$, where each row corresponds to a *unit* (called a *tuple* in database management). The columns of X correspond to *variables* (also known as *covariates*, *features*, or, *attributes*). Data must be categorical in order for exact matches to occur with non-zero probability; the match is approximate with binned real-valued data. Treatment assignment variable T takes binary values 1 (for treatment) and 0 (for control). $Y^{(1)}$ and $Y^{(0)}$ are the $n \times 1$ vectors of potential outcomes.

In real-world field studies, researchers tend to collect as many covariates as possible, including irrelevant covariates. Throughout the remaining exposition, we divide the covariates into relevant covariates: X^{REL} and irrelevant covariates: X^{IRR} and make the following standard assumptions: (1) Stable Unit Treatment Value (Rubin, 2005); (2) overlap of support (of the treated population and the control population); (3) Strong ignorability (Rosenbaum and Rubin, 1983): $Y^{(1)}, Y^{(0)} \perp T | X^{REL}$. (4) Further, for irrelevant covariates X^{IRR} we assume that $Y^{(1)}, Y^{(0)} \perp X^{IRR}$, and $T \perp X^{IRR} | X^{REL}$. Based on our assumptions, our estimand is $\mathbb{E}[Y^{(1)} - Y^{(0)} | X^{REL}] = \mathbb{E}[Y^{(1)} - Y^{(0)} | X]$. Within each matched group, we use the difference between the average outcome of the treated units and the average outcome of the control units to estimate CATE given the covariate values associated with that group. *Note that the matching does not produce estimation, it produces a partition of the covariate space, based on which we can estimate CATEs.*

FLAME learns an interpretable weighted Hamming distance on discrete data. This allows it to match on the important covariates, without necessarily matching on unimportant covariates. In the following subsection we discuss the importance of learning a distance metric.

3.1 Why Learn a Distance Metric for Matching?

In exact matching schemes, we would ideally match on as many covariates as possible. However, exact matching on as many covariates as possible leads to a serious issue, namely the *toenail problem*. The toenail problem is where irrelevant covariates dominate the distance metric for matching. Irrelevant covariates are related neither to treatment assignment or outcome, such as length of toenails when considering heart attack outcomes. The researcher typically chooses a distance metric in the high dimensional covariate space (Hamming distance, or squared distance perhaps) in matching without examining outcome variables. The researcher might weigh each covariate equally in the definition of the distance metric. By this choice of distance metric, the irrelevant covariates could dominate the distance measure; adding many covariates that are random noise would make the matches closer to random matches. The matches become essentially meaningless and resulting estimated CATEs are therefore meaningless. The problem becomes worse as more irrelevant covariates are included in the distance metric.

Irrelevant covariates should be irrelevant to the outcome of matching. However, if the distance metric is chosen without consideration of the importance of covariates, in the most extreme case, irrelevant covariates could be chosen adversarially, so that the matching method could produce any given unreasonable match assignment. In Proposition 1 below, we show how any unreasonable match assignment can appear to be reasonable, according

to a fixed (not-learned) Hamming distance, if we are permitted to append a set of irrelevant variables to our dataset.

A *match assignment* (MA) assigns an integer to each unit, and units with the same integer value are in the same *matched group*. A *reasonable match assignment* ($\texttt{MA}_\texttt{r}$) is one in which each unit in a matched group is at most distance $D_{\max}$ from any other unit assigned to the same match group, using, for example, Euclidean distance or Hamming distance. For any two units, with vector of covariates x_1 and x_2, in a matched group created by MA, the matching assignment is reasonable if distance$(x_1, x_2) \leqslant D_{\max}$. Similarly, an *unreasonable match assignment* ($\texttt{MA}_\texttt{un}$) is one in which there exists at least one treatment-control pair, with covariates respectively x_1 and x_2, in the same matched group where $distance(x_1, x_2) > D_{max}$.

Proposition 1 *Consider any matching method "*Match*" that creates reasonable match assignments for dataset* $\{X, T\}$. Match *is constrained so that if treatment unit* x_1 *and control unit* x_2 *obey Hamming distance*$(x_1, x_2) \leqslant D_{max}$*, and there are no other treatment or control units that are able to be matched with* x_1 *and* x_2*, then* x_1 *and* x_2 *will be assigned to the same matched group. Consider any unreasonable match assignment* $\texttt{MA}_\texttt{un}$ *for dataset* $\{X, T\}$*. If distance*$(\cdot, \cdot)$ *is* normalized by the number of covariates*, then it is possible to append a set of covariates* A *to* X *that: (i) can be chosen without knowledge of* X*, (ii) can be chosen without knowledge of* T*, (iii) matching on the appended dataset,* Match$(\{[X, A], T\})$*, will make the match assignment* $\texttt{MA}_\texttt{un}$ *reasonable (which is undesirable).*

Proof Consider unreasonable match x_a and x_b in $\texttt{MA}_\texttt{un}$. Then, create appended feature set A to have more features than X, such that units x_a and x_b would have identical rows of A. These identical rows can be chosen without knowledge of the values within x_a and x_b. This forces x_a and x_b to be matched. This would be repeated for every matched group in $\texttt{MA}_\texttt{un}$. This forces all matches in $\texttt{MA}_\texttt{un}$ to occur. ■

This completely adversarial setting is unlikely to occur in reality, however, situations where matches are disrupted by irrelevant covariates are realistic. The more irrelevant covariates are included, the more the matched groups tend to disintegrate in quality for most matching methods. A reasonable sanity check is that irrelevant covariates should be able to be eliminated automatically by the matching method.

As we discuss below, FLAME does not use a pre-determined distance for matching. It learns a distance for matching from a hold-out training set. In particular, it approximately solves the Full-AME problem formalized in the next subsection.

3.2 Full Almost-Matching-Exactly (Full-AME) Problem

While matching using irrelevant covariates is problematic (as we have shown), matching on too few relevant covariates is also problematic. *We would like to ensure that each unit is matched using at least a set of covariates that is sufficient to predict outcomes well.* Conversely, if a unit is matched using a set of covariates that do not predict outcomes sufficiently well, we would not trust the results from its matched group. Let us formalize the problem of matching each unit on a set of covariates that together predict outcomes well.

We will use $\boldsymbol{\theta} \in \{0,1\}^d$ to denote the variable selection indicator vector for a subset of covariates to match on. Throughout the following discussion, we consider a unit to be a triplet of (covariate value x, observed outcome y, treatment indicator t), unless otherwise stated. Given dataset $\mathcal{S}$, define the *matched group* for unit i with respect to covariates selected by $\boldsymbol{\theta}$ as the units in $\mathcal{S}$ that match i exactly on the covariates $\boldsymbol{\theta}$:

$$\mathcal{MG}_i(\boldsymbol{\theta}, \mathcal{S}) = \{i' \in \mathcal{S} : \mathbf{x}_{i'} \circ \boldsymbol{\theta} = \mathbf{x}_i \circ \boldsymbol{\theta}\}.$$

Here we use $\mathbf{x}_i$ and $\mathbf{x}_{i'}$ to denote the covariate values for unit i and i' respectively. Taking the union over all units, we also define the *matched units* for a selection indicator $\boldsymbol{\theta}$ as the collection of units that are matched on covariates defined by selection indicator $\boldsymbol{\theta}$:

$$\mathcal{MG}(\boldsymbol{\theta}, \mathcal{S}) = \{i \in \mathcal{S} : \exists i \neq i' \quad s.t. \quad \mathbf{x}_{i'} \circ \boldsymbol{\theta} = \mathbf{x}_i \circ \boldsymbol{\theta}\}. \tag{1}$$

The value of a set of covariates $\boldsymbol{\theta}$ is determined by how well these covariates can be used together to predict outcomes. Specifically, the prediction error $\texttt{PE}_{\mathcal{F}_k}(\boldsymbol{\theta})$ is defined with respect to a class of functions $\mathcal{F}_k := \{f : \{0,1\}^k \to \mathbb{R}\}$ $(1 \leqslant k \leqslant d)$ as:

$$\texttt{PE}_{\mathcal{F}_{\|\boldsymbol{\theta}\|_0}}(\boldsymbol{\theta}) = \min_{f^{(1)} \in \mathcal{F}_{\|\boldsymbol{\theta}\|_0}} \mathbb{E}[(f^{(1)}(\mathbf{x} \circ \boldsymbol{\theta}) - y)^2 | t = 1] + \min_{f^{(0)} \in \mathcal{F}_{\|\boldsymbol{\theta}\|_0}} \mathbb{E}[(f^{(0)}(\mathbf{x} \circ \boldsymbol{\theta}) - y)^2 | t = 0], \tag{2}$$

where the expectation is taken over $\mathbf{x}$ and y, when $\|\cdot\|_0$ is the count of nonzero elements of the vector. That is, $\texttt{PE}_{\mathcal{F}_{\|\boldsymbol{\theta}\|_0}}$ is the smallest prediction error we can get on both treatment and control populations using the features specified by $\boldsymbol{\theta}$. Consider a separate training dataset $\mathcal{S}^{tr}$. Let $\mathcal{S}_0^{tr}$ be the subset (of $\mathcal{S}^{tr}$) of control units (X^{tr}, Y^{tr}) with $T^{tr} = 0$, and let $\mathcal{S}_1^{tr}$ be the subset (of $\mathcal{S}^{tr}$) of treated units (X^{tr}, Y^{tr}) with $T^{tr} = 1$. The empirical counterpart of $\texttt{PE}_{\mathcal{F}_{\|\boldsymbol{\theta}\|_0}}$ is defined as:

$$\begin{aligned}\hat{\texttt{PE}}_{\mathcal{F}_{\|\boldsymbol{\theta}\|_0}}(\boldsymbol{\theta}, \mathcal{S}^{tr}) = &\min_{f^{(1)} \in \mathcal{F}_{\|\boldsymbol{\theta}\|_0}} \frac{1}{|\mathcal{S}_1^{tr}|} \sum_{(\mathbf{x}_i, y_i) \in \mathcal{S}_1^{tr}} (f^{(1)}(\mathbf{x}_i \circ \boldsymbol{\theta}) - y_i)^2 \\ &+ \min_{f^{(0)} \in \mathcal{F}_{\|\boldsymbol{\theta}\|_0}} \frac{1}{|\mathcal{S}_0^{tr}|} \sum_{(\mathbf{x}_i, y_i) \in \mathcal{S}_0^{tr}} (f^{(0)}(\mathbf{x}_i \circ \boldsymbol{\theta}) - y_i)^2.\end{aligned} \tag{3}$$

Given a matching dataset $\mathcal{S}^{ma}$ and a training dataset $\mathcal{S}^{tr}$, the best selection indicator we could achieve for a nontrivial matched group that contains treatment unit i would be:

$$\boldsymbol{\theta}^*_{i,\mathcal{S}^{ma}} \in \arg\min_{\boldsymbol{\theta}} \hat{\texttt{PE}}_{\mathcal{F}_{\|\boldsymbol{\theta}\|_0}}(\boldsymbol{\theta}, \mathcal{S}^{tr}) \text{ s.t. } \exists \ell \in \mathcal{MG}_i(\boldsymbol{\theta}, \mathcal{S}^{ma}) \text{ s.t. } t_\ell = 0,$$

where t_ℓ is the treatment indicator of unit ℓ. This constraint says that the matched group contains at least one control unit. The covariates selected by $\boldsymbol{\theta}^*_{i,\mathcal{S}^{ma}}$ are those that predict the outcome best, provided that at least one control unit has the same exact covariate values as i on the covariates selected by $\boldsymbol{\theta}^*_{i,\mathcal{S}^{ma}}$.

The *main matched group* for i is then defined as $\mathcal{MG}_i(\boldsymbol{\theta}^*_{i,\mathcal{S}^{ma}}, \mathcal{S}^{ma})$. *The goal of the Full-AME problem is to calculate the main matched group* $\mathcal{MG}_i(\boldsymbol{\theta}^*_{i,\mathcal{S}^{ma}}, \mathcal{S}^{ma})$ *for as many units* i *as possible.* Once the problem is solved, the main matched groups can be used to

estimate treatment effects, by considering the difference in outcomes between treatment and control units in each group, and possibly smoothing the estimates from the matched groups if desired, to prevent overfitting of treatment effect estimates.

We now present a worst-case bound on the bias induced by matching units in an almost-exact framework, and using the created matched groups to estimate CATEs. We will see subsequently that the FLAME procedure directly targets minimization of this bias.

3.3 Bias Bound in the Full-AME Problem

If we do not match on all relevant covariates, a bias is induced on the treatment effect estimates. As shown before, solving the Full-AME problem ensures that this bias is as small as possible, and zero if the covariates excluded are all irrelevant. Here we present a simple worst-case bound on the in-sample estimation bias when a CATE is estimated with units matched according to a chosen subset of covariates (defined by $\boldsymbol{\theta}$). This bound is worst-case in that it holds for any subset of relevant covariates. This implies that the bias resulting from Full-AME will be much smaller than the bound given here in most cases.

Proposition 2 *Let $g^{(1)}(\mathbf{x})$ and $g^{(0)}(\mathbf{x})$ represent nonrandom potential outcomes at arbitrarily chosen covariate value $\mathbf{x} \in \{0,1\}^p$, so that $g^{(1)}(\mathbf{x}_i) := y_i^{(1)}$ and $g^{(0)}(\mathbf{x}_i) := y_i^{(0)}$ are the potential outcomes in a sample of n units. Fix a value of the covariates $\mathbf{x} \in \{0,1\}^d$, and a value of $\boldsymbol{\theta} \in \{0,1\}^d$. Let $\mathcal{MG}(\mathbf{x}, \boldsymbol{\theta}, \mathcal{S}^{ma}) = \{i \in \mathcal{S}^{ma} : \mathbf{x}_i \circ \boldsymbol{\theta} = \mathbf{x} \circ \boldsymbol{\theta}\}$ be the set of units in the sample that have value equal to $\mathbf{x}$ on the covariates selected by $\boldsymbol{\theta}$. Define additionally: $n_t(\mathbf{x}, \boldsymbol{\theta}, \mathcal{S}^{ma}) = \sum_{i \in \mathcal{MG}(\mathbf{x}, \boldsymbol{\theta}, \mathcal{S}^{ma})} T_i$ and $n_c(\mathbf{x}, \boldsymbol{\theta}, \mathcal{S}^{ma}) = \sum_{i \in \mathcal{MG}(\mathbf{x}, \boldsymbol{\theta}, \mathcal{S}^{ma})} (1 - T_i)$. Let $\tau(\mathbf{x}) = g^{(1)}(\mathbf{x}) - g^{(0)}(\mathbf{x})$ be the CATE estimand of interest. For a weighted Hamming distance with positive weight vector $\mathbf{w}$ of length p, and $0 < \|\mathbf{w}\|_2 < \infty$, define $M = \max_{\substack{\mathbf{x}, \mathbf{x}' \in \{0,1\}^p \\ t \in \{0,1\}}} \frac{|g^{(t)}(\mathbf{x}') - g^{(t)}(\mathbf{x})|}{\mathbf{w}^T \mathbf{1}_{(\mathbf{x}' \neq \mathbf{x})}}$ and assume $M < \infty$. We have, for any $\boldsymbol{\theta} \in \{0,1\}^p$:*

$$\left| \frac{1}{n_t(\mathbf{x}, \boldsymbol{\theta}, \mathcal{S}^{ma})} \sum_{i \in \mathcal{MG}(\mathbf{x}, \boldsymbol{\theta}, \mathcal{S}^{ma})} Y_i T_i - \frac{1}{n_c(\mathbf{x}, \boldsymbol{\theta}, \mathcal{S}^{ma})} \sum_{i \in \mathcal{MG}(\mathbf{x}, \boldsymbol{\theta}, \mathcal{S}^{ma})} Y_i (1 - T_i) - \tau(\mathbf{x}) \right| \leqslant 2M \mathbf{w}^T (\mathbf{1} - \boldsymbol{\theta}). \quad (4)$$

As asserted by Proposition 2, we should select $\boldsymbol{\theta}$ to minimize $\mathbf{w}^T(\mathbf{1} - \boldsymbol{\theta})$ in order to minimize FLAME's bias. In real problems, we should think about $\mathbf{w}$ as a non-uniform vector that has some small entries. FLAME would tend to remove those entries so that the bias is minimized for the remaining covariates that are used for matching. A proof of Proposition 2 can be found in Appendix A. This bound provides guidance in designing the FLAME procedure as it suggests that the amount of bias, in estimating treatment effects with almost-exact matching, depends heavily on $\boldsymbol{\theta}$: FLAME will try to match on sets of covariates that minimize such bias.

3.4 FLAME's Backward Procedure to Full-AME

While solving the Full-AME problem is ideal, computational challenges prevent its usage on large datasets. We thus resort to an approximated solution. In determining which covariates are important, FLAME leverages the classic Efroymson's backward stepwise regression

(Efroymson, 1960; McCornack, 1970), which is widely adopted by practitioners in regular regression tasks. This procedure starts with all covariates in the model, and greedily removes one covariate at a time, according to the PE and BF criteria, defined below. By using this backward procedure, FLAME is able to strike a good balance between using more units and using more covariates. As evidenced in Section 6, this approximate solution outperforms existing methods on tasks of interest.

Specifically, FLAME performs the following procedure: for iteration $i = 1, \cdots, d$, FLAME selects a set of covariates indicated by $\boldsymbol{\theta}^i$ according to a certain criterion, and stops when a stopping condition is met. We restrict FLAME to perform a greedy selection, namely we require $\boldsymbol{\theta}^i \succeq \boldsymbol{\theta}^{i+1}$ for all i. Here $\succeq$ is the element-wise comparison. During this procedure, FLAME produces a sequence of selection indicators $\{\boldsymbol{\theta}^0, \boldsymbol{\theta}^1, \cdots, \boldsymbol{\theta}^d\}$ ($\boldsymbol{\theta}^i \in \{0,1\}^d$), where $\boldsymbol{\theta}^i$ denotes the selection indicator on iteration i.

The Prediction Error (PE) and Balancing Factor (BF) Criteria

As stated in Proposition 2, the bias of almost-matching-exactly estimates depends on which covariates are selected for matching: matching on irrelevant covariates at the expense of relevant ones will increase bias. FLAME uses the PE Criterion (Eq. 2, 3) to choose which covariates should be considered irrelevant and ignored when matching. FLAME further encourages larger and more balanced matched groups by using the Balancing Factor (BF). The BF is defined with respect to the selection indicators $\boldsymbol{\theta}$, and can be computed as follows:

$$\texttt{BF}(\mathcal{MG}(\boldsymbol{\theta}, \mathcal{S}^{ma})) = \frac{\#\text{ control in } \mathcal{MG}(\boldsymbol{\theta}, \mathcal{S}^{ma})}{\#\text{ available control}} + \frac{\#\text{ treated in } \mathcal{MG}(\boldsymbol{\theta}, \mathcal{S}^{ma})}{\#\text{ available treated}}. \tag{5}$$

Maximizing BF encourages a large fraction of both treatment and control units to be used for the matched groups. This implies that more units would be matched in earlier iterations; as we know, it is better to match units in earlier iterations because the matches made in earlier iterations depend on more variables and are thus higher quality.

FLAME's Procedure

FLAME's backward selection criterion is a combination of PE and BF. More specifically, at each iteration $i = 0, 1, 2, \cdots, p$, given a training dataset $\mathcal{S}^{tr}$ and matching dataset $\mathcal{S}^{ma}$, FLAME finds a selection indicator such that

$$\begin{aligned} \boldsymbol{\theta}^i \in \arg\max_{\boldsymbol{\theta}} & \left[-\hat{\texttt{PE}}_{\mathcal{F}_{\|\boldsymbol{\theta}\|_0}}(\boldsymbol{\theta}, \mathcal{S}^{tr}) + C \cdot \texttt{BF}(\mathcal{MG}(\boldsymbol{\theta}, \mathcal{S}^{ma})) \right] \\ & s.t.\ \ \boldsymbol{\theta}^i \in \{0,1\}^d, \|\boldsymbol{\theta}^i\|_0 = d - i, \boldsymbol{\theta}^i \preceq \boldsymbol{\theta}^{i-1}. \end{aligned} \tag{6}$$

Here $\preceq$ is for element-wise comparison, and C is a hyperparameter that trades off PE and BF.

Similarly to a general backward method, FLAME retains a threshold that prevents the algorithm from eliminating too many covariates. That is, FLAME retains a stopping criterion that ensures *every matched group is matched exactly on a set of covariates that, together, can predict outcomes well* (sacrificing no more than ϵ training accuracy on the hold-out training set). Let us first define feasible covariate sets for treatment unit u as

Algorithm 1 : FLAME Algorithm

Inputs Input data $\mathcal{S}^{ma} = (X, Y, T)$ for matching; training set $\mathcal{S}^{tr} = (X^{tr}, Y^{tr}, T^{tr})$; model classes $\mathcal{F}_1, \mathcal{F}_2, \cdots, \mathcal{F}_d$; stopping threshold ϵ; tradeoff parameter C.

Outputs A sequence of selection indicators $\boldsymbol{\theta}^0, \cdots, \boldsymbol{\theta}^d$, and a set of matched groups $\{\mathcal{MG}(\boldsymbol{\theta}^l, \mathcal{S}^l)\}_{l \geqslant 1}$. ▷ *$\mathcal{S}^l$ is defined in the algorithm.*

1: Initialize $\mathcal{S}^0 = \mathcal{S}^{ma} = (X, Y, T), \boldsymbol{\theta}^0 = \mathbf{1}_{d\times 1}, l = 1, run = True$.
▷ *l is the index for iterations.*

2: Compute exact matched groups $\mathcal{MG}(\boldsymbol{\theta}^0, \mathcal{S}^0)$ as defined in (1).
▷ *The detailed implementation is in Section 4.*

3: **while** $run = True$ **do**

4: Compute $\boldsymbol{\theta}^l$ using (6) on training set $\mathcal{S}^{tr}$, using $\mathcal{F}_{d-l}$ and tradeoff parameter C.
▷ *Determine which covariates to match on for this iteration.*

5: Compute matched groups $\mathcal{MG}(\boldsymbol{\theta}^{l-1}, \mathcal{S}^{l-1})$ as defined in (1).
▷ *The detailed implementation is in Section 4.*

6: $\mathcal{S}^l = \mathcal{S}^{l-1} \backslash \mathcal{MG}(\boldsymbol{\theta}^{l-1}, \mathcal{S}^{l-1})$. ▷ *These matched units are done.*

7: **if** $\hat{\mathtt{PE}}_{\mathcal{F}_{d-l}}(\boldsymbol{\theta}^l, \mathcal{S}^{tr}) > \hat{\mathtt{PE}}_{\mathcal{F}_d}(\mathbf{1}_{d\times 1}, \mathcal{S}^{tr}) + \epsilon$ OR $\mathcal{S}^l = \varnothing$ **then**

8: $run = False$ ▷ *Prediction error is too high to continue matching.*

9: $l = l + 1$

10: **Output** $\{\boldsymbol{\theta}^l, \mathcal{MG}(\boldsymbol{\theta}^l, \mathcal{S}^l)\}_{l \geqslant 1}$.

those covariate sets that predict well on the hold-out training set (and lead to a valid matched group for u):

$$\bar{\boldsymbol{\theta}}^{\text{feasible}} = \left\{ \boldsymbol{\theta} \in \{0,1\}^d : \mathtt{PE}_{\mathcal{F}_{\|\boldsymbol{\theta}\|_0}}(\boldsymbol{\theta}, \mathcal{S}^{tr}) \leqslant \mathtt{PE}_{\mathcal{F}_d}(\mathbf{1}_{d\times 1}, \mathcal{S}^{tr}) + \epsilon \right\}.$$

To incorporate this feasible set constraint, we simply add the requirement $\boldsymbol{\theta}^i \in \bar{\boldsymbol{\theta}}^{\text{feasible}}$ to (6).

In Algorithm 1, we summarize the FLAME procedure. The core task of the algorithm is to determine a sequence of variable selection indicators $\{\boldsymbol{\theta}^0, \boldsymbol{\theta}^1, \cdots, \boldsymbol{\theta}^d\}$. Matches can be easily determined given the sequence of variable selection indicators. In order to formally state the matching procedure, we use $\mathcal{S}^{ma} = (X, Y, T)$ to denote the triplet of (covariates, outcomes, treatment indicators) of the units for matching, $\mathcal{S}^{tr} = (X^{tr}, Y^{tr}, T^{tr})$ to denote the same for units in the training set.

The algorithm starts by initializing the selection indicator to include all covariates (Line 1). Exact matches are made when possible at Line 2. At each iteration of the while loop, FLAME computes the selection indicator for that iteration by minimizing the objective defined in (6) (Line 4) and matches units exactly on the covariates selected by the newly found selector (Line 5). Matched units are then excluded from the leftover data (Line 6), and the procedure is repeated until either prediction error increases too much from removing an additional covariate, or all the data are matched (Line 7).

FLAME's implementation performs matching without replacement, but can be adjusted to perform matching with replacement.

An estimate of the ATE is straightforward to compute once the treatment effects in each group (conditional average treatment effects – CATEs) are computed as differences in out-

come means between treatment and control units. For better CATE estimates, smoothing (e.g., regression) can be performed on the raw matched CATE estimates.

4. Implementing Matched Groups in FLAME

The workhorse behind FLAME's database implementation is its matching subroutine of finding $\mathcal{MG}$'s. We implement this procedure using the following two methods.

4.1 Implementation using Database (SQL) Queries

Exact matching is highly related to the `GROUP BY` operator used in database (SQL) queries, which computes aggregate functions (sum, count, etc) on groups of rows in a two-dimensional table having the same values of a subset of columns specified in the query. SQL queries can be run on any standard commercial or open-source relational database management system (*e.g.*, Microsoft SQL Server, Oracle, IBM DB2, Postgres, etc.). These database systems are highly optimized and robust for SQL queries, can be easily integrated with other languages (we used python and SQL), and scale to datasets with a large number of rows (n) or columns (p) that may not fit in the available main memory. In addition, SQL queries *declaratively* specify complex operations (we only specify 'what' we want to achieve, like matched groups on the same values of variables, and not 'how' to achieve them, *i.e.*, no algorithm has to be specified), and are therefore succinct. In our work, SQL enables us to execute a matching step in a single query as we discuss below. In this implementation, we keep track of matched units globally by keeping an extra column called `is_matched` in the input database $\mathcal{S}^{ma}$ (containing the data triplet $\mathcal{S}^{ma} = (X, Y, T)$ described in Section 3). For every unit, the value of `is_matched` = ℓ if the unit was matched in a valid main matched group with at least one treated and one control unit in iteration ℓ of Algorithm 1, and `is_matched = 0` if the unit is still unmatched. For notational simplicity, let $A_1, \cdots, A_k$ be the names of the covariates selected by $\boldsymbol{\theta}$, and $S = \mathcal{S}^{ma}$. The SQL query for the matching procedure without replacement on dataset `S` is given below.

```
WITH tempgroups AS
   (SELECT A_1, A_2, ..., A_k
                --(matched groups will be identified by their covariate values)
     FROM S
     WHERE  is_matched = 0
                --(use data that are not yet matched)
     GROUP BY A_1, A_2, ..., A_k
                --(create matched groups with identical values of covariates)
     HAVING SUM(T) > 0 AND SUM(T) < COUNT(*)
                --(groups have at least one treated and one control unit)
   )
UPDATE S
SET is_matched = ℓ
WHERE is_matched = 0 AND
   EXISTS
      (SELECT Q.A_1, Q.A_2, ..., Q.A_k
       FROM tempgroups AS Q
                --(set of covariate values for valid groups)
       WHERE  Q.A_1 = S.A_1 AND Q.A_2 = S.A_2 AND ... AND Q.A_k = S.A_k)
```

The *WITH clause* computes a temporary relation *tempgroups* that computes the combination of values of the covariates forming 'valid groups' (*i.e.*, groups with at least one treatment and at least one control unit) on unmatched units. The *HAVING clause* of the SQL query discards groups that do not satisfy this property – since treatment T takes binary values (0 or 1), for any valid group, the sum of T values will be strictly greater than 0 and strictly less than the total number of units in the group. Then we update the population table `S`, where the values of the covariates of the existing units match with those of a valid group in *tempgroups*. Several optimizations of this basic query are possible and are used in our implementation. Setting the `is_matched` value to level ℓ (instead of a constant value like 1) helps us compute the CATE for each matched group efficiently. (Note that for each value of ℓ, there might be more than one matched group, with different values of covariates being used for matching.)

4.2 Implementation using Bit Vectors

In an alternative bit-vector implementation, for binary covariates, we encode the combination of the to-match-on covariates of unit i as a binary number b_i; we also encode the to-match-on covariates, appended with the treatment indicator as the least significant digit, to form a binary number b_i^+. A unit i with ordered to-match-on variable values $(a_{i,d}, a_{i,d-1}, \cdots, a_{i,1})$ with $a_{i,k} \in \{0,1\}$ and treatment indicator $t_i \in \{0,1\}$ is associated with numbers $b_i = \sum_{k=1}^{d} a_{i,k} 2^{k-1}$ and $b_i^+ = \sum_{k=1}^{d} a_{i,k} 2^k + t_i$. Two units i and j have the same covariate values if and only if $b_i = b_j$. For each unit i, we count how many times b_i and b_i^+ appear (in the whole dataset), and denote the counts as c_i and c_i^+ respectively. A unit i is matched if and only if $c_i \neq c_i^+$, since the two counts differ *if and only if* the same b_i appears both as a treated instance and a control instance. This property is summarized in Proposition 3. For non-binary categorical data, if the k-th covariate is $h_{(k)}$-ary, we first rearrange the d covariates such that $h_{(k)} \leqslant h_{(k+1)}$ for all $1 \leqslant k \leqslant d-1$. Thus each unit $(a_{i,d}, a_{i,d-1}, \cdots, a_{i,1})$ uniquely represents the number $\sum_{k=1}^{d} a_{i,k} h_{(k)}^{k-1}$. From here we can apply the above method for the binary case.

Proposition 3 *A unit u is matched if and only if $c_u \neq c_u^+$, since the two counts b_u and b_u^+ differ if and only if the same combination of covariate values appear both as a treated unit and a control unit.*

An example of this procedure is illustrated in Table 1. We assume in this population the first variable is binary and the second variable is ternary. In this example, the number b_1 for the first unit is $0 \times 2^0 + 2 \times 3^1 = 6$; the number b_1^+ including its treatment indicator is $0 + 0 \times 2^1 + 2 \times 3^2 = 18$. Similarly, we can compute all the numbers b_u, b_u^+, c_u, c_u^+, and the matching results are listed in the last column in Table 1.

4.3 Comparison

The bit vector implementation typically outperforms the SQL implementation when the data fits in memory, but for large data and more covariates, SQL performs better. A detailed comparison is deferred to Section 6. Another limitation of the bit vector implementation is that the magnitude of the numeric representation grows exponentially and can cause

first variable	second variable	T	b_i	b_i^+	c_i	c_i^+	is matched?
0	2	0	6	18	1	1	No
1	1	0	4	11	2	1	Yes
1	0	1	1	3	1	1	No
1	1	1	4	12	2	1	Yes

Table 1: Example population table illustrating the *bit-vector* implementation. Here the second unit and the fourth unit are matched to each other while the first and third units are left unmatched.

overflow problems. This is another reason to use FLAME-db when the number of covariates or total number of categories is large.

5. Bias Calculations

In this section we provide two types of bias calculation. One (Theorems 4, 5, 6) provides a bias calculation with oracle covariate importance information, while the other (Proposition 7) serves as the empirical counterpart of Proposition 2.

5.1 Exact Bias Computations for Oracle FLAME

FLAME trades off statistical bias for computational speed. Here we provide insight into the bias that an oracle version of FLAME without replacement (defined below) induces when estimating heterogeneous causal effects as well as its unbiased performance when estimating an average causal effect. To evaluate the theoretical behavior of the algorithm we consider the outcome model

$$y_i = \alpha_0 + \sum_{j=1}^{d} \alpha_j x_{ij} + \beta_0 T_i + T_i \sum_{j=1}^{d} \beta_j x_{ij}$$

as a data generation process that corresponds to a treatment effect β_j being associated with every covariate x_{ij} for individual i. Here y_i is the observed outcome and T_i is the observed treatment indicator. We are interested in the bias of FLAME for this simple non-noisy outcome model. We define the Oracle FLAME algorithm as a simplified version of Algorithm 1 that knows the correct order of importance for the covariates. Without loss of generality, let that order be $d, d-1, \ldots, 1$. Given this ordering, we can directly compute the bias of the FLAME estimates for various combinations of covariates.

To compute the overall bias of Oracle FLAME, in theory we would enumerate all possible covariate allocations and run Oracle FLAME on each of those. For example, in the two-covariate setting with possible attribute values 0 and 1, there are $2^2 \times 2^2 = 16$ possible covariate allocations for treatment and for control units leading to $16 \times 16 = 256$ total possible allocations. Since we are interested only in the bias induced by the algorithm itself, we consider only treatment-control allocations where our procedure yields an estimate for each covariate combination. In cases where we do not have an estimate of treatment effect for each covariate combination, we cannot calculate the bias of the algorithm for any distribution that has support over the full covariate space; Oracle FLAME's bias estimates would not be defined on part of the covariate space in these cases. Note that this is

different than the standard overlap assumption made in most theory for causal inference as this calculates the bias when there is overlap at *some point* during the FLAME procedure.[1] For example, in the two covariate setting, the allocation of a total of one treated and one control unit with both having $x_1 = x_2 = 0$ would not be considered in our bias calculation since the exact matching procedure would obtain an estimate for that covariate combination only and not for the other three. An allocation consists of a set of covariate values, and treatment indicators. (The number of units in each covariate bin does not matter, what matters is that there is at least one treatment and one control in the bin, so that we can compute the bin's treatment effect.) For instance, one of the valid allocations would have at least one treatment and one control unit in each bin (a bin is a combination of covariates). Another valid allocation would have treatment and control units in most (but not all) bins, but when the bins are collapsed according to Oracle FLAME, each bin still receives an estimate. We perform these computations for two and three binary covariates and use the results to provide intuition for when we have arbitrarily many covariates. The main results are as follows.

Theorem 4 (Two covariates) *(i) There are 59 valid allocations. (ii) Under a uniform distribution over valid allocations, the biases (taking expectation over the allocations) are given by:*

$$\begin{aligned} bias &= (expected\ TE\ under\ FLAME) - (actual\ TE) \\ &= \begin{array}{c} \\ x_1 = 0 \\ x_1 = 1 \end{array} \begin{array}{c} \begin{array}{cc} x_2 = 0 & \quad x_2 = 1 \end{array} \\ \begin{pmatrix} \frac{20\beta_1 + 41/2\beta_2}{59} & \frac{20\beta_1 - 41/2\beta_2}{59} \\ \frac{-20\beta_1 + 41/2\beta_2}{59} & \frac{-20\beta_1 - 41/2\beta_2}{59} \end{pmatrix}. \end{array} \end{aligned}$$

Theorem 5 (Three covariates) *(i) There are 38070 valid allocations. (ii) Under a uniform distribution over valid allocations the biases (taking expectation over the allocations) are given by:*

$$\begin{array}{c} \\ x_1 \\ \\ \end{array} \begin{array}{c} \begin{array}{c} x_2 = 0 \end{array} \\ \begin{array}{cc} 0 & \\ 1 & \end{array} \begin{pmatrix} \frac{(5976+34/105)\beta_1 + (7854+61/210)\beta_2 + 11658\beta_3}{38070} \\ \frac{-(5976+34/105)\beta_1 + (7854+61/210)\beta_2 + 11658\beta_3}{38070} \end{pmatrix} \end{array}, x_3 = 0$$

$$\begin{array}{c} \\ x_1 \\ \\ \end{array} \begin{array}{c} \begin{array}{c} x_2 = 1 \end{array} \\ \begin{array}{cc} 0 & \\ 1 & \end{array} \begin{pmatrix} \frac{(12755+6/7)\beta_1 - (16513+4/21)\beta_2 + 19035\beta_3}{38070} \\ \frac{-(12755+6/7)\beta_1 - (16513+4/21)\beta_2 + 19035\beta_3}{38070} \end{pmatrix} \end{array}, x_3 = 0$$

$$\begin{array}{c} \\ x_1 \\ \\ \end{array} \begin{array}{c} \begin{array}{c} x_2 = 0 \end{array} \\ \begin{array}{cc} 0 & \\ 1 & \end{array} \begin{pmatrix} \frac{(12755+6/7)\beta_1 + (16513+4/21)\beta_2 - 19035\beta_3}{38070} \\ \frac{-(12755+6/7)\beta_1 + (16513+4/21)\beta_2 - 19035\beta_3}{38070} \end{pmatrix} \end{array}, x_3 = 1$$

1. We note that under the standard overlap assumptions, as those used in Section 5.2, all units would be matched in the first step.

$$\begin{array}{cc} & x_2 = 1 \\ x_1 & \begin{array}{c} 0 \\ 1 \end{array} \begin{pmatrix} \frac{(5976+34/105)\beta_1-(7854+61/210)\beta_2-11658\beta_3}{38070} \\ \frac{-(5976+34/105)\beta_1-(7854+61/210)\beta_2-11658\beta_3}{38070} \end{pmatrix} \end{array}, \; x_3 = 1.$$

A proof is available in the appendix.

As these theorem show, the FLAME estimates are biased only by fractions of the treatment effects associated with the covariates (β_j for $j > 0$) rather than any baseline information (the α's) or a universal treatment effect (β_0). This is an appealing quality as it suggests that *rare covariates that have large effects on baseline outcomes are unlikely to have undue influence on the bias of the treatment effect estimates.* This result can be made more concrete for an arbitrary number of covariates:

Theorem 6 *The bias of estimating heterogeneous causal effects due to Oracle FLAME is not a function of* $\alpha_0, \ldots, \alpha_d$.

A proof of this theorem is available in the appendix.

Theorems 4, 5, and 6 further mean that in cases where the causal effect is homogeneous (that is, $\beta_j = 0$ for $j > 0$), this effect can be estimated without any bias. In that sense, FLAME provides an advantage over matching methods that target average treatment effect (such as propensity score matching) since FLAME handles potential heterogeneity at no additional cost.

5.2 Empirical Bound for FLAME Bias

In this section, we discuss an empirical counterpart to Proposition 2 that provides a bias bound in terms of the `PE` (and `BF`) values during the FLAME procedure (Algorithm 1).

Proposition 7 *Let* $p_0(X)$ *and* $p_1(X)$ *be the covariate density functions for the control group and the treated group. Assume*[2] *that for any* $k \in \{1, 2, \cdots, d\}$, *there exists a constant* $\lambda_k > 0$ *such that the marginals* $p_0\left(X \circ \boldsymbol{\theta} = \mathbf{x} \circ \boldsymbol{\theta}\right) \geqslant \lambda_k$ *and* $p_1\left(X \circ \boldsymbol{\theta} = \mathbf{x} \circ \boldsymbol{\theta}\right) \geqslant \lambda_k$ *for any* $\mathbf{x}$ *and* $\boldsymbol{\theta}$ *with* $\|\boldsymbol{\theta}\|_0 = k$. *For a dataset* $\mathcal{S}^{ma} = (X, Y, T)$, *let* $\mathcal{S}_0^{ma}$ *be the subset of control units:* (X, Y) *with* $T = 0$, *and let* $\mathcal{S}_1^{ma}$ *be the subset of treated units* (X, Y) *with* $T = 1$. *For a dataset* $\mathcal{S}^{ma}$, *we overload notation by using* $\mathbf{x} \in \mathcal{S}^{ma}$ *and* $(\mathbf{x}, y) \in \mathcal{S}^{ma}$ *to refer to a covariate record in* $\mathcal{S}^{ma}$ *and a (covariate,outcome) pair record in* $\mathcal{S}^{ma}$. *Also, for a dataset* $\mathcal{S}^{ma}$ *(or* $\mathcal{S}_0^{ma}$, $\mathcal{S}_1^{ma}$*), let*

$$n(\mathbf{x}, \boldsymbol{\theta}, \mathcal{S}^{ma}) := \sum_{\mathbf{x}_i \in \mathcal{S}^{ma}} \mathbb{1}[\mathbf{x}_i \circ \boldsymbol{\theta} = \mathbf{x} \circ \boldsymbol{\theta}]$$

be the number of points that agree with $\mathbf{x}$ *on the covariates defined by* $\boldsymbol{\theta}$. *For any* $f \in \mathcal{F}_{\|\boldsymbol{\theta}\|_0}$, $t \in \{0, 1\}$, *and datasets* $\mathcal{S}_0^{ma}$, $\mathcal{S}_1^{ma}$, *write*

$$\hat{\epsilon}^{\max}(\mathcal{S}_t^{ma}, f, \boldsymbol{\theta}) := \max_{(\mathbf{x}, y) \in \mathcal{S}_t^{ma}} |f(\mathbf{x} \circ \boldsymbol{\theta}) - y| \,.$$

2. Note that this is equivalent to the standard assumption that the density of the covariates is bounded away from 0 (e.g., Imbens and Rubin, 2015).

Let $\hat{g}(\mathbf{x}, \boldsymbol{\theta}, \mathcal{S}_1^{ma})$ *(resp.* $\hat{g}(\mathbf{x}, \boldsymbol{\theta}, \mathcal{S}_0^{ma})$*) be the plain estimator of the treated (resp. control) outcome for covariate values* $\mathbf{x} \circ \boldsymbol{\theta}$*:*

$$\hat{g}(\mathbf{x}, \boldsymbol{\theta}, \mathcal{S}_1^{ma}) := \frac{1}{|n(x, \boldsymbol{\theta}, \mathcal{S}_1^{ma})|} \sum_{(\mathbf{x}_i, y_i) \in \mathcal{S}_1^{ma}} y_i \mathbb{1}[\mathbf{x}_i \circ \boldsymbol{\theta} = \mathbf{x} \circ \boldsymbol{\theta}], \text{ and}$$

$$\hat{g}(\mathbf{x}, \boldsymbol{\theta}, \mathcal{S}_0^{ma}) := \frac{1}{|n(x, \boldsymbol{\theta}, \mathcal{S}_0^{ma})|} \sum_{(\mathbf{x}_i, y_i) \in \mathcal{S}_0^{ma}} y_i \mathbb{1}[\mathbf{x}_i \circ \boldsymbol{\theta} = \mathbf{x} \circ \boldsymbol{\theta}].$$

Assume that the outcomes are noiseless and there are r *relevant covariates. Then for any fixed* $\boldsymbol{\theta}$*,* $f^{(0)}, f^{(1)} \in \mathcal{F}_{\|\boldsymbol{\theta}\|_0}$*, with probability at least*

$$1 - \left(1 - \lambda_{\|\boldsymbol{\theta}\|_0 + r}\right)^{|\mathcal{S}_0^{tr}|} - \left(1 - \lambda_{\|\boldsymbol{\theta}\|_0 + r}\right)^{|\mathcal{S}_1^{tr}|} - \left(1 - \lambda_{\|\boldsymbol{\theta}\|_0 + r}\right)^{|\mathcal{MG}(\boldsymbol{\theta}, \mathcal{S}_0^{ma})|} - \left(1 - \lambda_{\|\boldsymbol{\theta}\|_0 + r}\right)^{|\mathcal{MG}(\boldsymbol{\theta}, \mathcal{S}_1^{ma})|}, \tag{7}$$

we have

$$\left|\hat{g}(\mathbf{x}, \boldsymbol{\theta}, \mathcal{S}_1^{ma}) - g^{(1)}(\mathbf{x})\right| \leqslant 2\hat{\epsilon}^{\max}(\mathcal{S}_1^{tr}, f^{(1)}, \boldsymbol{\theta}) \quad \text{and}$$

$$\left|\hat{g}(\mathbf{x}, \boldsymbol{\theta}, \mathcal{S}_0^{ma}) - g^{(0)}(\mathbf{x})\right| \leqslant 2\hat{\epsilon}^{\max}(\mathcal{S}_0^{tr}, f^{(0)}, \boldsymbol{\theta}), \tag{8}$$

where $\mathcal{S}_0^{tr}$ *(resp.* $\mathcal{S}_1^{tr}$*) is the collection of control (resp. treated) units in the holdout set, and* g *is the true treatment outcome defined as in Proposition 2.*

Proof of this proposition is available in the appendix. During a FLAME run, part of FLAME's criterion is to minimize PE and therefore the corresponding $\hat{\epsilon}^{\max}$ values are controlled. This is because $\hat{\epsilon}^{\max}(\mathcal{S}^{ma}, f, \boldsymbol{\theta}) := \|\hat{\mathbf{y}} - \mathbf{y}\|_\infty$, where $\mathbf{y}$ is the vector formed by labels from $\mathcal{S}^{ma}$ and $\hat{\mathbf{y}}$ is the vector formed by predictions of instances in $\mathcal{S}^{ma}$ using f. Since $\widehat{\texttt{PE}}$ is proportional to the square of the L_2 norm of the same vector on the training set, minimizing $\widehat{\texttt{PE}}$ on the training set tends to control $\hat{\epsilon}^{\max}$ on the matching set. Also, part of FLAME's criterion maximizes the BF values, which implicitly maximizes $n(\mathbf{x}, \boldsymbol{\theta}, \mathcal{S}_0^{tr}), n(\mathbf{x}, \boldsymbol{\theta}, \mathcal{S}_1^{tr}), n(\mathbf{x}, \boldsymbol{\theta}, \mathcal{S}_0^{ma})$, and $n(\mathbf{x}, \boldsymbol{\theta}, \mathcal{S}_1^{ma})$. This in turn suggests the probability in (7) is large.

Proposition 7 can be linked to the almost-exact-matching problem in the following way: when $r \ll p$, and $\lambda_k = \Theta(\frac{1}{2^k})$, we have $(1 - \lambda_{\|\boldsymbol{\theta}\|_0 + r})^n$ being small for small $\|\boldsymbol{\theta}\|_0$ and large n. The implication of this is that, in a FLAME run with $r \ll p$, when many irrelevant covariates are eliminated, $\|\boldsymbol{\theta}\|_0$ becomes small and $|\mathcal{S}_0^{tr}|, |\mathcal{S}_1^{tr}|, |\mathcal{MG}(\boldsymbol{\theta}, \mathcal{S}_0^{tr})|, |\mathcal{MG}(\boldsymbol{\theta}, \mathcal{S}_1^{tr})|$ become large. In this case the probability in (7) is large. This serves as a type of justification for the FLAME procedure, since FLAME aims to be able to handle problems where $r \ll p$. As a result of $r \ll p$, we need only concentrate on matches (on $\boldsymbol{\theta}$) with $\|\boldsymbol{\theta}\|_0 \ll p$.

6. Experiments

In this section, we study the quality and scalability of FLAME on synthetic and real data. The real datasets we use are the US Census 1990 dataset from the UCI Machine Learning Repository (Lichman, 2013) and the US 2010 Natality data (National Center for

Health Statistics, NCHS, 2010). The bit-vector and SQL implementations are referred to as **FLAME-bit** and **FLAME-db** respectively. FLAME-bit was implemented using Python 2.7.13 and FLAME-db using Python, SQL, and Microsoft SQL Server 2016. We compared FLAME with several other (matching and non-matching) methods including: (1) **one-to-one Propensity Score Nearest Neighbor Matching (1-PSNNM)** (Ross et al., 2015), (2) 1-PSNNM with oracle variable selection, (3) **Genetic Matching** (GenMatch) (Diamond and Sekhon, 2013), (4) **Causal Forest** (Wager and Athey, 2018), (5) **Mahalanobis Matching**, (6) **double linear regression**, (7) **BART** (Chipman et al., 2010) and (8) **CTMLE** (Van Der Laan and Rubin, 2006). For 1-PSNNM with oracle variable selection, we reveal to the propensity score matcher the *true* important covariates for outcomes and/or the important covariates for propensity score, which covers the ideal case for propensity scare matching with variable reweighting (Schneeweiss et al., 2009). In double linear regression and BART, we fit two regressions, one for the treatment group and one for the control group; the estimate of treatment effect is given by the difference of the predictions of the two regressors. (Unlike the matching methods, the double regression method assumes a model for the outcome making it sensitive to misspecification. Here we correctly specify the linear terms of the generative model in the synthetic data experiments.) We also ran tests with **Coarsened Exact Matching** (CEM) (Iacus et al., 2011a) and **Cardinality Match** (Zubizarreta, 2012; Zubizarreta et al., 2014). CEM is not well suited for categorical data and the R package does not automatically handle more than 15 covariates. Cardinality match does not scale to the setting of our problem, since it tries to solve a computationally demanding mixed integer programming problem. In all experiments with synthetic data, the training set is generated with exactly the same setting as the dataset used for matching, unless otherwise specified. The computation time results are in Table 2a. The experiments were conducted on a Windows 10 machine with Intel(R) Core(TM) i7-6700 CPU processor (4 cores, 3.40GHz, 8M) and 32GB RAM.

6.1 Experiments with Synthetic Data

Since the true treatment effect cannot be obtained from real data, we created simulated data with specific characteristics and known treatment effects. Note that both FLAME-db and FLAME-bit always return the same treatment effects. Four of the simulated experiments use data generated from special cases of the following (treatment $T \in \{0, 1\}$):

$$y = \sum_{i=1}^{10} \alpha_i x_i + T \sum_{i=1}^{10} \beta_i x_i + T \cdot U \sum_{1 \leqslant i < \gamma \leqslant 5} x_i x_\gamma + \epsilon, \tag{9}$$

Here, $\alpha_i \sim N(10s, 1)$ with $s \sim \text{Uniform}\{-1, 1\}$, $\beta_i \sim N(1.5, 0.15)$, U is a constant and $\epsilon \sim \mathcal{N}(0, 0.1)$. This contains linear baseline effects and treatment effects, and a quadratic treatment effect term.

Below we present the main comparison of our approach to state-of-the-art causal methods. Section 6.1.1 provides evidence of significant improvements in terms of estimation of causal effects while Section 6.1.2 demonstrates the scalability of the method. Additional results on the effects of model misspecification on regression methods, scalability comparison of FLAME-bit and FLAME-db, the decay of performance for FLAME as less important

variables are eliminated, and the effect of the tuning parameter C are available in Appendix C.

6.1.1 Most matching methods cannot handle irrelevant variables

Most matching methods do not make a distinction between important and unimportant variables, and may try to match on all the variables, including irrelevant ones. Therefore, many matching methods may perform poorly in the presence of many irrelevant variables.

We consider 20,000 units (10,000 control and 10,000 treated) generated with (9) where $U = 1$. We also introduce 20 irrelevant covariates for which $\alpha_i = \beta_i = 0$, but the covariates are in the database. We generate $x_i \sim$ Bernoulli(0.5) for $1 \leqslant i \leqslant 10$. For $10 < i \leqslant 30$, $x_i \sim$ Bernoulli(0.1) in the control group and $x_i \sim$ Bernoulli(0.9) in the treatment group.

During the execution of FLAME, irrelevant variables are successively dropped before the important variables. We should stop FLAME before eliminating important variables when the `PE` drops to an unacceptable value. In this experiment, however, we also allowed FLAME to continue to drop variables until 28 variables were dropped and there were no remaining possible matches. Figure 1 compares estimated and true treatment effects for FLAME (Early Stopping), FLAME (Run Until No More Matches) and other methods. In this experiment, Figure 1a is generated by stopping when the `PE` drops (starting from values within [-2, -1]) to below -20, resulting in more than 15,000 matches out of 20,000 units. FLAME achieves substantially better performance than other methods, shown in Figure 1.

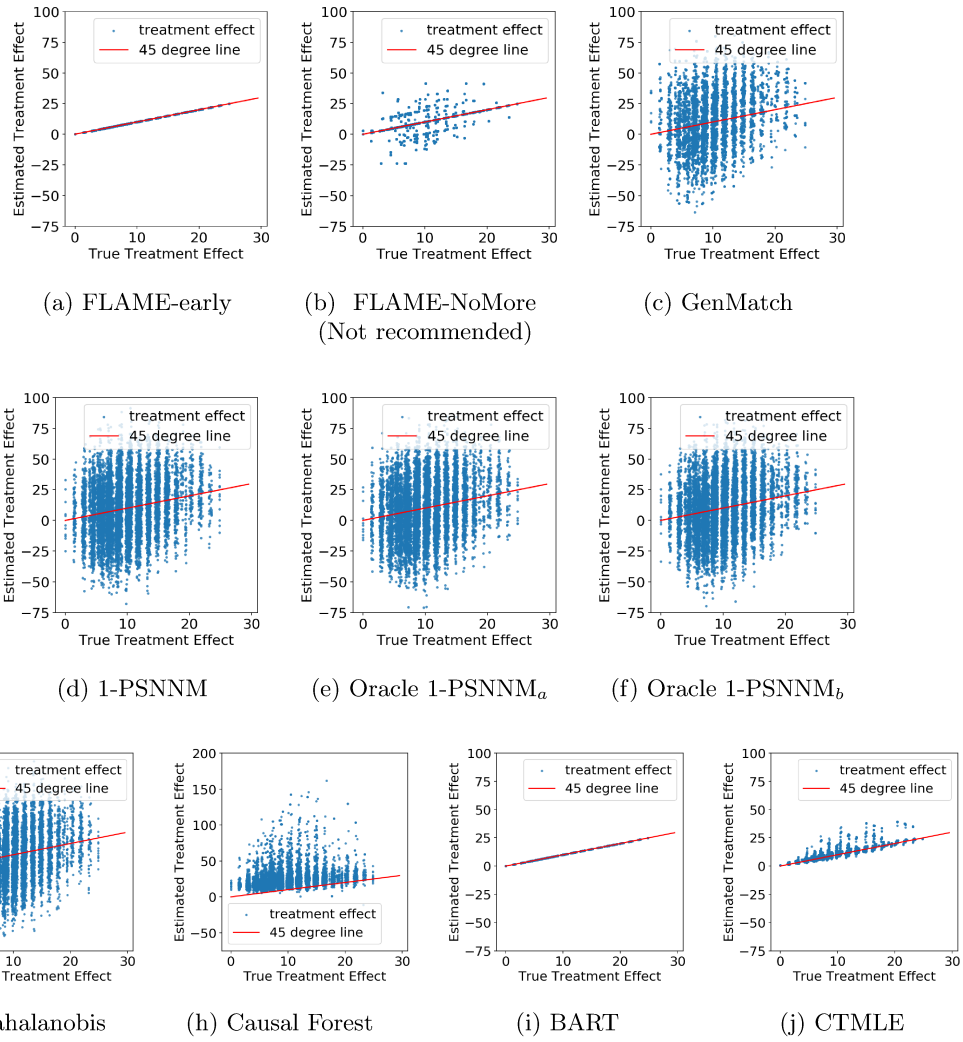
This experiment illustrates the main issues with classes of methods for causal inference: propensity score matching (of any kind) projects the data to one dimension, and thus cannot be used for CATE estimation. GenMatch has similar problems, and cannot be used for reliable CATE estimation. Regression and other modeling methods are subject to misspecification. Most methods do not produce interpretable matches, and those that do cannot scale to datasets of the sizes we consider, as we will soon see.

6.1.2 Scalability evaluation

We compare the execution time of FLAME-bit and FLAME-db with the other methods. FLAME-bit is faster than most of the other approaches (including FLAME-db) for the synthetic data considered in this experiment. However, for larger dataset, FLAME-db is much faster than FLAME-bit and all other methods. Table 2a compares the runtime when using the the US Census 1990 dataset (Lichman, 2013) with more than 1.2 million units and 59 covariates. Table 2b summarizes the runtime of all methods when using synthetic data.

6.1.3 Upper bound on CATE sample variance

While the main thrust of the methodology is in reducing the bias of estimating a CATE, the methodology naturally lends itself for estimating an upper bound on the variance of the CATE. Here we briefly discuss the conditional variance of the estimation in the setting of Figure 1. We note that asymptotic variance estimates for ATEs may also be obtained with the methods outlined in Abadie and Imbens (2012).



(a) FLAME-early (b) FLAME-NoMore (Not recommended) (c) GenMatch

(d) 1-PSNNM (e) Oracle 1-PSNNM$_a$ (f) Oracle 1-PSNNM$_b$

(g) Mahalanobis (h) Causal Forest (i) BART (j) CTMLE

Figure 1: Scatter plots of true treatment effect and estimated treatment effect on matched units. For FLAME-early (1a), we extract the results before the PE drops from [-2, -1] to below -20, and we call this early stopping. Before early stopping, we lose less than 2% in PE by removing covariates. In contrast, FLAME-NoMore (1b) is the result for running FLAME until there are no more matches. We do not recommend running FLAME past its stopping criteria but we did this for demonstration purposes. CTMLE uses BART as its base predictor. Oracle 1-PSNNM$_a$ runs 1-PSNNM with important covariates (the 10 covariates in Eq. 9), and Oracle 1-PSNNM$_b$ runs 1-PSNNM with unimportant covariates (all of the 20 covariates that do not contribute to outcomes but influence the propensity score). The latter setting covers the ideal scenario for the propensity score matching with variable reweighting/selection based on propensity scores (e.g., Schneeweiss et al., 2009), since all variables that affect the propensity score are included in the model.

Method	Time (hours)
FLAME-bit	Crashed
FLAME-db	1.33
Causal Forest	Crashed
1-PSNNM	> 10
Mahalanobis	> 10
GenMatch	> 10
Cardinality Match	> 10

(a) Timing results of different methods on the US Census 1990 dataset (Lichman, 2013).

Method	Time (seconds)
FLAME-bit	22.30 ± 0.37
FLAME-db	59.68 ± 0.24
Causal Forest	52.65 ± 0.41
1-PSNNM	13.88 ± 0.14
Mahalanobis	55.78 ± 0.14
GenMatch	> 150
Cardinality Match	> 150

(b) Timing results of FLAME compared with other methods on the synthetic data generated by the same procedure as in Figure 1, in the format of "average time" $\pm$ "standard deviation." It summarizes 3 runs.

Table 2: Timing results for FLAME.

While the covariance between the potential outcomes is not able to be calculated directly, we can upper bound it using the Cauchy-Schwarz inequality to give us:

$$\mathrm{Var}(y_t - y_c|\mathbf{x}) \leqslant \left(\mathrm{Std}(y_t|\mathbf{x}) + \mathrm{Std}(y_c|\mathbf{x})\right)^2 . \tag{10}$$

We note that if it is reasonable to assume that the potential outcomes are not negatively correlated, a tighter upper bound is possible as the cross-term can be ignored. We then use the sample variance to estimate $\mathrm{Std}(y_t|\mathbf{x})$ and $\mathrm{Std}(y_c|\mathbf{x})$ for each group. A simulated study of variance upper bound is shown in Figure 2, where the simulation data is generated using Eq. 9 (with 25,000 treated units, 25,000 control units, 20 irrelevant covariates, variance of Gaussian noise being 1, second order coefficient $U = 1$, and all other parameters as default). Each box plot in the figure summarizes the standard deviations of groups in a single iteration of the *while* loop. They are arranged according to the order in which the variables are dropped. This method estimates a small standard deviation for the first few levels as no relevant covariates have been eliminated. The standard deviation increases abruptly once a relevant covariate is dropped since this introduces variability in the treatment effects observed within each group. A sharp increase in the standard deviations thus likely corresponds to the dropping of important covariates and a decay in the match quality. This suggests another heuristic for the early stopping of FLAME, which is to stop when a sharp increase is observed.

6.2 Experiments with Real Data

Here we use FLAME to obtain estimates of treatment effects for an important societal question: the effect of prenatal smoking on the health of the newborn child.

There is a large literature studying the effects of maternal smoking during pregnancy on neonatal health outcomes. For example, recent work by Kondracki (2020) shows a significant effect of "extreme smoking" (defined as smoking at least 10 cigarettes daily throughout the pregnancy) on the odds of an infant requiring a Neonatal Intensive Care Unit (NICU) stay. In this section we employ FLAME to yield a more granular understanding of such

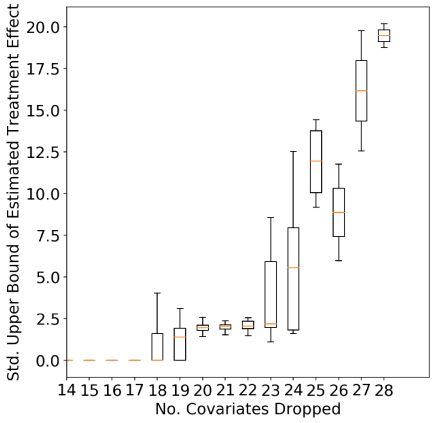


Figure 2: This figure plots the standard deviation upper bound estimates, i.e., empirical samples of square root of right-hand-side of Eq. 10. The box plots summarize the estimated standard deviation upper bounds of the groups created as covariates are eliminated. The orange lines in the boxes are medians. The lower and upper boundaries of the boxes are the first quartile and the third quartile of the distribution, while the lower and upper black bar correspond to 5th and 95th percentile, respectively.

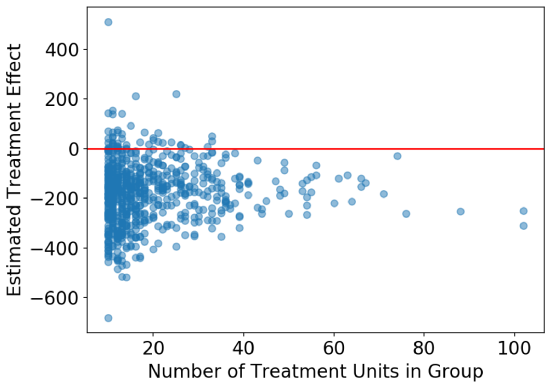


Figure 3: Scatter plots of estimated treatment effect (on newborn's weight in grams) against group size. Only matched groups with more than 30 control units and 10 treatment units are shown. The red line plots treatment effect being 0.

claims—endeavoring to understand whether heterogeneous effects might be present. To do this, we study the 2010 US Natality dataset from the National Vital Statistics System of the National Center for Health Statistics, NCHS (2010). As done by Kondracki (2020), we restrict ourselves to standard term (37+ weeks) singular (i.e., not twins, triplets, etc.) births. Treatment is defined as smoking at least 10 cigarettes per day for the duration of the pregnancy and our comparison group consists of those women who did not smoke at all during pregnancy (Kondracki, 2019). The demographic and medical variables we consider can be found in Table 4 in the appendix. The data we were provided with includes

discretized mother's and father's age variables, binned at five year increments, that we use for analysis. An alternative left for follow up work can leverage tools for continuous variables such as MALTS (Parikh et al., 2019) in conjunction with FLAME to analyze the raw data, including the raw age.

We study two topics: *NICU admissions* and *child's low birth weight*. Both demonstrate the strength of our approach in providing granular causal analyses, but in different ways. After eliminating units whose outcomes and/or treatment indicators are missing, there were $\sim$2.1M units, among which $\sim$75K units are treated units. We first estimate an overall odds ratio (by averaging odds ratio across matched groups) of NICU admission of approximately 2.6 which is in line with the literature (Adams et al., 2002). The power of our methodology is the ability to perform a deeper dive into the data. We identify multiple large groups with odds ratios that are substantially below 1, shown in Figure 10 in the appendix, which suggests that *the available data are not sufficient for granular analysis, or any strong conclusion on NICU admissions*, since it is unlikely that there is a protective effect of smoking on NICU admissions (Abrevaya, 2006; Honein et al., 2001). These results can potentially be explained by the diverse set of reasons that individuals might be admitted into the NICU that are not accounted for in these data (Mahendra et al., 2020). Such issues with these data have not (as far as we know) previously been identified despite the fact that they impact the trustworthiness of results on this important topic (Adams et al., 2002), where this same dataset is often used as the trusted source for studying this topic.

On the other hand, our analysis of the effect of "extreme smoking" on low birth weight leads to substantially clearer conclusions: The estimated average treatment effect of "extreme smoking" on birth weight is -248 grams of infant's weight, and the CATEs are presented for the high quality matched groups in Figure 3. We note that the estimates for the largest matched groups are most reliable and are also consistent with the literature (Kataoka et al., 2018; Kondracki, 2020). Importantly, this plot suggests little to no heterogeneity in the conditional treatment effects of smoking on birth weight; most of the extreme values are observed among small matched groups for which estimation is most difficult.

In summary, FLAME's ability to perform granular analysis is a determiner of trust: in some cases (e.g., smoking causing NICU admissions) we might question the results from other types of methods, e.g., those from methods that output an overall ATE. In other cases, FLAME might solidify our trust in the results (e.g., smoking causing lower birth weight). FLAME's results on NICU admission calls for (1) further medical study on the causal effect of prenatal smoking on NICU admission with a finer segmentation of the population, and (2) consideration of socio-economical causes (e.g., NICU care may not be affordable/available in certain circumstances) in order to obtain a cleaner medical causal relation between smoking and NICU admissions. Without the granular analysis that FLAME provides, it is possible that, in this analysis, as well as in many other analyses in other fields, incorrect/untrustworthy conclusions may not be noticed or investigated.

7. Conclusion

FLAME produces interpretable, high-quality matches. The defining aspects of FLAME are that it (i) *learns* an *interpretable* distance for matching by leveraging a training set, and (ii) uses an algorithm that eliminates successively less-relevant covariates, and permits SQL

and bit-vector operations to scale to millions of observations – these are datasets so large that they may not fit into main memory. We were able to theoretically evaluate the bias of FLAME under different distributions of the covariates. Through application to both simulated and real-world datasets, we were able to show that FLAME either ties with or outperforms existing matching methods in terms of estimation error, and provides a benefit in interpretability.

There are several natural extensions to the FLAME procedure: First, the less greedy approach of the Dynamic Almost Matching Exactly (DAME) algorithm (Dieng et al., 2019) extends the work in this paper by providing a solution to the FULL-AME problem, without backwards selection. However, DAME comes with a substantial computational burden as compared to FLAME and so can only be realistically applied to moderately sized problems. Dieng et al. (2019) demonstrate that in many practical settings FLAME and DAME achieve approximately the same accuracy for CATE estimation, and suggest that FLAME's backwards selection procedure be used until the number of covariates is small enough for a solution to DAME to be practical.

An obvious shortcoming of the FLAME framework is that it operates on categorical covariates. If we are given substantive information on what changes to the covariates would not influence outcomes, one can first coarsen continuous covariates (as in CEM), but such information is not always available. One solution for mixed continuous/discrete data is to use interpretable "stretch" matrices as distance metrics, and match to nearest neighbors using this stretched distance metric. This idea is the foundation for the MALTS algorithm (Parikh et al., 2018; Parikh et al., 2019). MALTS builds on FLAME by using a training set to learn the distance for matching. In doing this, Parikh et al. (2018) suggest to combine FLAME's learned Hamming distance for discrete data with MALTS' interpretable stretch metric for real-valued data. A further extension of FLAME, called Adaptive Hyperboxes (AHB), learns an interpretable box individually (and nonparametrically) around each data point (Morucci et al., 2020). Units within the box are matched. AHB relies on an underlying black box model to determine the distance metric, which is a key difference from FLAME and MALTS.

FLAME has been extended to handle network interference (Awan et al., 2020) and instrumental variable analysis (Awan et al., 2019).

The code for FLAME is available at `https://cran.r-project.org/web/packages/FLAME/index.html` (in R), and `https://www.github.com/almost-matching-exactly/` (in Python). An introduction to the project with links to the code is also found at `https://almost-matching-exactly.github.io/`. Gupta et al. (2021) provides a short overview of the FLAME-DAME software package.

Acknowledgments

We thank Jerry Chang for coding early versions of the FLAME ecosystem and to Neha Gupta and Vittorio Orlandi for the production versions of the Python and R packages. We would also like to thank the other members of the Almost Matching Exactly Lab at Duke for invaluable help: Harsh Parikh, Xian Sun, Thomas Howell, and Angikar Ghosal. We would like to thank Zijun Gao and Guillaume Basse for helpful discussions. This work was

partially supported by NIH QuBBD award R01EB025021, NSF award IIS-1703431, and by the Duke Energy Initiative.

References

Alberto Abadie and Guido W Imbens. A martingale representation for matching estimators. *Journal of the American Statistical Association*, 107(498):833–843, 2012.

Jason Abrevaya. Estimating the effect of smoking on birth outcomes using a matched panel data approach. *Journal of Applied Econometrics*, 21(4):489–519, 2006.

Kathleen E Adams, Vincent P Miller, Carla Ernst, Brenda K Nishimura, Cathy Melvin, and Robert Merritt. Neonatal health care costs related to smoking during pregnancy. *Health Economics*, 11(3):193–206, 2002.

Susan Athey, Julie Tibshirani, Stefan Wager, et al. Generalized random forests. *The Annals of Statistics*, 47(2):1148–1178, 2019.

M. Usaid Awan, Yameng Liu, Marco Morucci, Sudeepa Roy, Cynthia Rudin, and Alexander Volfovsky. Interpretable almost matching exactly with instrumental variables. In *Proceedings of the Thirty-Fifth Conference on Uncertainty in Artificial Intelligence, UAI*, 2019.

M. Usaid Awan, Marco Morucci, Vittorio Orlandi, Sudeepa Roy, Cynthia Rudin, and Alexander Volfovsky. Almost-matching-exactly for treatment effect estimation under network interference. *Proceedings of The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS*, 2020.

Alexandre Belloni, Victor Chernozhukov, and Christian Hansen. Inference on treatment effects after selection among high-dimensional controls. *The Review of Economic Studies*, 81(2):608–650, 2014.

M Alan Brookhart, Sebastian Schneeweiss, Kenneth J Rothman, Robert J Glynn, Jerry Avorn, and Til Stürmer. Variable selection for propensity score models. *American Journal of Epidemiology*, 163(12):1149–1156, 2006.

Matias D Cattaneo and Max H Farrell. Efficient estimation of the dose-response function under ignorability using subclassification on the covariates. *Advances in Econometrics*, 27:93, 2011.

F.S. Chapin. *Experimental Designs in Sociological Research*. Harper; New York, 1947.

Hugh A Chipman, Edward I George, Robert E McCulloch, et al. BART: Bayesian additive regression trees. *The Annals of Applied Statistics*, 4(1):266–298, 2010.

William G Cochran and Donald B Rubin. Controlling bias in observational studies: A review. *Sankhyā: The Indian Journal of Statistics, Series A*, pages 417–446, 1973.

Alexis Diamond and Jasjeet S. Sekhon. Genetic matching for estimating causal effects: A general multivariate matching method for achieving balance in observational studies. *The Review of Economics and Statistics*, 95(3):932–945, 2013.

Awa Dieng, Yameng Liu, Sudeepa Roy, Cynthia Rudin, and Alexander Volfovsky. Interpretable almost-exact matching for causal inference. In *Proceedings of Artificial Intelligence and Statistics (AISTATS)*, pages 2445–2453, 2019.

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*, 34(1):43–68, 2019.

MA Efroymson. Multiple regression analysis. *Mathematical Methods for Digital Computers*, pages 191–203, 1960.

Max H Farrell. Robust inference on average treatment effects with possibly more covariates than observations. *Journal of Econometrics*, 189(1):1–23, 2015.

Max H Farrell, Tengyuan Liang, and Sanjog Misra. Deep neural networks for estimation and inference: Application to causal effects and other semiparametric estimands. *arXiv preprint arXiv:1809.09953*, 2018.

Ernest Greenwood. *Experimental Sociology: A Study in Method*. King's Crown Press, 1945.

Neha R. Gupta, Vittorio Orlandi, Chia-Rui Chang, Tianyu Wang, Marco Morucci, Pritam Dey, Thomas J. Howell, Xian Sun, Angikar Ghosal, Sudeepa Roy, Cynthia Rudin, and Alexander Volfovsky. dame-flame: A python library providing fast interpretable matching for causal inference. arXiv 2101.01867, 2021.

Jinyong Hahn. Functional restriction and efficiency in causal inference. *Review of Economics and Statistics*, 86(1):73–76, 2004.

Günter J Hitsch and Sanjog Misra. Heterogeneous treatment effects and optimal targeting policy evaluation. *Available at SSRN 3111957*, 2018.

MA Honein, LJ Paulozzi, and ML Watkins. Maternal smoking and birth defects: validity of birth certificate data for effect estimation. *Public Health Reports*, 116(4):327, 2001.

Stefano M Iacus, Gary King, and Giuseppe Porro. Causal inference without balance checking: Coarsened exact matching. *Political Analysis*, 20(1):1–24, 2011a.

Stefano M Iacus, Gary King, and Giuseppe Porro. Multivariate matching methods that are monotonic imbalance bounding. *Journal of the American Statistical Association*, 106 (493):345–361, 2011b.

Guido W Imbens and Donald B Rubin. *Causal inference in statistics, social, and biomedical sciences*. Cambridge University Press, 2015.

Mariana Caricati Kataoka, Ana Paula Pinho Carvalheira, Anna Paula Ferrari, Maíra Barreto Malta, Maria Antonieta de Barros Leite Carvalhaes, and Cristina Maria Garcia de Lima Parada. Smoking during pregnancy and harm reduction in birth weight: a cross-sectional study. *BMC Pregnancy and Childbirth*, 18(1):1–10, 2018.

Anthony J Kondracki. Prevalence and patterns of cigarette smoking before and during early and late pregnancy according to maternal characteristics: the first national data based on the 2003 birth certificate revision, United States, 2016. *Reproductive Health*, 16(1): 142, 2019.

Anthony J Kondracki. Low birthweight in term singletons mediates the association between maternal smoking intensity exposure status and immediate neonatal intensive care unit admission: the e-value assessment. *BMC Pregnancy and Childbirth*, 20:1–9, 2020.

M. Lichman. UCI machine learning repository, 2013. URL `http://archive.ics.uci.edu/ml`.

Malini Mahendra, Martina Steurer-Muller, Samuel F Hohmann, Roberta L Keller, Anil Aswani, and R Adams Dudley. Predicting NICU admissions in near-term and term infants with low illness acuity. *Journal of Perinatology*, pages 1–8, 2020.

Robert L McCornack. A comparison of three predictor selection techniques in multiple regression. *Psychometrika*, 35(2):257–271, 1970.

Marco Morucci, Md. Noor-E-Alam, and Cynthia Rudin. Hypothesis tests that are robust to choice of matching method. *arXiv preprint arXiv:1812.02227*, 2018.

Marco Morucci, Vittorio Orlandi, Sudeepa Roy, Cynthia Rudin, and Alexander Volfovsky. Adaptive hyper-box matching for interpretable individualized treatment effect estimation. *Proceedings of the Thirty-Sixth Conference on Uncertainty in Artificial Intelligence, UAI*, 2020.

National Center for Health Statistics, NCHS. User guide to the 2010 natality public use file. Technical report, Centers for Disease Control and Prevention (CDC), 2010.

M. Noor-E-Alam and C. Rudin. Robust nonparametric testing for causal inference in natural experiments. *Working paper*, 2015.

Harsh Parikh, Cynthia Rudin, and Alexander Volfovsky. MALTS: Matching After Learning to Stretch. *arXiv e-prints: arXiv:1811.07415*, Nov 2018.

Harsh Parikh, Cynthia Rudin, and Alexander Volfovsky. An application of matching after learning to stretch (MALTS) to the ACIC 2018 causal inference challenge data. *Observational Studies*, pages 118–130, 2019.

PostgreSQL. *PostgreSQL*, 2016. URL `http://www.postgresql.org`.

Jeremy A Rassen and Sebastian Schneeweiss. Using high-dimensional propensity scores to automate confounding control in a distributed medical product safety surveillance system. *Pharmacoepidemiology and drug safety*, 21(S1):41–49, 2012.

María Resa and José R Zubizarreta. Evaluation of subset matching methods and forms of covariate balance. *Statistics in Medicine*, 35(27):4961–4979, 2016.

Paul R Rosenbaum. Imposing minimax and quantile constraints on optimal matching in observational studies. *Journal of Computational and Graphical Statistics*, 2016.

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, 1983.

Michelle E Ross, Amanda R Kreider, Yuan-Shung Huang, Meredith Matone, David M Rubin, and A Russell Localio. Propensity score methods for analyzing observational data like randomized experiments: challenges and solutions for rare outcomes and exposures. *American Journal of Epidemiology*, 181(12):989–995, 2015.

Donald B Rubin. Matching to remove bias in observational studies. *Biometrics*, pages 159–183, 1973a.

Donald B Rubin. The use of matched sampling and regression adjustment to remove bias in observational studies. *Biometrics*, pages 185–203, 1973b.

Donald B Rubin. Multivariate matching methods that are equal percent bias reducing, i: Some examples. *Biometrics*, pages 109–120, 1976.

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

Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1:206–215, May 2019.

Sebastian Schneeweiss, Jeremy A Rassen, Robert J Glynn, Jerry Avorn, Helen Mogun, and M Alan Brookhart. High-dimensional propensity score adjustment in studies of treatment effects using health care claims data. *Epidemiology*, 20(4):512, 2009.

Mark J Van Der Laan and Daniel Rubin. Targeted maximum likelihood learning. *The International Journal of Biostatistics*, 2(1), 2006.

Stefan Wager and Susan Athey. Estimation and inference of heterogeneous treatment effects using random forests. *Journal of the American Statistical Association*, 113(523):1228–1242, 2018.

Jay L Zagorsky. Marriage and divorce's impact on wealth. *Journal of Sociology*, 41(4): 406–424, 2005.

José R Zubizarreta. Using mixed integer programming for matching in an observational study of kidney failure after surgery. *Journal of the American Statistical Association*, 107 (500):1360–1371, 2012.

José R Zubizarreta, Ricardo D Paredes, Paul R Rosenbaum, et al. Matching for balance, pairing for heterogeneity in an observational study of the effectiveness of for-profit and not-for-profit high schools in chile. *The Annals of Applied Statistics*, 8(1):204–231, 2014.

José R. Zubizarreta and Luke Keele. Optimal multilevel matching in clustered observational studies: A case study of the effectiveness of private schools under a large-scale voucher system. *Journal of the American Statistical Association*, 112(518), 2017.

Appendix

A. Proof of Proposition 2

For a set of $i = 1, \ldots, n$ units, consider potential outcomes $g^{(t)}(\mathbf{x}_i)$, with covariates $\mathbf{x} \in \{0,1\}^p$ and treatment indicator $t \in \{0,1\}$. Observed treatments are denoted by the random variable T_i for all units i, and observed outcomes are denoted by $Y_i = T_i g^{(1)}(\mathbf{x}_i) + (1 - T_i) g^{(0)}(\mathbf{x}_i)$. Let $\mathbf{1}_{(\mathbf{x} \neq \mathbf{x}')}$ be a vector that is one at position j if $x_j \neq x'_j$ and 0 everywhere else. Let $\mathbf{w}^T \mathbf{1}_{(\mathbf{x} \neq \mathbf{x}')}$ be the (positive) weighted Hamming distance between $\mathbf{x}$ and $\mathbf{x}'$ with a vector of weights of length p, denoted by $\mathbf{w}$, where each entry of $\mathbf{w}$ is a positive real number, and such that $0 < \|\mathbf{w}\|_2 < \infty$. Define the matched group for a value $\mathbf{x}$ as $\mathcal{MG}(\mathbf{x}, \boldsymbol{\theta}, \mathcal{S}^{ma}) = \{\mathbf{x}_i,\ i = 1, \ldots, n\ s.t:\ \mathbf{x}_i \circ \theta = \mathbf{x} \circ \theta\}$, i.e., the set of units in the sample that have value $\mathbf{x}$ of the covariates selected by θ. Define also $n_t(\mathbf{x}, \boldsymbol{\theta}, \mathcal{S}^{ma}) = \sum_{\mathbf{x}_i \in \mathcal{MG}(\mathbf{x}, \boldsymbol{\theta}, \mathcal{S}^{ma})} T_i$ and $n_c(\mathbf{x}, \boldsymbol{\theta}, \mathcal{S}^{ma}) = \sum_{\mathbf{x}_i \in \mathcal{MG}(\mathbf{x}, \boldsymbol{\theta}, \mathcal{S}^{ma})} (1 - T_i)$. Let:

$$M := \max_{\substack{\mathbf{x}, \mathbf{x}' \in \{0,1\}^p \\ t \in \{0,1\}}} \frac{|g^{(t)}(\mathbf{x}') - g^{(t)}(\mathbf{x})|}{\mathbf{w}^T \mathbf{1}_{(\mathbf{x}' \neq \mathbf{x})}},$$

and assume that $M < \infty$. Then, for all $\mathbf{x}, \mathbf{x}' \in \{0,1\}^p$ and $t \in \{0,1\}$, by the definition of M above, we have: $|g^{(t)}(\mathbf{x}) - g^{(t)}(\mathbf{x}')| \leqslant M \mathbf{w}^T \mathbf{1}_{(\mathbf{x}' \neq \mathbf{x})}$ and

$$g^{(t)}(\mathbf{x}) - M \mathbf{w}^T \mathbf{1}_{(\mathbf{x} \neq \mathbf{x}')} \leqslant g^{(t)}(\mathbf{x}') \leqslant g^{(t)}(\mathbf{x}) + M \mathbf{w}^T \mathbf{1}_{(\mathbf{x} \neq \mathbf{x}')}. \tag{11}$$

Now we pick an arbitrary pair of $\mathbf{x}$ and $\boldsymbol{\theta}$ and consider the term $\mathbf{w}^T \mathbf{1}_{(\mathbf{x} \neq \mathbf{x}')}$ for any $\mathbf{x}' \in \mathcal{MG}(\mathbf{x}, \boldsymbol{\theta}, \mathcal{S}^{ma})$:

$$\begin{aligned}
\mathbf{w}^T \mathbf{1}_{(\mathbf{x} \neq \mathbf{x}')} &= \sum_{j=1}^{p} w_j \mathbf{1}_{(x_j \neq x'_j)} && (x_j (\text{resp. } x'_j) \text{ is the } j\text{-th entry of } \mathbf{x} \text{ (resp. } \mathbf{x}')) \\
&= \sum_{j=1}^{p} w_j (1 - \theta_j) \mathbf{1}_{(x_j \neq x'_j)} && (\text{Since } \mathbf{x}' \in \mathcal{MG}(\mathbf{x}, \boldsymbol{\theta}, \mathcal{S}^{ma})) \\
&\leqslant \sum_{j=1}^{p} w_j (1 - \theta_j) = \mathbf{w}^T (\mathbf{1} - \boldsymbol{\theta}), && (12)
\end{aligned}$$

where $\mathbf{1}$ is a vector of length p that is one in all entries. The second line follows because $\mathbf{x}' \in \mathcal{MG}(\mathbf{x}, \boldsymbol{\theta}, \mathcal{S}^{ma})$ implies that $\mathbf{x}'$ must match exactly with $\mathbf{x}$ on the covariates selected by $\boldsymbol{\theta}$. We want to use the estimator $\hat{\tau}(\mathbf{x}, \boldsymbol{\theta}) = \frac{1}{n_t(\mathbf{x}, \boldsymbol{\theta}, \mathcal{S}^{ma})} \sum_{\mathbf{x}_i \in \mathcal{MG}(\mathbf{x}, \boldsymbol{\theta}, \mathcal{S}^{ma})} Y_i T_i - \frac{1}{n_c(\mathbf{x}, \boldsymbol{\theta}, \mathcal{S}^{ma})} \sum_{\mathbf{x}_i \in \mathcal{MG}(\mathbf{x}, \boldsymbol{\theta}, \mathcal{S}^{ma})} Y_i (1 - T_i)$ to estimate $\tau(\mathbf{x}) = g^{(1)}(\mathbf{x}) - g^{(0)}(\mathbf{x})$ for some fixed $\mathbf{x}$ and $\boldsymbol{\theta}$. We can construct an upper bound on the estimation error of this estimator as

follows:

$$
\begin{aligned}
&\frac{1}{n_t(\mathbf{x},\boldsymbol{\theta},\mathcal{S}^{ma})}\sum_{\mathbf{x}_i\in\mathcal{MG}(\mathbf{x},\boldsymbol{\theta},\mathcal{S}^{ma})} Y_iT_i - \frac{1}{n_c(\mathbf{x},\boldsymbol{\theta},\mathcal{S}^{ma})}\sum_{\mathbf{x}_i\in\mathcal{MG}(\mathbf{x},\boldsymbol{\theta},\mathcal{S}^{ma})} Y_i(1-T_i) - \tau(\mathbf{x}) \\
=&\frac{1}{n_t(\mathbf{x},\boldsymbol{\theta},\mathcal{S}^{ma})}\sum_{\mathbf{x}_i\in\mathcal{MG}(\mathbf{x},\boldsymbol{\theta},\mathcal{S}^{ma})} g^{(1)}(\mathbf{x}_i)T_i \\
&- \frac{1}{n_c(\mathbf{x},\boldsymbol{\theta},\mathcal{S}^{ma})}\sum_{\mathbf{x}_i\in\mathcal{MG}(\mathbf{x},\boldsymbol{\theta},\mathcal{S}^{ma})} g^{(0)}(\mathbf{x}_i)(1-T_i) - (g^{(1)}(\mathbf{x}) - g^{(0)}(\mathbf{x})),
\end{aligned} \tag{13}
$$

where we use $Y_i = T_i g^{(1)}(\mathbf{x}_i) + (1-T_i)g^{(0)}(\mathbf{x}_i)$, and $T_i^2 = T_i$ and $T_i(1-T_i) = 0$ for any i. Then from (11) we get:

$$
\begin{aligned}
&\frac{1}{n_t(\mathbf{x},\boldsymbol{\theta},\mathcal{S}^{ma})}\sum_{\mathbf{x}_i\in\mathcal{MG}(\mathbf{x},\boldsymbol{\theta},\mathcal{S}^{ma})} g^{(1)}(\mathbf{x}_i)T_i \\
&- \frac{1}{n_c(\mathbf{x},\boldsymbol{\theta},\mathcal{S}^{ma})}\sum_{\mathbf{x}_i\in\mathcal{MG}(\mathbf{x},\boldsymbol{\theta},\mathcal{S}^{ma})} g^{(0)}(\mathbf{x}_i)(1-T_i) - (g^{(1)}(\mathbf{x}) - g^{(0)}(\mathbf{x})) \\
\leqslant&\frac{1}{n_t(\mathbf{x},\boldsymbol{\theta},\mathcal{S}^{ma})}\sum_{\mathbf{x}_i\in\mathcal{MG}(\mathbf{x},\boldsymbol{\theta},\mathcal{S}^{ma})} [g^{(1)}(\mathbf{x}) + M\mathbf{w}^T\mathbf{1}_{(\mathbf{x}\neq\mathbf{x}_i)}]T_i \\
&- \frac{1}{n_c(\mathbf{x},\boldsymbol{\theta},\mathcal{S}^{ma})}\sum_{\mathbf{x}_i\in\mathcal{MG}(\mathbf{x},\boldsymbol{\theta},\mathcal{S}^{ma})} [g^{(0)}(\mathbf{x}) - M\mathbf{w}^T\mathbf{1}_{(\mathbf{x}\neq\mathbf{x}')}](1-T_i) \\
&- g^{(1)}(\mathbf{x}) + g^{(0)}(\mathbf{x}).
\end{aligned} \tag{14}
$$

Next, we combine (13) and (14) to get:

$$
\begin{aligned}
&\frac{1}{n_t(\mathbf{x},\boldsymbol{\theta},\mathcal{S}^{ma})}\sum_{\mathbf{x}_i\in\mathcal{MG}(\mathbf{x},\boldsymbol{\theta},\mathcal{S}^{ma})} Y_iT_i - \frac{1}{n_c(\mathbf{x},\boldsymbol{\theta},\mathcal{S}^{ma})}\sum_{\mathbf{x}_i\in\mathcal{MG}(\mathbf{x},\boldsymbol{\theta},\mathcal{S}^{ma})} Y_i(1-T_i) - \tau(\mathbf{x})\\
\leqslant&\frac{1}{n_t(\mathbf{x},\boldsymbol{\theta},\mathcal{S}^{ma})}\sum_{\mathbf{x}_i\in\mathcal{MG}(\mathbf{x},\boldsymbol{\theta},\mathcal{S}^{ma})} [g^{(1)}(\mathbf{x}) + M\mathbf{w}^T\mathbf{1}_{(\mathbf{x}\neq\mathbf{x}_i)}]T_i\\
&-\frac{1}{n_c(\mathbf{x},\boldsymbol{\theta},\mathcal{S}^{ma})}\sum_{\mathbf{x}_i\in\mathcal{MG}(\mathbf{x},\boldsymbol{\theta},\mathcal{S}^{ma})} [g^{(0)}(\mathbf{x}) - M\mathbf{w}^T\mathbf{1}_{(\mathbf{x}\neq\mathbf{x}')}](1-T_i)\\
&-g^{(1)}(\mathbf{x}) + g^{(0)}(\mathbf{x})\\
=&g^{(1)}(\mathbf{x})\frac{1}{n_t(\mathbf{x},\boldsymbol{\theta},\mathcal{S}^{ma})}\sum_{\mathbf{x}_i\in\mathcal{MG}(\mathbf{x},\boldsymbol{\theta},\mathcal{S}^{ma})} T_i\\
&-g^{(0)}(\mathbf{x})\frac{1}{n_c(\mathbf{x},\boldsymbol{\theta},\mathcal{S}^{ma})}\sum_{\mathbf{x}_i\in\mathcal{MG}(\mathbf{x},\boldsymbol{\theta},\mathcal{S}^{ma})} (1-T_i) - g^{(1)}(\mathbf{x}) + g^{(0)}(\mathbf{x})\\
&+\sum_{\mathbf{x}_i\in\mathcal{MG}(\mathbf{x},\boldsymbol{\theta},\mathcal{S}^{ma})} M\mathbf{w}^T\mathbf{1}_{(\mathbf{x}\neq\mathbf{x}_i)}\left(\frac{T_i}{n_t(\mathbf{x},\boldsymbol{\theta},\mathcal{S}^{ma})} + \frac{1-T_i}{n_c(\mathbf{x},\boldsymbol{\theta},\mathcal{S}^{ma})}\right)\\
=&\sum_{\mathbf{x}_i\in\mathcal{MG}(\mathbf{x},\boldsymbol{\theta},\mathcal{S}^{ma})} M\mathbf{w}^T\mathbf{1}_{(\mathbf{x}\neq\mathbf{x}_i)}\left(\frac{T_i}{n_t(\mathbf{x},\boldsymbol{\theta},\mathcal{S}^{ma})} + \frac{1-T_i}{n_c(\mathbf{x},\boldsymbol{\theta},\mathcal{S}^{ma})}\right)\\
\leqslant&\sum_{\mathbf{x}_i\in\mathcal{MG}(\mathbf{x},\boldsymbol{\theta},\mathcal{S}^{ma})} M\mathbf{w}^T(\mathbf{1}-\theta)\left(\frac{T_i}{n_t(\mathbf{x},\boldsymbol{\theta},\mathcal{S}^{ma})} + \frac{1-T_i}{n_c(\mathbf{x},\boldsymbol{\theta},\mathcal{S}^{ma})}\right)\\
=&2M\mathbf{w}^T(\mathbf{1}-\boldsymbol{\theta}),
\end{aligned}
$$

where the inequality in the second to last line follows from Equation (12). Using the same set of steps we can construct a lower bound on the estimation error:

$$
\begin{aligned}
&\frac{1}{n_t(\mathbf{x},\boldsymbol{\theta},\mathcal{S}^{ma})}\sum_{\mathbf{x}_i\in\mathcal{MG}(\mathbf{x},\boldsymbol{\theta},\mathcal{S}^{ma})} Y_iT_i - \frac{1}{n_c(\mathbf{x},\boldsymbol{\theta},\mathcal{S}^{ma})}\sum_{\mathbf{x}_i\in\mathcal{MG}(\mathbf{x},\boldsymbol{\theta},\mathcal{S}^{ma})} Y_i(1-T_i) - \tau(\mathbf{x})\\
\geqslant&\frac{1}{n_t(\mathbf{x},\boldsymbol{\theta},\mathcal{S}^{ma})}\sum_{\mathbf{x}_i\in\mathcal{MG}(\mathbf{x},\boldsymbol{\theta},\mathcal{S}^{ma})} [g^{(1)}(\mathbf{x}) - M\mathbf{w}^T\mathbf{1}_{(\mathbf{x}\neq\mathbf{x}_i)}]T_i\\
&-\frac{1}{n_c(\mathbf{x},\boldsymbol{\theta},\mathcal{S}^{ma})}\sum_{\mathbf{x}_i\in\mathcal{MG}(\mathbf{x},\boldsymbol{\theta},\mathcal{S}^{ma})} [g^{(0)}(\mathbf{x}) + M\mathbf{w}^T\mathbf{1}_{(\mathbf{x}\neq\mathbf{x}')}](1-T_i)\\
&-g^{(1)}(\mathbf{x}) + g^{(0)}(\mathbf{x})\\
\geqslant&-\sum_{\mathbf{x}_i\in\mathcal{MG}(\mathbf{x},\boldsymbol{\theta},\mathcal{S}^{ma})} M\mathbf{w}^T(\mathbf{1}-\theta)\left(\frac{T_i}{n_t(\mathbf{x},\boldsymbol{\theta},\mathcal{S}^{ma})} + \frac{1-T_i}{n_c(\mathbf{x},\boldsymbol{\theta},\mathcal{S}^{ma})}\right)\\
=&-2M\mathbf{w}^T(\mathbf{1}-\boldsymbol{\theta}).
\end{aligned}
$$

Putting together these two bounds we obtain the statement in the proposition.

B. Proof of Theorems in Section 5

Proof of Theorems 4 and 5: The proofs of these theorems are computer programs. They calculate the bias analytically (rather than numerically). They enumerate all possible covariate allocations. For each allocation, the program checks if the allocation is valid and then runs Oracle FLAME to compute the conditional average treatment effect in each bin. Subtracting these estimate from the true CATE values and averaging (uniformly) over bins yields the bias for that allocation. The theorems report the average over valid allocations. ■

Proof of Theorem 6: The results follows from the symmetry in the assignment of treatments and controls to valid allocations. Note that if an allocation to n units $T_n = (t_1, \ldots, t_n)$ is a valid allocation then so is $1 - T_n$. If the estimate of a CATE under the assignment T_n is $\sum \gamma_j \alpha_j + \sum \xi_j \beta_j$ then necessarily the estimate of the same CATE under $1 - T_n$ is $-\sum \gamma_j \alpha_j + \sum \eta_j \beta_j$ for some η_j. Thus the contribution of baseline covariate j to the estimation of the CATE is necessarily identical under T_n and $1 - T_n$ since it does not depend on the treatment value of any unit. Let us define $\widehat{CATE}_X(T_n)$ as the conditional average treatment effect for covariate combination X under assignment T_n. Note that the bias of estimating any CATE can be written as

$$\frac{1}{2^n}\left(\sum_{T_n:\text{unit 1 treated}} \widehat{CATE}_X(T_n) + \sum_{T_n:\text{unit 1 control}} \widehat{CATE}_X(T_n)\right) - CATE_X$$

where the summations are over all possible assignments. For every assignment T_n in the first summand, $1 - T_n$ must appear in the second summand, thus canceling the contribution of all α_j to the bias. ■

Proof of Proposition 7: We focus on the treated case, and the control counterpart follows naturally using the same argument. Since $p_1(X \circ \boldsymbol{\theta}) \geqslant \lambda_{\|\boldsymbol{\theta}\|_0}$, the probability of the error maximizing point of a fixed $f^{(1)} \in \mathcal{F}_{\|\boldsymbol{\theta}\|_0}$ not in $\mathcal{S}_1^{ma}$ is at most $(1-\lambda_{\|\boldsymbol{\theta}\|_0+r})^{|\mathcal{MG}(\boldsymbol{\theta},\mathcal{S}_1^{ma})|}$. This is true because the max error for a given $f^{(1)}$ only depends on the covariates selected by $\boldsymbol{\theta}$ and the r relevant covariates that contribute to the outcome (in the noiseless setting). Similarly, the probability that the error maximizing point not in $\mathcal{S}_1^{tr}$ is at most $(1-\lambda_{\|\boldsymbol{\theta}\|_0+r})^{|\mathcal{MG}(\boldsymbol{\theta},\mathcal{S}_1^{tr})|}$. Thus with probability at least $1 - (1 - \lambda_{\|\boldsymbol{\theta}\|_0+r})^{|\mathcal{S}_1^{tr}|} - (1 - \lambda_{\|\boldsymbol{\theta}\|_0+r})^{|\mathcal{MG}(\boldsymbol{\theta},\mathcal{S}_1^{ma})|}$, the error maximizing point for $f^{(1)}$ is in both $\mathcal{MG}(\boldsymbol{\theta}, \mathcal{S}_1^{tr})$ and $\mathcal{S}_1^{ma}$. When this happens, we have

$$\begin{aligned}
\left|\hat{g}(\mathbf{x}, \boldsymbol{\theta}, \mathcal{S}_1^{ma}) - g^{(1)}(\mathbf{x})\right| &\leqslant \left|\hat{g}(\mathbf{x}, \boldsymbol{\theta}, \mathcal{S}_1^{ma}) - f^{(1)}(\mathbf{x} \circ \boldsymbol{\theta})\right| + \left|f^{(1)}(\mathbf{x} \circ \boldsymbol{\theta}) - g^{(1)}(\mathbf{x})\right| && (15)\\
&\leqslant \hat{\epsilon}_1^{\max}(\mathcal{S}_1^{ma}, f^{(1)}, \boldsymbol{\theta}) + \hat{\epsilon}_1^{\max}(\mathcal{S}_1^{tr}, f^{(1)}, \boldsymbol{\theta}) && (16)\\
&= 2\hat{\epsilon}_1^{\max}(\mathcal{S}_1^{tr}, f^{(1)}, \boldsymbol{\theta}), && (17)
\end{aligned}$$

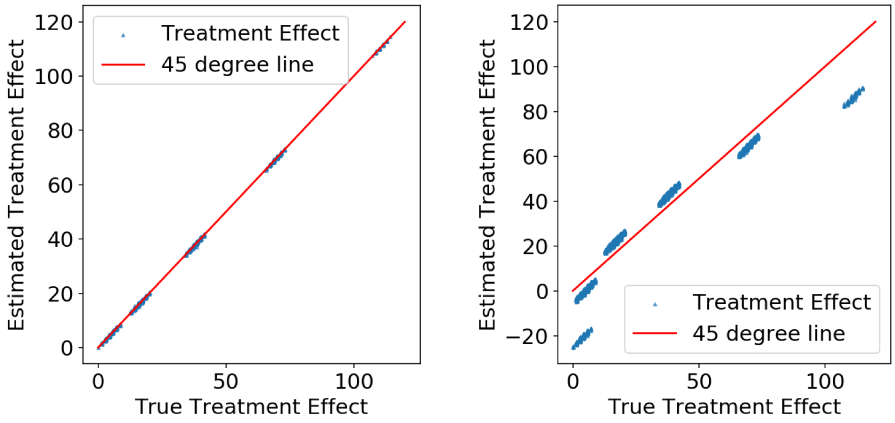
where (16) and (17) are due to that both $\hat{\epsilon}_1^{\max}(\mathcal{S}_1^{ma}, f^{(1)}, \boldsymbol{\theta})$ and $\hat{\epsilon}_1^{\max}(\mathcal{S}_1^{tr}, f^{(1)}, \boldsymbol{\theta})$ in this case are true max error for $f^{(1)}$.

■

C. Additional Experiment Details

C.1 Regression cannot handle model misspecification

We generated 20,000 observations from (9), with 10,000 treatment units and 10,000 control units, where U=10 and $x_i^T, x_i^C \sim$ Bernoulli(0.5). The nonlinear terms will cause problems for the (misspecified) linear regression models, but matching methods generally should not have trouble with nonlinearity. We ran FLAME, and all points were matched exactly on the first iteration yielding perfect CATEs. Scatter plots from FLAME and regression are in Figure 4. The axes of the plots are predicted versus true treatment effects, and it is clear that the nonlinear terms negatively impact the estimates of treatment effects from the regression models, whereas FLAME does not have problems estimating treatment effects under nonlinearity.



(a) FLAME

(b) Double linear regressors

Figure 4: Scatter plots of true treatment effect versus estimated treatment effect on every unit in the synthetic data. The regression model is misspecified, and performs poorly.

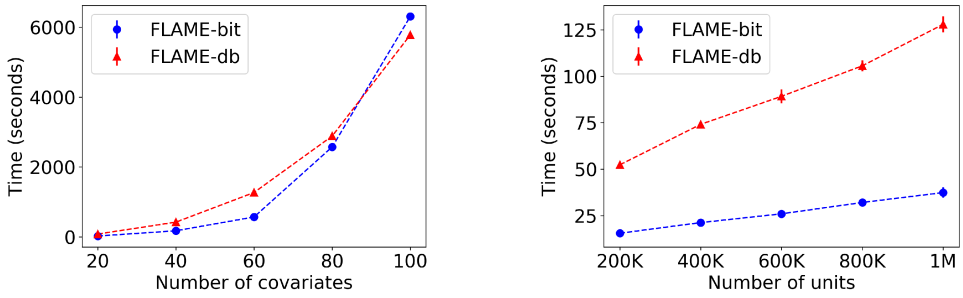
C.2 Scalability Comparison between FLAME-bit and FLAME-db

In this part, we evaluate scalability of both implementations of FLAME (FLAME-db and FLAME-bit) in a more complete manner. We generate synthetic datasets using the method described in (9) for different values of n and/or p. As in the previous settings, $x_i^C \sim$ Bernoulli(0.1) and $x_i^T \sim$ Bernoulli(0.9) for $i > 10$.

We compare the run time of FLAME-bit and FLAME-db as functions of p in Figure 5a (with n fixed to 100,000), and of n in Figure 5b (with p fixed to 15). In Figure 5, each dot represents the average runtime over a set of 4 experiments with the same settings; the vertical error bars represent the standard deviations in the corresponding set of experiments (we omit the very small error bars). The plots suggest that FLAME-db scales better with the number of covariates (though that happens more often when the number of covariates is large, beyond what we show here), whereas FLAME-bit scales better with the number of

units, since pre-processing and matrix operations used in FLAME-bit are expensive when the number of columns is large.

A major advantage of FLAME-db is that it can be used on datasets that are too large to fit in memory, whereas FLAME-bit cannot be used for such large datasets.



(a) $n = 100,000$ (b) $p = 15$

Figure 5: This figure shows how the runtime scales with the number of units and number of covariates for FLAME-db and FLAME-bit. In general, FLAME-bit is faster when the number of covariates is relatively small so that the multiplication of the bit vectors is not too expensive. On the other hand, FLAME-db is faster when the number of variables is relatively large. In both subfigures, the size of holdout sets is fixed to 100,000 (50,000 control and 50,000 treated).

C.3 Effect of the C parameter on the behavior FLAME

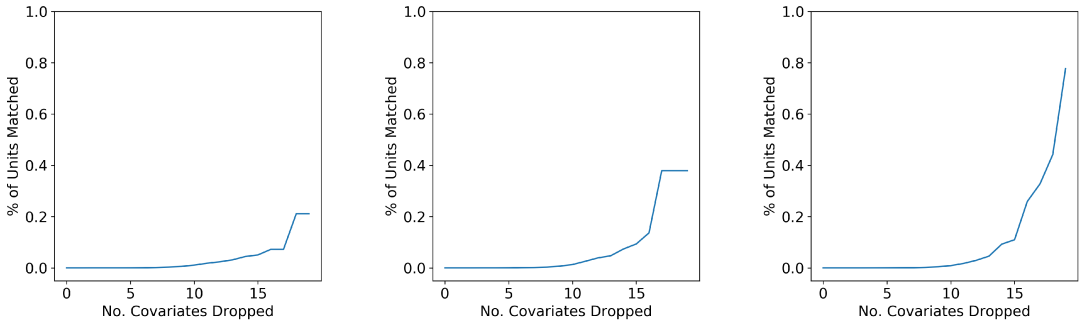
This experiment aims to understand how the parameter C would affect the estimation quality. Since the parameter C trades off the Prediction Error (`PE`) and Balancing Factor (`BF`), we create a setting where variables that are more important to outcome prediction are less balanced. More specifically, we created 15,000 control units and 15,000 treated units with the outcomes given by

$$y = \sum_{i=1}^{20} \frac{1}{i} x_i + 10T + \epsilon \tag{18}$$

where $x_i \sim \text{Bernoulli}\left(0.1 + \frac{3(i-1)}{190}\right)$ for the control group and $x_i \sim \text{Bernoulli}\left(0.9 - \frac{3(i-1)}{190}\right)$ for the treatment group, $T \in \{0, 1\}$ is the treatment indicator, and $\epsilon \sim \mathcal{N}(0, 0.1)$. Based on (6), we expect that the larger the parameter C, the earlier the algorithm eliminates covariates of higher balancing factor.

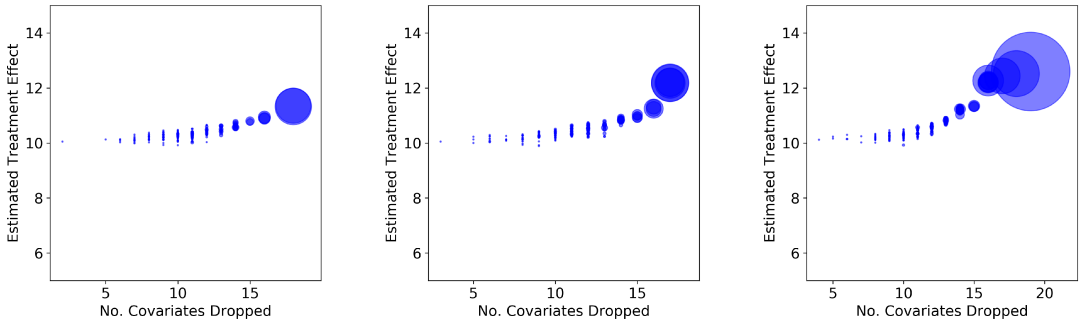
As we can see from Figure 6 and 7, larger C values encourage FLAME to sacrifice some prediction quality in return for more matches; and vice versa. Better prediction quality leads to less biased estimates while a larger balancing factor leads to more matched units. This is a form of bias-variance tradeoff. Figure 8 summarizes Figures 6 and 7 by plotting the vertical axis of Figure 6 against the vertical axis of Figure 7. Since the percent of units matched in Figure 6 is cumulative, the horizontal axis in Figure 8 also (nonlinearly) corresponds to the dropping of covariates. In Figure 7, each blue dot on the figures represents a matched

group. The bias-variance tradeoff between estimation quality versus more matched groups is apparent; the left figure shows fewer but high quality (low bias) matches, whereas the right figure shows more matches that are lower quality (higher bias).



(a) $C = 0.1$ (b) $C = 0.5$ (c) $C = 1$

Figure 6: This figure shows the percentage of units matched as FLAME eliminates variables with C being 0.1, 0.5 and 1. More units are matched when the value of C is large. The counts in these subfigures are cumulative. The vertical axis denotes the percentage of units matched.

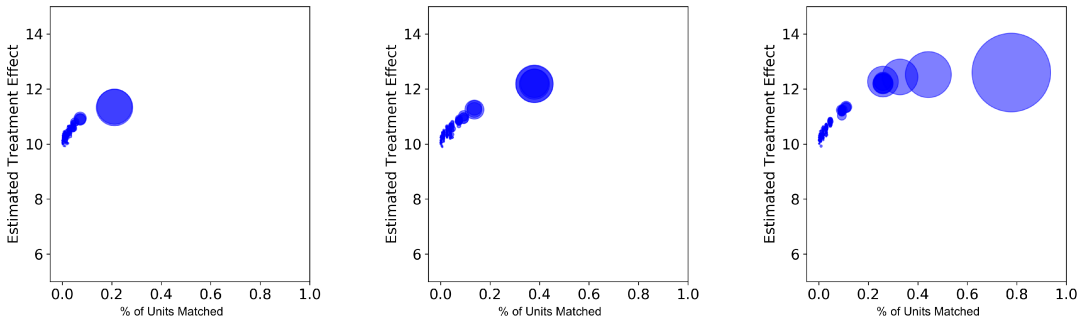


(a) $C = 0.1$ (b) $C = 0.5$ (c) $C = 1$

Figure 7: This figure shows how estimation quality changes as FLAME eliminates variables with C being 0.1, 0.5 and 1. The bias is smaller when the value of C is small. Here the size of the dots represents the number of units corresponding to that dot.

C.4 Decay of performance as less important variables are eliminated

To better understand the behavior of FLAME as it eliminates variables, we create a setting where the variables are all assigned non-zero importance. In this case, 20 covariates are used. As FLAME drops variables, prediction quality smoothly degrades. This is meant to represent problems where the importance of the variables decreases according to an exponential or a power-law, as is arguably true in realistic settings. Accordingly, we create



(a) $C = 0.1$ (b) $C = 0.5$ (c) $C = 1$

Figure 8: Estimated treatment effect versus % of units matched. This figure is a summary of Figure 6 and Figure 7. Here the horizontal axis is the vertical axis of Figure 6 and the vertical axis is the vertical axis of Figure 7. Size of the dots represents the number of units corresponding to that dot.

10,000 control units and 10,000 treated units with the outcomes generated as follows:

$$y = \sum_{i=1}^{20} \alpha_i x_i + 10T + \epsilon, \tag{19}$$

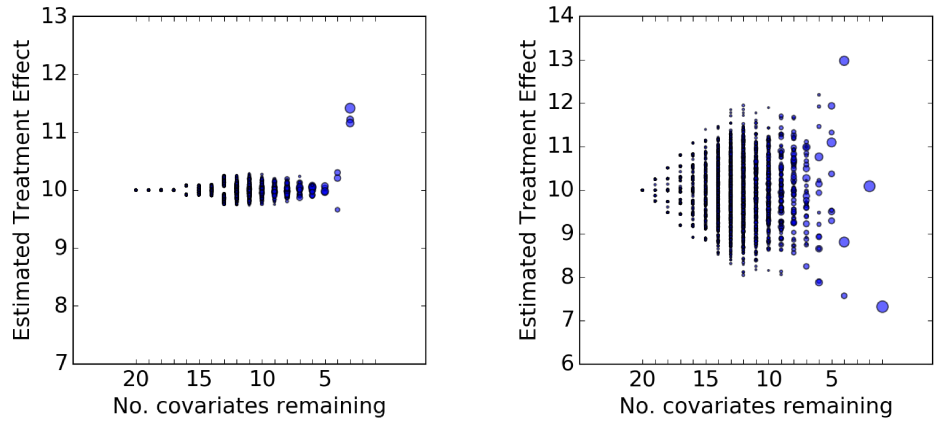
where $T \in \{0, 1\}$ is the binary treatment indicator, $x_i \sim$ Bernoulli(0.5), $\epsilon \sim \mathcal{N}(0, 0.1)$, $\alpha_i = 5 \times \left(\frac{1}{2}\right)^i$ for exponential decay (in Figure 9a), and $\alpha_i = 5 \times \frac{1}{i}$ for power-law decay (in Figure 9b). For both exponential and power law decay, variables with smaller indices are more important, and variables with higher indices are less important. In (19), all of the covariates contribute to the outcome positively. In the real world, variables can contribute to the outcome either positively or negatively, which leads to a smaller estimation bias. This is because when positive covariates and negative covariates are dropped in mixed order, some bias cancels out. The case we are considering, where all α_i's are positive, is essentially a worst case. It allows us to see more easily how prediction quality degrades.

The results from FLAME are shown in Figure 9, which shows that (i) the variability of estimation degrades smoothly as more variables are dropped, and (ii) the bias still remains relatively small. In this experiment, the value of C is set to be 0.001. The effect of varying the parameter C is studied in Appendix C.3. Since C is set to be small, FLAME dropped the covariates in ascending order of importance.

C.5 Natality Data Additional Details

C.5.1 Preprocessing for Natality Data

Different versions of birth certificate. As of 2010, two versions of birth certificates are used in the United States: the 1989 Revision of the U.S. Standard Certificate of Live Birth (unrevised version) and the 2003 revision of the U.S. Standard Certificate of Live Birth (revised version). The two versions record different variables about the parents and the infant. We therefore focus only on the data records of the revised version.



(a) Exponential decay

(b) Power law decay

Figure 9: Degradation of treatment effect estimates as FLAME drops variables. *Left:* variable importance decreases exponentially with base being $\frac{1}{2}$. *Right:* the variable importance decreases according to a power-law with exponent -1. The true treatment effect is 10 for all units in both subfigures. Dot size represents the number of units corresponding to that dot. There are fewer matches in high dimensions, and those matches are of higher quality. This shows that the bias of estimation degrades smoothly as we eliminate variables.

Multiple columns coding the same information. In the raw data, several columns encode the same information when a numeric variable has a large range. For a few columns encoding the same information, one of them contains the raw values of the variable, and the rest are typically discretizations to different coarsening scales.

C.5.2 Covariates Dropping Order

The order of covariate dropping for FLAME on the natality data is summarized in Tables 3 and 4.

C.5.3 Evaluating NICU admissions

This subsection includes a figure of the matched group size versus estimated odds ratio of NICU admission due to smoking. See Figure 10.

C.6 US Census Data Additional Details

For preprocessing, we first discard the units whose income is zero (most of these units have age smaller than 16). This gives us 1,220,882 units to work with. Using 59 of the variables ($p = 59$) and these units, we evaluate scalability of different methods (Table 2a). More specifically, the experiment is posited as studying the causal effect of *marital status* (T) on *wage or salary income* (Y) in the year 1989, with the assumptions stated in the introduction (e.g., no unmeasured confounding). The wage variable (dIncome1) is binarized in the following way: 0 for $(0, 15,000]$ (low income), and 1 for larger than

Table 3: Covariates dropping order of a FLAME run on the natality data when the outcome is NICU admission. Covariates dropped earlier are listed closer to the top. In the table, we use *the Manual* to refer to the Natality Data Manual (National Center for Health Statistics, NCHS, 2010).

covariate	additional remarks
Father's Age	Father's Age Recode 11 column in the manual.
Weekday	Weekday column in the manual.
Mother's Education	Mother's Education column in the manual.
Total Birth Order	Total Birth Order Recode column in the manual.
Live Birth Order	Live Birth Order Recode column in the manual.
Father's Race	Father's Bridged Race column in the manual.
Mother's Race	Mother's Bridged Race column in the manual.
Mother's Age	Mother's Age Recode 9 column in the manual.
Chronic Hypertension	Chronic Hypertension column in the manual.
Previous Cesarean Deliveries	Previous Cesarean Deliveries column in the manual.
Mother's Marital Status	Mother's Marital Status column in the manual.
Prepregnancy Diabetes	Prepregnancy Diabetes column in the manual.
Prepregnancy Hypertension	Prepregnancy Hypertension column in the manual.
Previous Preterm Birth	Previous Preterm Birth column in the manual.
Birth Place	Birth Place Recode column in the manual.
Sex of Infant	Sex of Infant column in the manual.

15,000 (high income). This binarization gives 564,755 low-income people and 656,127 high-income people. The marital state is binarized with 1 for being married and 0 for unmarried (including divorced, widowed and separated). This preprocessing gives 722,688 treated units (married) and 498,194 control units (unmarried). We randomly sampled 10% of these units (122,089 units) as the training set. As a real dataset with a large number of rows and columns, the US Census Data is mainly used for scalability evaluation. For the estimated treatment effect, in agreement with the literature (Zagorsky, 2005), FLAME suggests marriage contributes positively to income.

Running time comparison: Methods in Table 2a other than FLAME-db either did not finish within 10 hours or crashed, whereas FLAME-db took 1.33 hours. In this case, FLAME-bit encounters a overflow problem.

D. Causal Forest is Not a Matching Method

The causal forest algorithm (Wager and Athey, 2018; Athey et al., 2019) is a method that learns both a model for the counterfactuals and one for the propensity. While causal forests are used to estimate CATEs, the method should not be interpreted as a matching method. Matching methods match a small subset of units to estimate CATEs. In practice, matching methods are essentially subgroup analysis methods. One top desiderata of a matching method is that one unit should not be matched to too many other units. If a unit is

Table 4: Covariates dropping order of a FLAME run on the natality data when the outcome is infant's birth weight. Covariates dropped earlier are listed closer to the top. In the table, we use *the Manual* to refer to the Natality Data Manual (National Center for Health Statistics, NCHS, 2010).

covariate	**additional remarks**
Father's Age	Father's Age Recode 11 column in the manual.
Mother's Education	Mother's Education column in the manual.
Weekday	Weekday column in the manual.
Mother's Age	Mother's Age Recode 9 column in the manual.
Father's Race	Father's Bridged Race column in the manual.
Mother's Marital Status	Mother's Marital Status column in the manual.
Prepregnancy Diabetes	Prepregnancy Diabetes column in the manual.
Previous Cesarean Deliveries	Previous Cesarean Deliveries column in the manual.
Total Birth Order	Total Birth Order Recode column in the manual.
Prepregnancy Hypertension	Prepregnancy Hypertension column in the manual.
Chronic Hypertension	Chronic Hypertension column in the manual.
Live Birth Order	Live Birth Order Recode column in the manual.
Previous Preterm Birth	Previous Preterm Birth column in the manual.
Sex of Infant	Sex of Infant column in the manual.
Mother's Race	Mother's Bridged Race column in the manual.
Birth Place	Birth Place Recode column in the manual.

matched to too many others, then the match will cease to be interpretable, which does not easily permit case-based reasoning.

In order to interpret causal forests as a matching method, one would need to define a match to be the number of points sharing at least one leaf of one tree in the forest with a given unit. However, with causal forests, this number increases rapidly with the number of trees in the forest. For instance, using the data generation process from Section 6 (repeated below), if one runs causal forest for 1000 iterations, the average number of units matched to any given unit is 450. After 5000 iterations, the average number of units matched to a given unit is over 1000. Let us demonstrate this:

We use the same model as in Section 6 to generate synthetic data, for each unit i:

$$y_i = \sum_{j}^{p} \alpha_j x_{ij} + T \sum_{j=1}^{p} \beta_j x_{ij} + T_i \cdot U \sum_{j,\gamma,\gamma>j} x_{ij} x_{i\gamma},$$

where i indexes units and j indexes covariates, α_j is a baseline effect of each covariate, β_j is the treatment effect conditional on covariate j. The model also includes a nonlinear interaction effect between covariates: $x_j x_\gamma$, where each covariate interacts with all the covariates that succeeds it. This term is weighted by the coefficient U. We simulate data from a model with 10,000 treatment and 10,000 control units, and 30 covariates, with 10 important and 20 unimportant (i.e., such that $\alpha_j, \beta_j = 0$). We implement causal forests with 200 trees and 1000 training samples per tree.

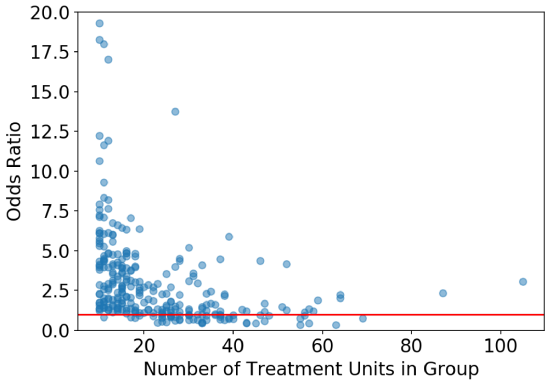


Figure 10: Scatter plot of odds ratios versus matched group size that is used to diagnose data quality issues for granular causal inference. The red line is $y = 1$. Only matched groups with more than 30 control units and 10 treatment units are shown. The overall odds ratio (weighted by group sizes) is approximately 2.67, which indicates a harmful outcome resulting from smoking. Our plot shows that in some cases the odds ratio is smaller than 1, even if the group size is relatively large. This variance calls for more investigation on the effect of smoking on NICU admission, especially from a medical perspective.

The result is in Figure 11, illustrating how the size of matched groups grows as a function of the number of iterations of the algorithm.

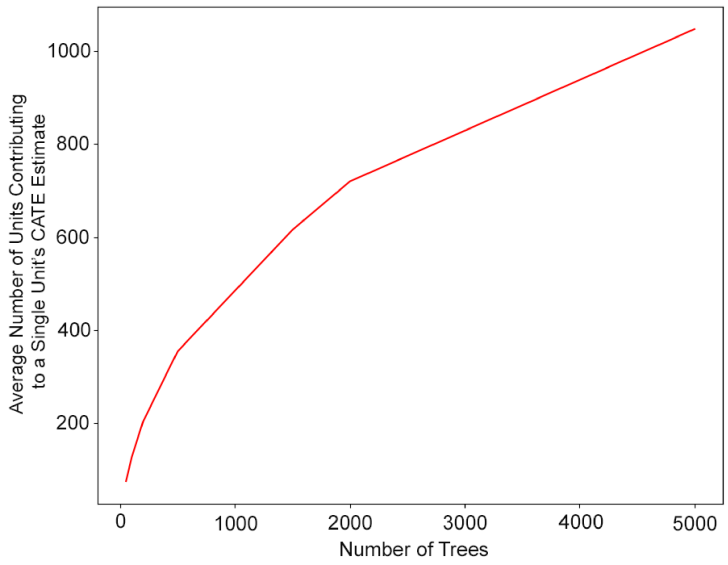


Figure 11: Average number of units used to predict a single unit's CATE by causal forests as the number of trees increases.

Thus, causal forests cannot be interpreted as a matching method for the following reasons:

- The size of the matched groups depends on the runtime of the method. Thus it is not clear where one would stop to define it as a matching method or as a non-matching method.
- If one were to try to obtain interpretable matched groups from causal forest, one would need to stop after only a few trees, which is not recommended by Wager and Athey (2018); Athey et al. (2019) because it would hurt performance.
- Causal forest was not designed to be a matching method. Examining its large matched groups would not necessarily achieve the main goals of case-based reasoning, such as troubleshooting variable quality or investigating the possibility of missing confounders.