A Free Lunch with Influence Functions? An Empirical Evaluation of Influence Functions for Average Treatment Effect Estimation

Matthew J. Vowels *matthew.vowels@unil.ch*
Institute of Psychology
University of Lausanne
Switzerland

Sina Akbari *sina.akbari@epfl.ch*
BAN
EPFL
Switzerland

Necati Cihan Camgoz *n.camgoz@surrey.ac.uk*
CVSSP
University of Surrey
U.K.

Richard Bowden *r.bowden@surrey.ac.uk*
CVSSP
University of Surrey
U.K.

Reviewed on OpenReview: *https://openreview.net/forum?id=dQxBRqCjLr*

Abstract

The applications of causal inference may be life-critical, including the evaluation of vaccinations, medicine, and social policy. However, when undertaking estimation for causal inference, practitioners rarely have access to what might be called 'ground-truth' in a supervised learning setting, meaning the chosen estimation methods cannot be evaluated and must be assumed to be reliable. It is therefore crucial that we have a good understanding of the performance consistency of typical methods available to practitioners. In this work we provide a comprehensive evaluation of recent semiparametric methods (including neural network approaches) for average treatment effect estimation. Such methods have been proposed as a means to derive unbiased causal effect estimates and statistically valid confidence intervals, even when using otherwise non-parametric, data-adaptive machine learning techniques. We also propose a new estimator 'MultiNet', and a variation on the semiparametric update step 'MultiStep', which we evaluate alongside existing approaches. The performance of both semiparametric and 'regular' methods are found to be dataset dependent, indicating an interaction between the methods used, the sample size, and nature of the data generating process. Our experiments highlight the need for practitioners to check the consistency of their findings, potentially by undertaking multiple analyses with different combinations of estimators.

1 Introduction

Causal inference and causal effect estimation are of the upmost importance for policy making (Kreif & DiazOrdaz, 2019), the development of medical treatments (Petersen et al., 2017), the evaluation of evidence

within legal frameworks (Pearl, 2009; Siegerink et al., 2016), and others. Unfortunately, causal inference is different to the typical supervised learning paradigm in machine learning insofar as ground-truth for the causal effect is not available. Practitioners are therefore forced to take a leap of faith, hoping that their estimation methods are reliable and yield meaningful estimates. It is therefore crucial that we have a good understanding of the consistency of the methods we use, before deploying them in practice.

In restricted circumstances we can use well-studied techniques which have known performance guarantees (*e.g.*, linear models for linear data generating processes and Gaussian errors). Indeed, most methods being utilized in empirical fields such as psychology or epidemiology are parametric models (van der Laan & Rose, 2011; Blanca et al., 2018), which are convenient because they facilitate closed-form statistical inference and confidence intervals (*e.g.* for the purpose of null hypothesis testing). However, there now exist numerous powerful, non-parametric, data-adaptive machine learning techniques which do not require strong assumptions about the functional form (*e.g.*, such as the assumption of linearity). These methods can be 'augmented' with semiparametric techniques, which can be used to reduce bias and derive confidence intervals for performing statistical inference. Practitioners may also find themselves choosing between the alluring 'deep learning' based methods and those more conventional machine learning approaches which perhaps, rightly or wrongly, have less associated hype.

Unfortunately, the empirical consistency of the performance of these machine learning approaches for average treatment effect estimation (both with and without the use of semiparametric techniques) is not clear. Indeed, recent work by Curth et al. (2021b) has already highlighted that conventional benchmark datasets for evaluating estimators can be misleading. Furthermore, a theoretical framework for establishing the statistical guarantees of neural-network based approaches is yet elusive (Curth et al., 2021a), although one notable recent contribution is presented by Farrell et al. (2021). In this work we undertake a much needed evaluation of machine learning and semiparametric approaches to average treatment effect estimation.

The associated contributions of this paper are:

- A new update step method 'MultiStep' (see Section 4) which attempts to improve upon existing update methods by continuously optimizing the solution according to two criteria which characterize the optimum solution (namely, finding the IF with the smallest expectation and variance)
- A new method 'MultiNet' (see Section 5) which attempts to mimic the performance of an ensemble with a single NN
- An extensive comparison (see Section 6) of the estimation performance of NNs and other algorithms with and without semiparametric techniques

We evaluate causal inference task performance in terms of (a) precision in estimation (and the degree to which we can achieve debiasing using semiparametric techniques), (b) double robustness (the extent to which we can improve the robustness of our estimators using semiparametric techniques), and (c) normality of the distribution of estimates (thus, by implication, whether it is possible to use closed-form expressions for confidence intervals and statistical inference). In order to evaluate the methods in terms of these three criteria, we 10 different datasets (varying in positive violations, non-linearity, and sample size). We evaluate across three relevant metrics, and implement all possible combinations of approaches to the estimation. Our findings suggest that evaluation on a handful of benchmark datasets is not sufficient for helping us understand the likelihood of causal estimation methods to perform well in real-world applications.

In general, the benefits of influence functions and the performance of a range of contemporary estimators is found to be inconsistent[1] Whilst our MultiNet and MultiStep methods provide competitive performance across datasets, and whilst we observe that initial estimation methods can sometimes benefit from the application of the semiparametric techniques, our evaluation highlights interactions between the underlying data generating process, sample sizes, and the estimators and update steps used. The conclusion is thus that

[1]It is hopefully clear from the context, but consistency can have two meanings. Firstly in statistical/technical terms, for whether an estimator converges towards some value as the sample size increases. Secondly, in the practical/empirical sense, we use it to describe whether or not a particular method, algorithm, or estimation approach performs well across a range of scenarios, over a range of metrics.

practitioners should take care when interpreting their results, and attempt to validate them by undertaking multiple analyses with different estimators. This is particularly important for the task of causal inference where, in real-world applications, ground truth is unlikely to be available.

The paper is structured as follows: We begin by reviewing previous work in Sec. 2 and provide background theory on the motivating case of estimating causal effects from observational data in Sec. 3. In this section, we also provide a top level introduction to IFs (Sec. 3.2). We present our own update approach MultiStep in Sec. 4. Our NN method MultiNet is presented in Sec. 5. The evaluation is presented in Sec. 6 and at the beginning of this section, we summarise the open questions which inform our subsequent evaluation design. We present and discuss results in Sec. 6.3 and Sec. 6.4 and we provide a summary of the experiments, conclusions, and opportunities for further work in Sec. 7. Finally, a broader impact statement is included in Sec. 8

Code for reproducing experiments is provided here: `https://github.com/matthewvowels1/FreeLunchSemiParametrics`

2 Context and Previous Work

Being able to perform statistical tests and reliably quantify uncertainty is especially important when evaluating the efficacy of treatments or interventions. One approach to perform such tests is by assuming a parametric model for the underlying generating mechanism, and *e.g.* normally distributed estimates. However, it has been argued that linear models are incapable of modeling most realistic data generating processes and that we should instead be using modern machine learning techniques (van der Laan & Rose, 2011; van der Laan & Gruber, 2012; van der Laan & Starmans, 2014; Vowels, 2021). Unfortunately, most machine learning models are non-parametric insofar as the estimates derived using such techniques are not directly parameterizable as (*e.g.*) a Gaussian with a mean and variance. As such, the estimates derived using such non-parametric techniques are not readily amenable to null-hypothesis significance testing or other common statistical inference tasks. Furthermore, even though machine learning algorithms are more flexible, they are still likely to be biased because they are not targeted to the specific parameter of interest (van der Laan & Rose, 2011). So, what can we do?

By leveraging concepts from the field of semiparametric statistics we can try to address these issues. Indeed, by combining elements of semiparametric theory with machine learning methods, we can, at least theoretically, enjoy the best of both worlds: We can avoid having to make unreasonably restrictive assumptions about the underlying generative process, and can nonetheless undertake valid statistical inference. Furthermore, we can also leverage an estimator update process to achieve greater precision in existing estimators, without needing additional data (van der Laan & Rose, 2011; Tsiatis, 2006; Bickel et al., 1998), an advantage which we might call a 'free lunch'[2], and one we wish to explore and test empirically in this paper.

In terms of the available estimation techniques, one approach which combines machine learning and semiparametric theory is known as targeted learning (van der Laan & Rose, 2011; van der Laan & Starmans, 2014).[3] This technique, and many related techniques involving influence functions (IFs) and semiparametric theory, have primarily been popularized outside the field of machine learning. In parallel, machine learning has focused on the development of equivalent methods using deep neural network (NN) methods for causal inference (see *e.g.*, Bica et al., 2020; Wu & Fukumizu, 2020; Shalit et al., 2017; Yoon et al., 2018; Louizos et al., 2017), which, owing to their 'untargeted' design (more on this below), may exhibit residual bias. As such, many of the principles and theory associated with semiparametrics and IFs are underused and underappreciated within the machine learning community, and it remains unknown to what extent these techniques can be applied to NN based estimators.

The possible applications of semiparametrics in machine learning are broad but under-explored, and IFs in particular have only seen sporadic application in explainable machine learning (Koh & Liang, 2017; Sani

[2]The term 'free lunch' is a reference to the adage of unknown origin (but probably North American) 'there ain't no such thing as a free lunch'. It was famously used by Wolpert and Macready in the context of optimization (Wolpert & Macready, 1997).

[3]For an overview of some other related methods see (Curth et al., 2021a).

et al., 2020), natural language processing (Han et al., 2020) models, causal model selection (Alaa & van der Schaar, 2019) and uncertainty quantification for deep learning (Alaa & van der Schaar, 2020). Outside of machine learning, in particular in the fields of epidemiology and econometrics, semiparametric methods are becoming more popular, and include targeted learning (van der Laan & Rose, 2011) and the well-known double machine learning approach by Chernozhukov et al. (2018). In statistics, alternatives have been developed which include doubly robust conditional ATE estimation (Kennedy, 2020) and IF-learning (Curth et al., 2021a).

Within the field representing the confluence of causal inference and machine learning, the focus seems to have been on the development of NN methods. See, for example, CEVAE (Louizos et al., 2017), CFR-Net (Shalit et al., 2017), GANITE (Yoon et al., 2018), Intact-VAE (Wu & Fukumizu, 2022) etc. However, these methods have been developed without a consideration for statistical inference or semiparametric theory, and this gap has been noted by Curth et al. (2021b) and Curth & van der Schaar (2021). Indeed, to the best of our knowledge, the application of semiparametric theory to debias neural network-based estimators has only be used three times in the field representing the confluence of machine learning and causal inference. Firstly, in DragonNet (Shi et al., 2019), a method designed for ATE estimation; secondly in TVAE (Vowels et al., 2021), a variational, latent variable method for conditional ATE and ATE estimation; and thirdly, by Farrell et al. (2021) where a restricted class of multilayer perceptrons were evaluated for their performance potential as plug-in estimators for semiparameteric estimation of causal effects and shown to yield promising performance. The first two methods incorporate targeted regularization, but do not readily yield statistical inference because to do so requires asymptotic normality (and this is not evaluated in the studies) as well as explicit evaluation of the IF. More broadly, semiparametrics has been discussed in relation to theory in machine learning, for example Bhattacharya et al. (2020) provides a discussion of influence functions in relation to Directed Acyclic Graphs with hidden variables, Rotnitzky & Smucler (2020) and Henckel et al. (2020) discuss the application of semiparametric techniques for identifying efficient adjustment sets for causal inference tasks, Zhong & Wang (2021) apply semi-parametrics with deep neural networks to achieve statistical inference in the partially linear quantile regression setting, and Jung et al. (2020) generalize the coverage of work on semiparametric estimation to general causal estimands. However, in general the work is quite sparse, particularly in relation to the applicability of the theory to neural networks, and the accessibility of the relevant theory to general practitioners of machine learning.

Prior comparisons of the performance of semiparametric approaches and causal inference methods exist. For example, the robustness of targeted learning approaches to causal inference on nutrition trial data was presented by Li et al. (2022) and includes a useful summary table of previous findings and includes its own evaluations. However, it does not include comparisons with NN-based learners, and seeks the answers to different questions relevant to practitioners in the empirical fields. Another example evaluation was undertaken by Luque-Fernandez et al. (2018) but has a didactic focus. Finally, and more broadly, Curth et al. (2021b) evaluate the use of benchmark datasets for the evaluation of treatment effect estimators. They find that certain datasets '...systematically favor some algorithms over others', highlighting a need for fairer evaluations of techniques designed for causal inference. This kind of benchmark dependency, referred to as the 'benchmark lottery', has been noted before in other supervised-learning domains such as computer vision and natural language processing (Dehghani et al., arXiv preprint). It is worth noting, however, that in contrast to supervised learning tasks, ground-truth is rarely available for causal inference, making it all the more crucial that we understand the performance of our estimators.

3 Background

Regarding notation, we use upper-case letters *e.g.* A, B to denote random variables, and bold font, upper-case letters to denote sets of random variables *e.g.* $\mathbf{A}, \mathbf{B}$. Lower-case a and b indicate specific realisations of random variables A and B. Specifically, we use $\mathbf{x}_i \sim P(\mathbf{X}) \in \mathbb{R}^m$ to represent the m-dimensional, pre-treatment covariates (we use bold symbols to signify multi-dimensional variables) for individual i assigned factual treatment $t_i \sim P(T|\mathbf{X}) \in \{0, 1\}$ resulting in outcome $y_i \sim P(Y|\mathbf{X}, T)$. Together, these constitute dataset $\mathcal{D} = \{[y_i, t_i, \mathbf{x}_i]\}_{i=1}^n$ where n is the sample size, sampled from a 'true' population distribution $\mathcal{P}$.

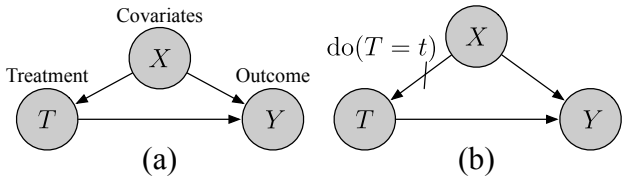

Figure 1: Directed Acyclic Graphs (DAGs) for estimating the effect of treatment $T = t$ on outcome Y with confounding X.

3.1 Causal Inference

Fig. 1a is characteristic of observational data, where the outcome is related to the covariates as well as the treatment, and treatment is also related to the covariates. For example, if we consider age to be a typical covariate, young people may opt for surgery, whereas older people may opt for medication. Assuming that an age-related risk mechanism exists, then age will confound our estimation of the causal effect of treatment on outcome.

One of the most common causal estimands is the Average Treatment Effect (ATE):

$$\tau(\mathbf{x}) = \mathbb{E}_{\mathbf{x}\sim P(\mathbf{X})}[\mathbb{E}_{y\sim P(Y|do(T=1)\mathbf{X}=\mathbf{x})}[y] - \mathbb{E}_{y\sim P(Y|do(T=0)\mathbf{X}=\mathbf{x})}[y]] \tag{1}$$

Here, the use of the *do* operator (Pearl, 2009) in $do(T = 1)$ and $do(T = 0)$ simulates interventions, setting treatment to a particular value regardless of what was observed. In Fig. 1b, such an intervention removes the dependence of T on $\mathbf{X}$, and this graph is the same as the one for an RCT, where the treatment is unrelated to the covariates. Using *do*-calculus we can establish whether, under a number of strong assumptions[4], the desired causal estimand can be expressed in terms of a function of the observed distribution, and thus whether the effect is *identifiable*. For the graph in Fig. 1a, the outcome under intervention can be expressed as $\mathbb{E}_{y\sim P(Y|do(T=t'))}[y]$ which is estimable from observational data. Here, t' is the specific intervention of interest (*e.g.*, $t' = 1$). In particular, it tells us that adjusting for the covariates $\mathbf{X}$ is sufficient to remove the bias induced through the 'backdoor' path $\mathbf{X} \to T \to Y$. This particular approach is sometimes referred to as backdoor adjustment.

One may use a regression to approximate the outcome under intervention, and indeed, plug-in estimators $\hat{Q}$ can be used for estimating the difference between the outcome under two different interventions on treatment (the Average Treatment Effect - ATE) as:

$$\hat{\boldsymbol{\tau}}(\hat{Q};\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n}(\hat{Q}(T=1,\mathbf{X}=\mathbf{x}_i) - \hat{Q}(T=0,\mathbf{X}=\mathbf{x}_i)), \tag{2}$$

We use the circumflex/hat ($\hat{\cdot}$) notation to designate an estimated (rather than true/population) quantity. In the simplest case, we may use a linear or logistic regression for the estimator $\hat{Q}$, depending on whether the outcome is continuous or binary. Unfortunately, if one imagines the true joint distribution to fall somewhere within an infinite set of possible distributions, we deliberately handicap ourselves by using a family of linear models because such a family is unlikely to contain the truth. The consequences of such model misspecification can be severe, and results in biased estimates (Vowels, 2021; van der Laan & Rose, 2011). In other words, no matter how much data we collect, our estimate will converge to the incorrect value, and this results in a false positive rate which converges to 100%. This clearly affects the interpretability and reliability of null-hypothesis tests. Furthermore, even with correct specification of our plug-in estimators (*i.e.*, we have achieved identification of the causal effect and our assumptions about the structure and functional relationships hold), our models are unlikely to be 'targeted' to the desired estimand, because they often

[4]These assumptions are the Stable Unit Treatment Value Assumption (SUTVA), Positivity, and Ignorability/Unconfoundedness - see Section A below for more information.

estimate quantities superfluous to the estimand but necessary for the plug-in estimator (*e.g.*, other relevant factors or statistics of the joint distribution). As a result, in many cases there exist opportunities to reduce residual bias using what are known as *influence functions*.

3.2 Influence Functions

Semiparametric theory and, in particular, the concept of Influence Functions (IFs), are known to be challenging to assimilate (Fisher & Kennedy, 2019; Levy, 2019; Hines et al., 2021). Here we attempt to provide a brief, top-level intuition, but a detailed exposition lies beyond the scope of this paper. Interested readers are encouraged to consider work by Kennedy (2016); Fisher & Kennedy (2019); Hampel (1974); Ichimura & Newey (2021); Hines et al. (2021); Bickel et al. (1998); Newey (1994; 1990); Chernozhukov et al. (2018); van der Laan & Rubin (2006), and Tsiatis, 2006.

An estimator $\Psi(\hat{\mathcal{P}}_n)$ for an estimand $\Psi(\mathcal{P})$ (for example, the ATE) has an IF, ϕ, if it can be expressed as follows:

$$\Psi(\mathcal{P}) = \Psi(\hat{\mathcal{P}}_n) + \frac{1}{n}\sum_{i}^{n}\phi(y_i, \hat{\mathcal{P}}_n) + o_p(1/\sqrt{n}) \tag{3}$$

where y_i is a sample from the true distribution $\mathcal{P}$, $\hat{\mathcal{P}}_n$ is the empirical distribution or, alternatively, a model of some part thereof (*e.g.*, a predictive distribution parameterized by a NN, or a histogram estimate for a density function, etc.), ϕ is a function (the Influence Function) with a mean of zero and finite variance (Tsiatis, 2006, pp.21), and $o_p\,(1)$ is an error term that converges in probability to zero.

In many cases $\hat{\mathcal{P}}_n$ is not equivalent to the sample distribution, perhaps because some or all of it is being modelled with estimators. As a result, the error does not converge in probability to zero and some residual error remains. The IF ϕ is being used to model the residual bias that stems from the fact that $\hat{\mathcal{P}}_n$ is no longer equivalent to a direct sample from $\mathcal{P}$. We can imagine the sample distribution $\hat{\mathcal{P}}_n$ lies on a linear path towards the true distribution $\mathcal{P}$. This linear model can be expressed using what is known as a parametric submodel, which represents a family of distributions indexed by a parameter ϵ. The direction associated with $\mathcal{P}_\epsilon$ can then be expressed as a pathwise derivative in terms of the function representing our estimand Ψ.

The IF itself is a function which models how much our estimate deviates from the true estimand, up to the error term. If an estimator can be written in terms of its IF, then by central limit theorem and Slutsky's theorem, the estimator converges in distribution to a normal distribution with mean zero and variance equal to the variance of the IF. This is a key result that enables us to derive confidence intervals and perform statistical inference (such as null hypothesis significance testing). Note that the convergence in distribution to a normal distribution does not impact our assumptions about the functional form governing the data generating process itself. In other words, by deriving a normally distributed parameter estimate, we do not sacrifice flexibility in how we model the observed distribution, or flexibility in the functions used to derive the estimates themselves. Appendix B contains information about how to leverage this normality to derive confidence intervals and p-values, and Appendix C derives the IF for general estimands as well as presenting an algorithm for doing so automatically.

4 The MultiStep Update Method

This section presents the first contribution - a new process for updating initial estimators using the influence function. The current work concerns the estimation of the Average Treatment Effect (ATE). For the DAG: $T \rightarrow Y$, $T \leftarrow X \rightarrow Y$ (also see Fig. 1a), the IF for the ATE can be expressed as (Hines et al., 2021; van der Laan & Rose, 2011):

$$\phi_{ATE}(\mathbf{Z}, \hat{\mathcal{P}}_n) = \left(\frac{\delta_{\tilde{t}}(1)}{\hat{G}(\tilde{\mathbf{x}})} - \frac{1 - \delta_{\tilde{t}}(0)}{1 - \hat{G}(\tilde{\mathbf{x}})} \right) \left(\tilde{y} - \hat{Q}(t, \tilde{\mathbf{x}})] \right) + \hat{Q}(1, \tilde{\mathbf{x}}) - \hat{Q}(0, \tilde{\mathbf{x}}) - \Psi_{ATE}(\mathcal{P}). \tag{4}$$

where $\mathbf{Z} = (\mathbf{X}, T, Y)$. In a slight abuse of notation, $\delta_{\tilde{y}}$ is the Dirac delta function at the point at which $y = \tilde{y}$, where $\tilde{y}$ can be a datapoint in our empirical sample (note the shift from specific datapoint y_i to generic empirical samples $\tilde{y}$). In order to evaluate this we need to evaluate it at $\hat{\mathcal{P}}_n$, and we also need plug-in estimators $\hat{G}(\tilde{\mathbf{x}}) \approx f(t|\tilde{\mathbf{x}})$ (propensity score model), and $\hat{Q}(t, \tilde{\mathbf{x}}) \approx \mathbb{E}_{y \sim P(Y|T=t, \mathbf{X}=\tilde{\mathbf{x}})}[y]$ (outcome model).

4.1 Updating Initial Estimators

If we can estimate the IF ϕ then we can update our initial estimator $\Psi(\hat{\mathcal{P}}_n)$ according to Eq. 3 in order to reduce the residual bias which the IF is essentially modeling. To be clear, this means we can improve our initial NN estimators, without needing additional data. We consider four ways to leverage the IF to reduce bias which we refer to as (1) the one-step update, (2) the submodel update (sometimes referred to as a targeted update), (3) our own proposed MultiStep procedure (which we present in this section), and (4) targeted regularization. A brief description of methods (1), (2), and (4) is provided in Appendix D). The first three approaches can be trivially applied to estimators which have already been trained, making them attractive as post-processing methods for improving estimation across different application areas. To illustrate these approaches, we consider the ATE to be our chosen target estimand, the IF for which is defined in Equation 4. Fig. 2 provides an illustration of the components involved in updating the estimator in the context of ATE estimation.

In order to motivate our own proposed MultiStep method, we begin by noting the limitations of the one-step and submodel update processes. In general, these updates are performed only once (Hines et al., 2021; van der Laan & Rose, 2011). In other words, the bias of our initial estimator must be able to be approximated by a linear submodel, such that taking a single step in the direction of the gradient is sufficient. We attempt to improve the empirical robustness of the one-step and submodel update steps by modifying the objective in the update step itself.

4.2 Explicit Optimization Goals

Under the assumptions described above, the one-step and the submodel update approaches yield what is known as the *efficient influence function*. Importantly, the efficient influence function is the one which meets two conditions: (A) the influence function with the smallest variance, and (B) the influence function with a mean of zero, *i.e.*, $\sum_i^n \phi(\mathbf{z}_i, \hat{\mathcal{P}}_n) \approx 0$ (Tsiatis, 2006). Our idea is thus to explicitly specify these two conditions as goals in a continuous optimization problem.

In contrast, the one-step and submodel processes achieve these two conditions *indirectly* by following the Von Mises gradient step and finding the least-squares (or maximum-likelihood) solution to a supervised learning problem, respectively (see Eq. 15 in the Appendix for more details). In particular, the one-step approach updates an initial estimator $\hat{Q}(t, \mathbf{x}_i)$ with the empirical estimate of the influence function (which serves as the gradient), whilst the submodel method updates it by adjusting for a some amount $\hat{\gamma}$ of a biasing quantity known as the clever covariate $H(\mathbf{z}_i)$. We refer to both update processes as 'indirect' because, in contrast, the objective used to find the influence function with the smallest variance and zero mean can be specified explicitly.

We refer to our update variant as MultiStep because whilst it still uses the linear submodel (see Eq. 15 and Appendix D for more details), we optimize the expression 5 below by searching over $\hat{\gamma} \in \Gamma$ such that the mean and variance of the influence function are minimized:

$$\min_{\hat{\gamma} \in \Gamma} \left[\alpha_1 [\widehat{\mathbb{E}}[\phi(\mathbf{z}_i, \hat{\mathcal{P}})]] + \alpha_2 [\widehat{\mathrm{Var}}[\phi(\mathbf{z}_i, \hat{\mathcal{P}})]] \right]. \tag{5}$$

In words, rather than implicitly finding the solution to the IF indirectly via a gradient step (one-step) or a maximum likelihood submodel approach, we explicitly specify that the solution should minimize empirical

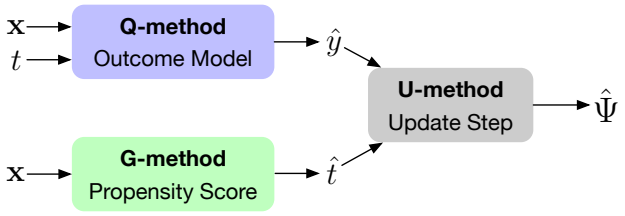

Figure 2: This figure illustrates the components involved in using IFs to improve our estimates of the Average Treatment Effect (ATE), where the ATE is our target estimand Ψ. We combine the output from an outcome model $\hat{Q}$, with a propensity score model $\hat{G}$ and an update step method U. This yields an estimate $\hat{\Psi}$.

approximations (circumflex/hat notation) of both (A) the expectation and (B) the variance of the influence function. The degree to which each of the constraints are enforced depends on hyperparameters $\alpha_1 \in \mathcal{R}^+$ and $\alpha_2 \in \mathcal{R}^+$ which weight the two constraints. In this objective, the parameters $\hat{\gamma}$ are related to the influence function as follows:

$$\begin{aligned} \phi_{ATE}(\mathbf{z}_i, \hat{\mathcal{P}}_n) = H(\mathbf{z}_i, t_i)\left(y_i - \hat{Q}(t_i, \mathbf{x}_i) - \hat{\gamma} H(\mathbf{z}_i)\right) \\ +(\hat{Q}(1, \mathbf{x}_i) + \hat{\gamma} H(\mathbf{z}_i, 1)) - (\hat{Q}(0, \mathbf{x}_i) + \hat{\gamma} H(\mathbf{z}_i, 0)) - \Psi_{ATE}(\hat{\mathcal{P}}_n). \end{aligned} \tag{6}$$

In other words, $\hat{\gamma}$ is the coefficient on the clever covariate which itself represents the quantity which is biasing our principal estimator Q. The objective in expression 5 therefore finds the $\hat{\gamma}$ which corrects for the biasing action of H by minimizing the mean and the variance of the influence function, which itself is computed according to Eq. 6.

One of the consequences of finding the efficient IF is that we also achieve improved model robustness. This is because, when the update step is used to derive an unbiased estimate, we achieve consistent estimation (*i.e.*, we converge in probability to the true parameter as the sample size increases) even if one of the models is misspecified (*e.g.*, the ATE requires both a propensity score model and an outcome model, and thus the IF facilitates *double* robustness). Furthermore, in cases where both models are well-specified, we achieve efficient estimation. It is worth noting, however, that this double-robustness property does not apply to the limiting distribution of the estimates being Gaussian when data-adaptive plug-in estimators are used (Benkeser et al., 2017; van der Laan, 2014). In other words, if only one or both of the two models is/are incorrectly specified, the estimates may not be normally distributed, thus invalidating statistical inference. In our later evaluation, we thus might expect models to fail at achieving *normally distributed* estimates before they fail at yielding *unbiased* estimates. It is possible to extend the framework such that the double robustness property also applies to the limiting normal distribution of the estimates (Benkeser et al., 2017; van der Laan, 2014), but we leave this to future work. For more technical details on the double robustness property see van der Laan & Rose (2011); Hines et al. (2021); Benkeser et al. (2017), and Kurz (2021).

5 The MultiNet Estimator

This section concerns our second contribution - a proposal for a new estimator for the ATE, 'MultiNet', which aims to compete with existing estimation methods described in Section 2. It represents a combination of ideas from two well-known approaches in causal inference - the Super Learner (van der Laan et al., 2007), and CFR-Net, both of which we already mention above.

5.1 Motivation

One of the primary considerations when choosing estimation algorithms/models is whether the estimator can represent a family of distributions which is likely to contain the true Data Generating Process (DGP). Indeed, one of the motivations for semiparametrics is to be able to use non-parametric data-driven algorithms

which have the flexibility to model complex DGPs, whilst still being able to perform statistical inference. If the estimator is functionally misspecified (*i.e.*, misspecified in terms of the functional form used to model the relationships in the DGP - linear, quadratic, spline etc.), then we are unlikely to arrive at an estimator which is asymptotically linear and therefore also amenable to the Influence Function update process. This behooves us to seek estimators which 'let the data speak' (van der Laan & Starmans, 2014).

Consider the Super Learner (SL) (van der Laan et al., 2007), which is an ensemble method especially designed for parameter estimation in the context of causal inference. The SL process involves taking a weighted average of predictions from each candidate learner, and this quantity is taken as the output. The advantage of a SL is that the candidate library includes sufficient diversity with respect to functional form and complexity such that the true DGP is likely to fall within the family of statistical models which can be represented by the ensemble. The motivation for reducing bias resulting from *functional* misspecification is therefore similar to the motivation for influence functions. In both cases, accuracy/precision in estimation is the priority in the domain of causal inference, where the parameters concern critical decision making processes, such as the efficacy of medications.

In contrast with the SL, many methods (including boosted trees, neural networks, or linear regressions) are based on single learners with one outcome prediction. As part of the development and evaluation of Influence Function updating methods for reducing bias and deriving consistent estimators, here we also present a new neural-network based estimator / learning algorithm. Early experimentation highlighted to us that even though NNs are flexible universal function approximators (Hornik, 1993; Hornik et al., 1989), they may nonetheless yield estimators which are not 'good enough' to enable us to leverage their asymptotic properties (such as bias reduction with IFs). In such cases, the IF update may actually *worsen* the initial estimate, pushing us further off course. This problem arose even for simple datasets with only quadratic features. Indeed, the problem with using neural networks for 'tabular' data (as opposed to, say image data) is well known in the machine learning community, and interested readers are directed towards the survey by Kadra et al. (2021). Researchers have, in general, noted that gradient boosted trees (Freund & Schapire, 1997) to consistently outperform neural network based learners (Shwartz-Ziv & Armon, 2021; Kadra et al., 2021; Borisov et al., 2022). However, Borisov et al. (2022) also found that ensembles of boosted trees and neural networks can nonetheless outperform boosted trees alone, and Kadra et al. (2021) found that sufficiently regularized neural networks could yield competitive performance, or even exceed the performance of boosted trees. Thus, in our view the avenues for research into neural network methods for tabular data are still open (and research on the subject continues regardless - see TVAE, CFR-net, and DragonNet, for example). Furthermore, if neural network based methods work well in ensemble combinations with boosted trees, we should attempt to maximise the performance of the neural network learners in order to maximise the performance of the associated ensemble.

5.2 Details

A block diagram for MultiNet is shown in Fig. 3. The method comprises three main elements: (1) a CounterFactual Regression (CFR) network backbone (Shalit et al., 2017), (2) outcome prediction 'taps' at each layer, thereby emulating the outputs from multiple discrete learners in a traditional ensemble and (3) a constrained regression procedure designed to reflect the equivalent 'meta-learner' idea in the Super Learner and which combines the intermediate predictions from each of the layers. We discuss each of these three elements in turn below. The idea therefore represents a combination of two well-known methods in causal inference - CFR-net, and Super Learner.

5.2.1 CFR Backbone

CFR is a popular NN method for causal inference tasks comprised of fully connected layers. It includes separate outcome arms depending on the treatment condition, and forms the backbone of MultiNet. Specifically, for *each consecutive layer* in MultiNet, we predict $y|t, \mathbf{x}$ for $t = \{0, 1\}$ and compute the corresponding layer-wise cross-entropy loss (for a binary outcome). This simulates the multiple outputs of a typical ensemble method - each layer represents a different (and increasingly complex) function of the input, *e.g.*, for a network with L layers, the predicted outcome $\hat{y}$ under treatment $t = 1$ for the l^{th} layer of the network

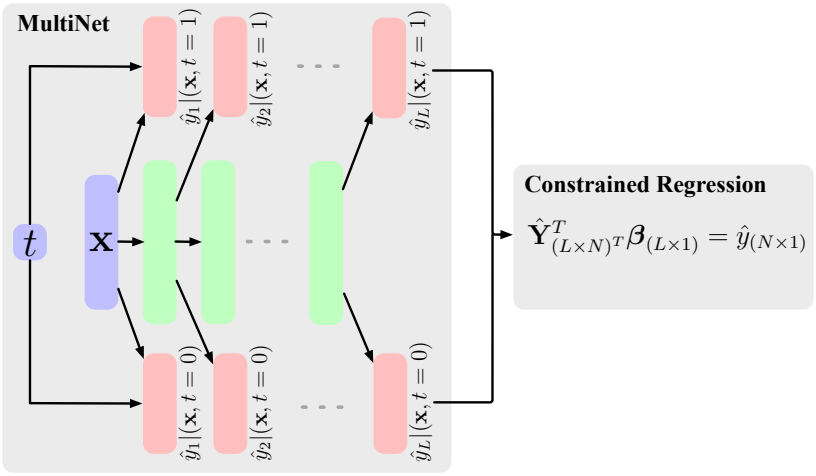

Figure 3: Block diagram for MultiNet. At each layer $l = \{1, ..., L\}$ of the network, the outcome y is estimated using covariates $\mathbf{x}$ (which can include treatment t). The treatment is used to select between two estimation arms. Once the network has been trained, the outcomes from each layer are combined and a constrained regression is performed. The weights $\boldsymbol{\beta}$ in the regression are constrained to be positive and sum to 1. An equivalent single-headed network can be used for the treatment model $\hat{t}|\mathbf{x}$.

is $\hat{y}_l(1) = q_l^1(q_{l-1}^1(q_{l-2}^1(\mathbf{x}, t=1))))$ where q_l^1 represents the l^{th} layer of the network for the CFR arm when treatment is 1. The equivalent for the outcome under no treatment is thus $\hat{y}_l(0) = q_l^0(q_{l-1}^0(q_{l-2}^0(\mathbf{x}, t=0))))$. In our implementation, we use the ReLU activation function.

5.2.2 Ensemble Behavior

Collecting the layer-wise predictions for the outcome $\hat{y}_1^t$ to $\hat{y}_L^t$ together we form a matrix $\hat{\mathbf{Y}}$ which has shape $(L \times N)$ where L is the number of layers and N is the number of datapoints. This idea stems from the one presented as part of the SuperLearner. By consequence of the diversity afforded by multiple learners and the 'meta-learning' ensemble combination of the predictions from these learners, the Super Learner is able to achieve efficient rates of convergence and simultaneously reducing the chances of overfitting.

5.2.3 Ensemble Meta-Learner / Constrained Regression

The constrained regression combines the predictions accumulated from each layer of the 'ensemble', and is only applied after MultiNet has been trained. For each observed treatment condition in the training dataset, we train a meta-learner by solving $\hat{\mathbf{Y}}^T\boldsymbol{\beta} = y$, with layer-wise weights $\boldsymbol{\beta}$ which are constrained to sum to one and be non-negative (similar to a non-negative least squares objective). These weights therefore determine how much importance is given to each layer in the network. We expect, for instance, that for linear datasets the early layers will be assigned a greater weight, whereas for more complex problems, predictions from deeper layers will become more important. Indeed, we observed this behaviour during development, but leave a detailed exploration of this to future work.

For solving the constrained regression we use a SciPy (Jones et al., 2001) non-negative least squares solver. The weights are then used to create weighted averages of the predictions from the consecutive network layers for subsequent predictions, emulating the SuperLearner-style ensemble approach with a single network. Note that one of the strengths of this approach is that the layer-wise outputs and constrained regression techniques can be flexibly applied to other neural network architectures.

5.2.4 Variations

In order to explore the possibility for increased performance, we explore some additional variations on the core idea. Firstly, we allow each layer-wise loss gradient to influence all prior network parameters. This

is similar to the implementation of the auxiliary loss idea in the Inception network (Szegedy et al., 2015), and we refer to this variant as 'MN-Inc'. The second variant involves only updating the parameters of the corresponding layer, preventing gradients from updating earlier layers. We call this variant the 'cascade' approach, and refer to this variant as 'MN-Casc'.

Finally, in order to increase the diversity across the layers and to approximate the diversity of an ensemble, we also explore the use of loss masking. For this, we partition the training data such that each layer has a different 'view' of the observations. The loss is masked such that each layer is trained on a different, disjoint subset of the data. We refer to variants of MultiNet with loss masking as 'MN+LM'.

5.2.5 Objective Function

The objective function of MultiNet is:

$$\mathcal{L} = \min \left[\frac{1}{n} \sum_{i}^{n} \frac{1}{L} \sum_{l}^{L} m_i^l \mathcal{L}_i^l \right], \tag{7}$$

where m_i^l is the mask for datapoint i in layer l, and $\mathcal{L}_i^l$ is the cross-entropy loss for datapoint i, as predicted by layer l. Note that, according to the CFR backbone, there are two prediction arms for each layer, one for when treatment $t = 1$ and one for when $t = 0$. In the variants without loss masking, we set m in the objective function to be equal to 1 for all layers and all datapoints.

It is worth noting that other ensemble neural network approaches exist, such as the BatchEnsemble approach by Wen et al. (2020), and the well-known use of dropout by Gal & Ghahramani (2016). Whilst we incorporate dropout into MultiNet's training procedure, we do not use the dropout itself as a means to create an ensemble, although this represents an interesting opportunity for further work. We instead explore 'tapping' each subsequent layer in a neural network to collect intermediate outcome predictions, and combining them with constrained regression the manner of the Super Learner. We indeed observe improvements in performance over the CFR-Net backbone, highlighting that this simple modification can provide enough diversity to yield competitive, ensemble-style performance.

6 Evaluation

This section concerns our third contribution - a comprehensive evaluation of the performance of estimators and influence functions for the task of ATE estimation. The main results of this evaluation are given in Sec. 6.3, and the results of a Shapley value analysis (Shapley, 1953; Lundberg et al., 2020) are presented in Sec. 6.4). Additional experimental results and discussion can be found in the appendices.

6.1 Open Questions

So far, we have presented the relevant background for causal inference and IFs, proposed a new MultiStep update process and proposed a new NN based estimator called MultiNet. The following open questions remain: (1) Can estimation methods be improved using the one-step, submodel, MultiStep (ours), or targeted regularization approaches? (2) How do various different outcome, propensity score, and update step methods compare? We aim to answer these questions through an extensive evaluation of different methods (Sections 6.3 and 6.4). In particular, we examine the performance of the different approaches in terms of (a) precision in estimation, (b) robustness, and (c) statistical inference (normality of the distribution of estimates). We use these open questions to inform the design of our experiments, which are described below.

6.2 Experimental Setup

6.2.1 Methods, and Evaluation Criteria

We evaluate a number of different methods in terms of their ability to estimate the ATE. A summary of the complete set of methods explored as part of the evaluation is shown in Table 1. As described above, we

are interested in three properties relating to performance: estimation precision, robustness, and normality. Estimation precision is evaluated using mean squared error (MSE) calculated as $r^{-1}\sum_i^r[\hat{\tau}_i - \tau]^2$ where $r = 100$ is the number of simulations, and the standard error (s.e.) of the ATE estimates is computed as the standard deviation of $\hat{\tau}$. Note that this is not the same as estimating the precision in the Conditional ATE, because the indexing occurs over simulations (*i.e.* τ_i is an average treatment effect, estimated for an entire empirical sample), not individual participants. The MSE was chosen for its sensitivity to large errors (arguably important in the application domain of causal inference), and because it was found that it provided a more informative spread of results than the root-MSE. Robustness will be evaluated by comparing initial estimators that fail to exhibit the desired properties, with the results once these estimators have been updated. For normality, we examine the empirical distribution of the estimates. Using these distributions, we provide p-values from Shapiro-Wilk tests for normality (Shapiro & Wilk, 1965). Doing so provides an indication of the estimator's asymptotic linearity and whether the IFs are facilitating statistical inference as intended.

6.2.2 Data

Recent work has highlighted the potential for the performance of modern causal inference methods to be heavily dataset-dependent, and has recommended the use of bespoke datasets which transparently test specific attributes of the evaluated models across different dimensions (Curth et al., 2021b). We therefore undertake most of the evaluation using variants of a DGP which we refer to as the LF-dataset and which has been used for similar evaluations in the literature (Luque-Fernandez et al., 2018). We also evaluate using the well-known IHDP dataset (Hill, 2011; Dorie, 2016), as well as a generalized version of the LF-dataset.

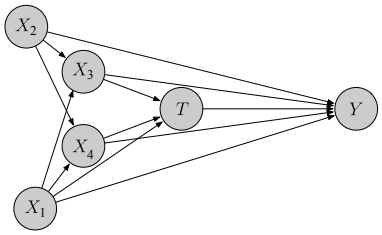

Figure 4: Graph for the 'LF' dataset used by Luque-Fernandez et al. (2018).

LF Dataset Variants: The initial and original LF-dataset variant, (v1), models 1-year mortality risk for cancer patients treated with monotherapy or dual therapy. One motivation for starting with this DGP is that its polynomial functional form is not sufficiently complex to unfavourably bias the performance of any method from the start. The dataset also exhibits near-positivity violations, and will therefore highlight problems associated with the propensity score models which are necessary for the update process. We also adjust the level of non-linearity in order to assess the robustness of each method to increased complexity. Accordingly, we introduce an exponential response into the potential outcome under monotherapy ($t = 1$) for the second variant (v2). Our LF-datasets comprise 100 samples from a set of generating equations. Both variants are designed to highlight problems which may arise due to near positivity violations.

The graph for the synthetic 'LF' dataset used in work by Luque-Fernandez et al. (2018) is given in Fig. 4. The DGP is based on a model for cancer patient outcomes for patients treated with monotherapy ($t = 1$) and dual therapy ($t = 0$). We create two version of this dataset (v1) and (v2), where the data generating process of the second variant includes an exponential non-linearity in the outcome model. The two variants are designed to yield near positivity violations in order to highlight weaknesses in methods which depend on a reliable propensity score model.

We also create a generalized version of the LF (v2) dataset which we refer to as 'General', or 'Gen' for short, in the experimental results. The functional form underlying the key relationships in this data generating process are the same as LF (v2), except we include more structural complexity. The corresponding graph can be found in Fig. 5. Specifically, we include a collider structure (C), a mediator (M), two instrumental and two risk variables (I_1, I_2, R_1, and R_2, respectively), and a new confounder substructure ($T \leftarrow X_5 \leftarrow X_1 \rightarrow Y$).

Q Method	G Method	U Method	Datasets	Evaluation Criteria
Linear/Logistic Regression (Q-LR)	Linear/Logistic Regression (G-LR)	OneStep (U-ones)	LF (v1) n={500, 5000, 10000}	Mean Squared Error (MSE)
SuperLearner (Q-SL)	SuperLearner (G-SL)	Submodel (U-sub)	LF (v2) n={500, 5000, 10000}	Shapiro-Wilk Test (p)
CFR (Q-CFR)	CFR (G-CFR)	MultiStep (U-multi)	Gen n={500, 5000, 10000}	Standard Error of Estimation (s.e.)
MultiNet (Q-MN) + variants	MultiNet (G-MN) + variants	Targeted Regularization (treg)	IHDP n=747	
TVAE (Q-TVAE)	P-learner (G-P)	None (U-Base)		
DragonNet (Q-D)	DragonNet (G-D)			
S-learner (Q-S)				
T-learner (Q-T)				
DML* (DML)				

Table 1: A summary of all variants and metrics explored as part of the evaluation. Q-methods concern the model for the outcome, G-methods concern the propensity score models for the treatment assignment, and U-methods concern the manner in which the influence function is used to update the initial estimates. * Note that the Double Machine Learning (DML) method is treated on its own, without combination with G- and U-Methods.

For estimation our adjustment set comprises all the X variables, even though it is possible (a) to include the risk variables, and (b) to omit either X_5 or X_1.

For LF (v1), LF (v2), and Gen, we create further variants with different sample sizes $n = \{500, 5000, 10000\}$ in order to explore sensitivity to finite samples. The generating equations for these variant datasets (including propensity scores for v1 and v2) are given in Appendix E.

IHDP: The final dataset comprises 100 simulations from the well-known IHDP[5] dataset. We use the version corresponding with usual setting A of the NPCI data generating package Dorie, 2016 (see Shi et al., 2019; Shalit et al., 2017, and Yao et al., 2018) and comprises 608 untreated and 139 treated samples (747 in total). This variant actually corresponds with variant B from Hill (2011). There are 25 covariates, 19 of which are discrete/binary, and the rest are continuous. The outcome generating process is designed such that under treatment, the potential outcome is exponential, whereas under no treatment the outcome is a linear function of the covariates (Curth et al., 2021b). This dataset represents a staple benchmark for causal inference in machine learning. However, it is worth noting that recent work has shown it to preferentially bias certain estimators (Curth et al., 2021b), so we include this dataset for completeness but discount our interpretation of the results accordingly.

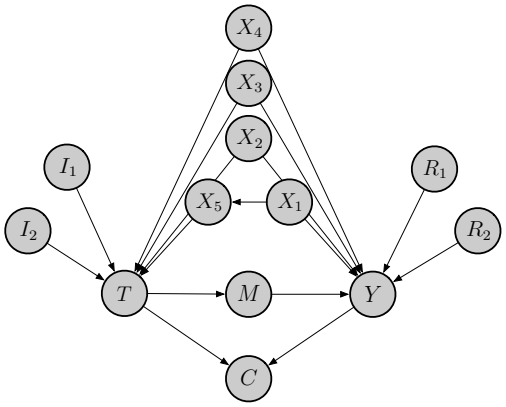

Figure 5: A generalized version of the LF dataset which we refer to as the 'Gen' dataset in the experimental results.

6.2.3 Algorithms/Estimators

The training details and hyperparameters are presented in Appendix F. For the outcome model Q we compare linear/logistic regression (LR); a Super Learner (SL) comprising a LR, a LR with extra quadratic features, a Support Vector classifier, a random forest classifier (Breiman, 2001), a nearest neighbours classifier

[5]Available from `https://www.fredjo.com/`

(Altman, 1992), and an AdaBoost classifier (Freund & Schapire, 1997); an implementation of the backbone to CounterFactural Regression network (without the integral probability metric penalty) (Shalit et al., 2017) (CFR); DragonNet (D) with and without targeted regularization (Shi et al., 2020); TVAE (Vowels et al., 2021) (which includes targeted regularization); T-learner (T) (Kunzel et al., 2019) with a gradient boosting machine (Friedman, 2001); S-learner (S) (Kunzel et al., 2019) with a gradient boosting machine (Friedman, 2001); and our MultiNet (MN) variants (*MN-Inc*, *MN-Casc*, *MN-Inc+LM*, *MN-Casc+LM*). When estimating the IF of the ATE, we also need estimators for the propensity score / treatment model, which we refer to as G. For this we use LR and SL, ElasticNet 'P-learner' (Zou & Hastie, 2005), DragonNet, as well as CFR and MN. The latter two NN methods must be modified for this task, and for this we simply remove one of the outcome arms, such that we can estimate $t|\mathbf{x}$. This seletion of algoritms was chosen to represent a suitable diverse set of common yet weak learners (*e.g.*, logistic regression), modern/well-known neural network causal inference methods (*e.g.*, CFR-net), neural network methods which already incorporate semi-parameteric techniques (DragonNet and TVAE), and recent and/or popular methods proposed in the causal inference or biostatistics literature (*e.g.* Super Learner, T- and S-learners). Finally, we also include the Double Machine Learning (DML) method Chernozhukov et al. (2018) as implemented in the DoubleMLIRM class in the DoubleML python library Bach et al. (2022). The DML estimator uses a stacking classifier comprising a LR, a random forest, a nearest neighbours algorithm, a support vector machine, and an AdaBoost algorithm (and is thus similar in its adaptability to the Super Learner, as specified above). Note that the DML method does not need to be 'updated' (it is already a doubly-robust estimator), and as such we consider it separately and provide results for DML, and compare it with the other methods, in the appendices.

6.2.4 Update Steps

We evaluate the onestep (U-ones), submodel (U-sub), MultiStep (U-multi), and targeted regularization (Treg) approaches to the update process.

The MultiStep update variants are optimized using the Adam (Kingma & Ba, 2017) optimizer. For small datasets ($n < 1000$) we undertake full gradient descent (*i.e.*, using the full data), and for larger datasets we use stochastic mini-batch gradient descent. The batch size for datasets with a sample size $n > 1000$ is set to 500, we undertake 4000 steps of optimization, and the learning rate for the Adam optimizer is set to 5×10^{-4}. The MultiStep objective has hyperparameters α_1 and α_2 which weight the constraints in the objective (expectation and variance of the influence function, respectively). We set both to one.

6.3 Results

Given the large number of combinations in a full-factorial design (approximately 5000 results), we undertake an initial set of experiments to narrow down the evaluation space to focus on the most competitive methods. With this 'shortlist', we investigate the contribution of each Q-, G-, and U-method across the 7 different dataset variants. The results for this initial evaluation are presented in Appendix H. In summary, we identified that CFR did not perform sufficiently well to warrant further investigation (note, as mentioned above, that our implementation of CFR does not utilize the full objective function presented by the original authors, only their proposed architectural foundation). Furthermore, the best performing MN variant was MN-Inc+LM, and we use this variant for the subsequent analyses. Finally, targeted regularization was inconclusive. However, previous work has identified its potential to improve DragonNet and TVAE (Shi et al., 2020; Vowels et al., 2021) and so we restrict the application of targeted regularization to these methods only, in the main evaluation presented below. Note that even following this initial evaluation / shortlisting process, the number of results remains large. We have attempted to summarize them in Figs. 6-11, but include further results in the Appendix in Table 4, Table 5 and Figs. 20-29.

Whilst it is possible and potentially helpful to simply present the full set of results, the overwhelming number of combinations does little to help us understand whether the use of particular Q-, G- or U-methods are more or less likely to improve or worsen the performance in any particular combination on any particular dimension/metric. Therefore, Figs. 6, 7, and 8 provide results for $p(O|M) = p(M|O)p(O)/p(M)$ across the LF and General dataset variants for MSE and s.e. Here, M is the method, and O is the quantile (we split into 5 quantiles) for MSE and s.e., respectively. In words, the associated plots provide an estimation for the

probability of achieving a performance result in each quantile O, for a given method M, thereby providing a means to directly assess the relative performance of each Q-, G-, and U-method. For instance, we can split the MSE results into equal probability quantiles, and count the number of times the use of each outcome, propensity score, and update method results in a performance which falls into each of these quantiles. Using Bayes rule we get an estimate for the probability of achieving results in a particular quantile (*e.g.*, the best performing methods fall in the zeroth quantile of MAE results), given a particular choice of method. Using these calculated probabilities, we also select all results from the best quantile, and see how the performance shifts over different sample sizes. Note that because these results are based on a rank ordering, it is not possible to judge absolute performance, only relative performance. Indeed, the purpose of the initial results above was to use the absolute performance as a way of shortlisting the methods so that a more comparative evaluation could be undertaken using the more competitive methods.

To evaluate the normality of the estimates, after calculating the p-value from the Shapiro-Wilk test, we calculate the proportion of each Q-, G-, and U-methods which yield normally distributed estimates ($p > 0.01$). For example, if a particular Q-method has a high 'probability of normality' according to *e.g.* Fig. 10, this means that a large proportion of the results yielded normally distributed estimates.

In the following Sections 6.3.1-6.3.7 we review the performance of each method for each of the three performance metrics in turn.

Note: When interpreting the results shown in Figs. 6-10, it may be useful to recognise the 'ideal/desired' curve as one which takes on high values on the left-hand-side, representing a high-probability of the method yielding results in the top quantile(s). In contrast, curves which take on high values on the right hand side represent those with a high probability of yielding poor results in the lower quantile(s), relative to the other methods.

6.3.1 Q-Methods - MSE

Beginning with Fig. 6, the results for the outcome model Q-methods on the LF dataset variants are shown in the first column. In Fig. 6a we see that our Q-MN achieves the highest probability of being in the best quantile for MSE when used as an outcome model Q for **LF (v1)** $n = 500$, followed closely by Q-LR and Q-SL, and Q-TVAE and Q-D in the second-best quantile. In contrast, Q-D without targeted regularization, Q-T, and Q-S all had higher probabilities of yielding results in the later quantiles (*i.e.*, their performance was worse). Increasing the sample size to $n = 5000$, and considering Fig. 6d, we see similar results, with MN again yielding the highest probability of the achieving the best results, with Q-D, Q-S, Q-T, and Q-D without targeted regularization performing the worst. Finally, for LF (v1) $n = 10000$, we see in Fig. 6e that Q-MN is superseded by Q-LR and Q-SL, followed by Q-TVAE. Q-T, Q-S, and Q-D perform poorly again.

These results suggest that Q-LR and Q-SL perform consistently well over different sample sizes, and that Q-MN can perform well in small sample sizes, but may start to overfit as the sample size increases. Recall that the task of causal inference is different from the typical supervised learning task, and more data does not necessarily imply that it is easier to estimate the difference between two response surfaces, particularly when this difference (which is the treatment effect) is of low-complexity relative to the response surfaces themselves.

Now consider Figs. 6(j, m, p) for **LF (v2)**, which introduces additional non-linearity into the outcome model. We initially observe similar results for $n = 500$ in 6j, with Q-MN, Q-LR, and Q-SL achieving the best results, and Q-S, Q-D without targeted regularization, and Q-T populating the later quantiles. Increasing the sample size to $n = 5000$, we see in Fig. 6m that Q-TVAE now becomes the most likely to yield the best results, followed by Q-SL and, interestingly, Q-D without targeted regularization. Q-D, Q-T, and Q-S, however, still perform poorly. Finally, for $n = 10000$, we see Q-TVAE maintain the lead, once again followed by Q-SL. The worst performers were, again, Q-D, Q-S, and Q-T. This suggests once again that Q-SL provides consistent performance across sample sizes, and that Q-MN is a good option for smaller sample sizes.

For the **Gen** dataset, we see from Figs. 8(a,d,g) that the results are again quite mixed, although there is some indicate that Q-LR, the logistic regressor, performs reasonable well, and, for instance, that Q-TVAE performs poorly as the sample size increases.

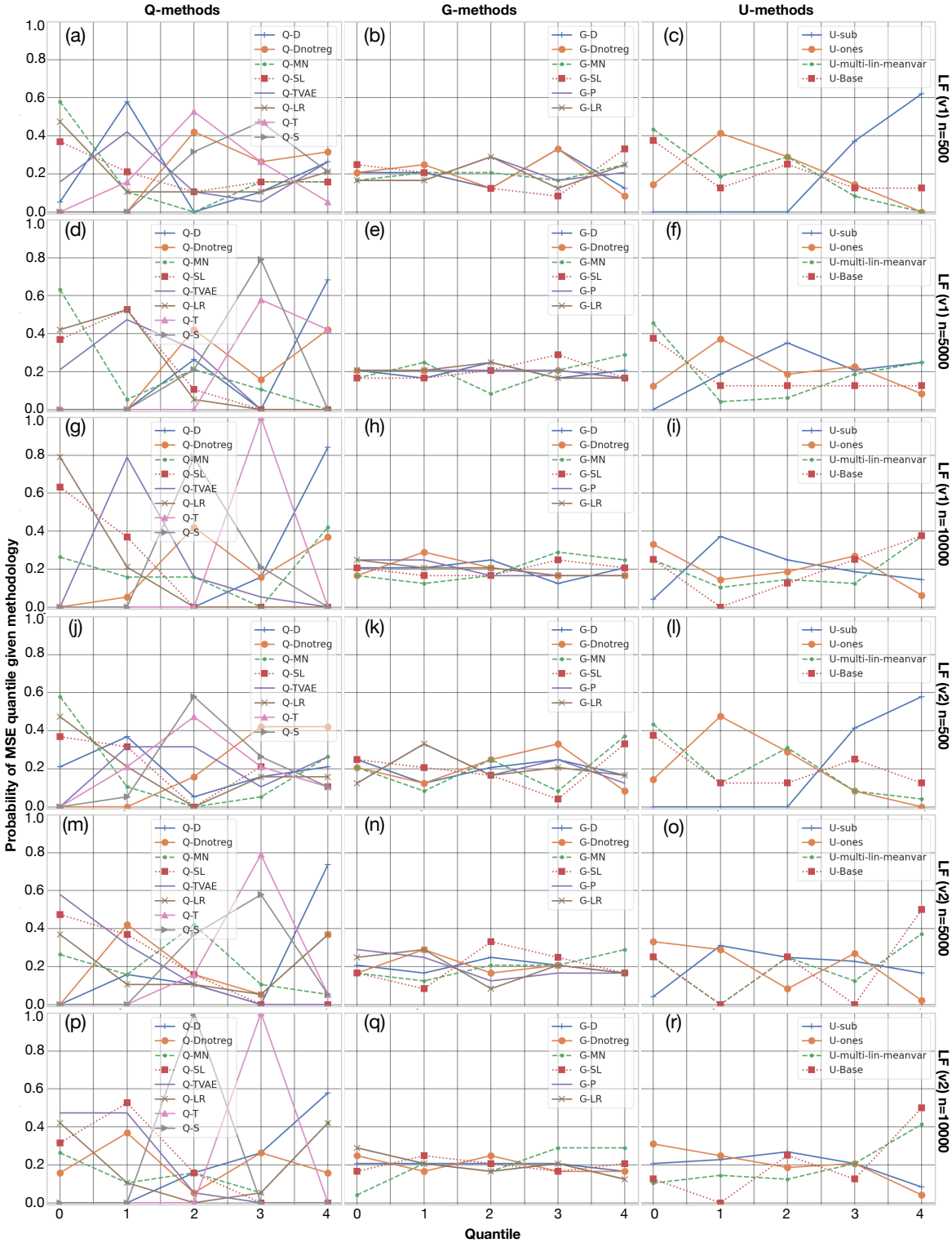

Figure 6: After recording the MSE for each Q (outcome), G (propensity score), and U (update step) method combination, we rank order them (from lowest to highest MSE), and calculate $p(O|M)$ where O is the MSE quantile, and M is the method. For 5 quantiles, this enables us to find *e.g.*, the probability of getting a MSE in the best quantile given a particular method $p(O = 0|M = m)$. If a method performs well, we expect to have high probability of achieving an MSE in the top two quantiles.

For the **IHDP** dataset, we use a fixed sample size of $n = 747$, and the results are shown in Fig. 9. Here it can be seen that Q-T and Q-TVAE achieve the best results, followed by Q-S and Q-MN. The worst performer was Q-LR. These results are consistent with previous work which highlighted state-of-the-art performance of TVAE on IHDP (Vowels et al., 2021). Similarly, the fact that LR did so poorly possibly highlights the non-linearity of the data generating process for IHDP. The fact that Q-S and Q-T did so well is surprising given their relatively poor performance on the LF datasets described above. Such dataset dependence for the performance of causal estimators has also been previously noted by Curth et al. (2021b).

6.3.2 G-Methods - MSE

The MSE results for the propensity score G-methods can be seen in the second column of Figs. 6 and 9, and in plots (b, e and h) in Fig. 8. Interestingly, there is very little dependence between the performance of the different methods. Arguably, there is some evidence that G-MN performs slightly worse than other methods in Fig. 6q, and that G-D performs worse in Fig. 9b but the differences are not convincing. For instance, G-D performs quite well in Fig. 8(b) on the 'Gen' dataset when $n = 500$, but not noticeably well as sample size increases. This suggests that, at least in our experiments, the MSE results are relatively robust to the choice of propensity score model.

6.3.3 U-Methods - MSE

The MSE results for the update U-methods are shown in the third column of Fig. 6 for the LF datasets. In Fig. 6c we see that the U-Base model and the U-multi update methods perform the best, with the U-ones model close behind. The submodel update is more likely to be the lower quantiles. As the sample size increases to $n = 5000$ and $n = 10000$ in Figs. 6f and 6i we see the U-sub and, to a lesser extent, the U-ones performance shift. This behaviour has been observed before in work by Neugebauer & van der Laan (2005), who found that the performance of U-ones increased with sample sizes. Indeed, their own proposition for a multistep update process also performed more consistently in small samples, as does our U-multi. Similar patterns of performance are seen in Figs. 6l, 6o, and 6r for the LF (v2) dataset.

In Fig. 8 we again see mixed results on the 'Gen' dataset. Whilst, in Fig. 9, there is some indication that the U-sub and U-ones performed approximately equally well and better than U-multi and U-Base on the IHDP dataset, the results are still far from indicating a clear pattern in performance.

6.3.4 Q-Methods - s.e.

The standard error (s.e.) results are shown in Fig. 7 and the bottom row of plots in Fig. 9. Starting with Fig. 7a, we find the methods yielding the tightest distribution of estimates for the LF (v1) dataset $n = 500$ are Q-MN, Q-LR, and Q-TVAE, followed by Q-D, Q-SL, and Q-T. At the lower end we find Q-D without targeted regularization, and Q-S. As the same size increases to $n = 5000$ Q-MN provides estimates which are even more likely to be the tightest, followed again by Q-LR, Q-TVAE, and Q-SL. Q-D is not far behind, with Q-S, Q-T, and Q-D without targeted regularization performing the worst. With $n = 10000$, Q-MN is overtaken by Q-LR in terms of the tightness of the estimation, which is understandable given that Q-MN has a large number of hyperparameters (Q-LR has none), which contributes to variability in performance. Q-TVAE once again follows closesly behind, with the worst performers being Q-D without targeted regularization, Q-S, and Q-T. Interestingly Q-MN exhibits a rise in the probability of being one of the worst performers, suggesting that there may exist better or worse combinations of G- and U-methods with Q-MN. Once again, it is worth consulting the full set of rank-ordered results in the Appendix. With the results for LF (v2) in Figs. 7j, 7m, and 7p we see a similar pattern of results, in spite of the introduction of additional non-linearity in this dataset variant.

There is some indication in Fig. 8(j,m,p) that Q-Sl and Q-LR perform well, whilst Q-D and Q-T have a higher chance of yielding results in the less competitive quantiles. However, the results are quite mixed, once again. Finally, for the IHDP results in Fig. 9d we see Q-LR and Q-SL provide the tightest estimates, followed by Q-MN, Q-D without targeted regularization, then Q-TVAE, Q-S, and Q-T.

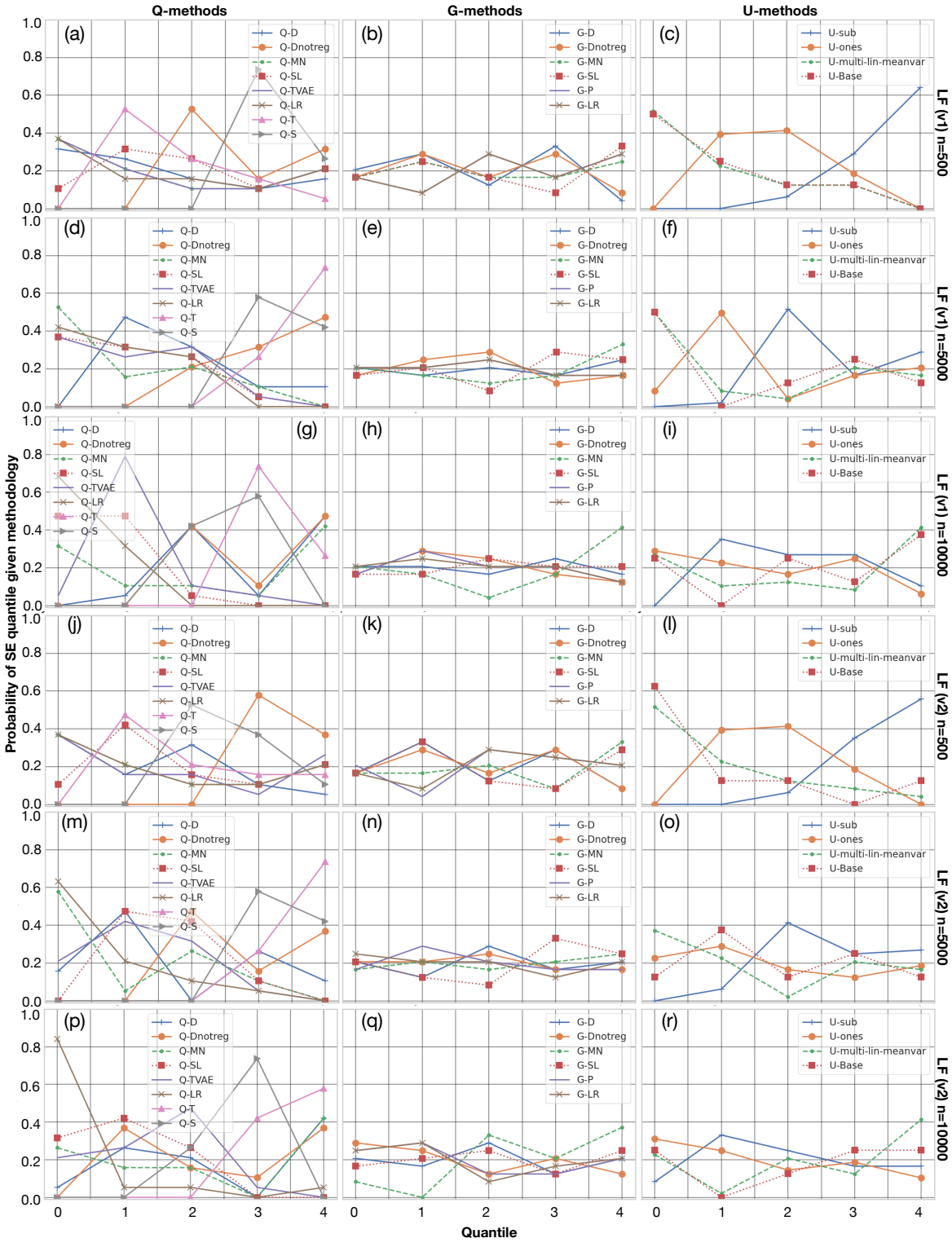

Figure 7: After recording the standard error (s.e.) of the 100 ATE estimates for each LF dataset and for each Q (outcome), G (propensity), and U (update step) method combination, we rank order them (from low to high), and calculate $p(O|M)$ where O is the quantile, and M is the method. This enables us to find the probability of getting a s.e. in the best quantile given a particular method $p(O = 0|M = m)$. If a method performs well, we expect to have high probability of achieving an s.e. in the top two quantiles.

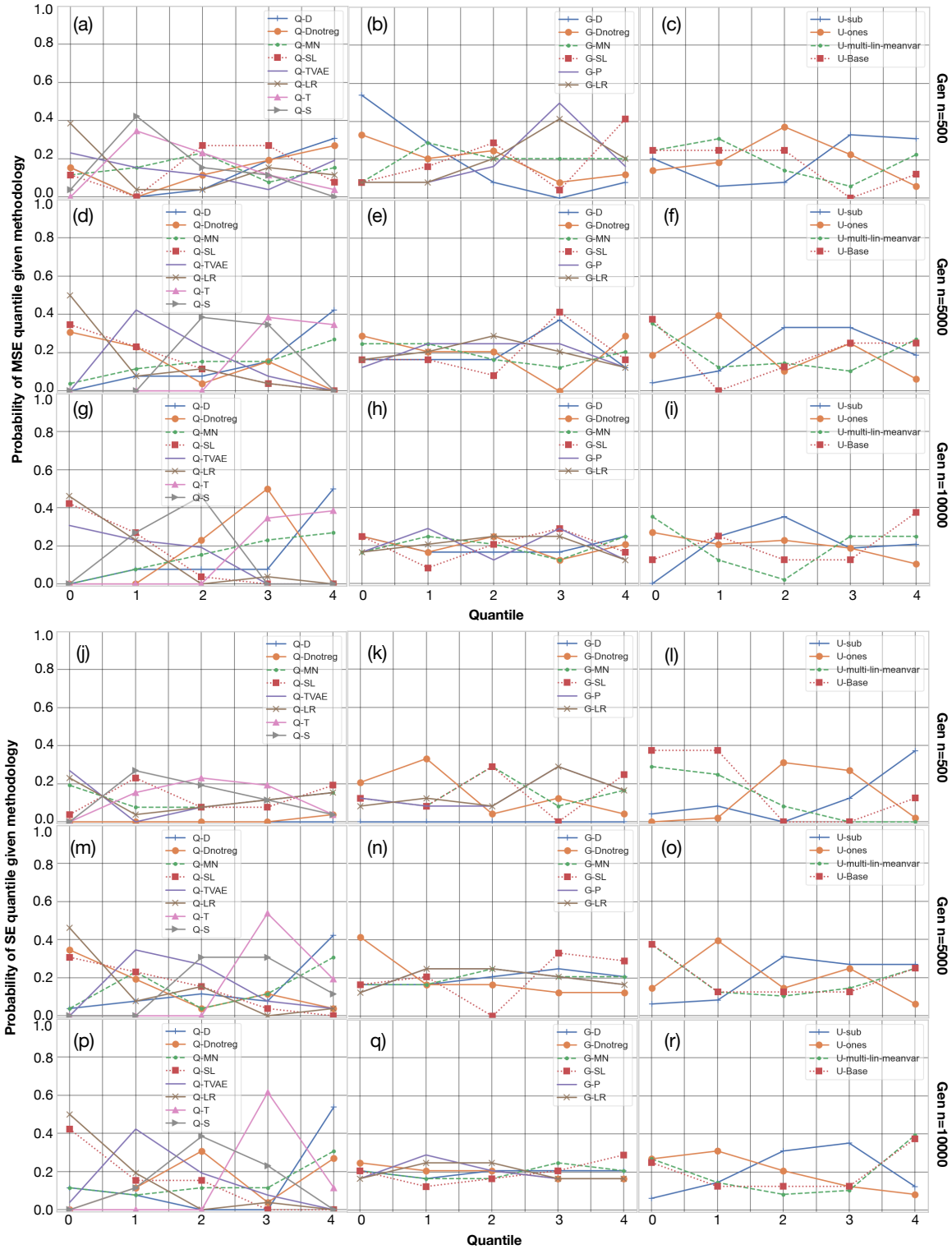

Figure 8: After recording the MSE and standard error (s.e.) for each Q (outcome), G (propensity score), and U (update step) method combination on the 'Gen' dataset variations, we rank order them (from lowest to highest MSE), and calculate $p(O|M)$ where O is the MSE quantile, and M is the method. For 5 quantiles, this enables us to find *e.g.*, the probability of getting a MSE or s.e. in the best quantile given a particular method $p(O = 0|M = m)$. If a method performs well, we expect to have high probability of achieving an MSE in the top two quantiles.

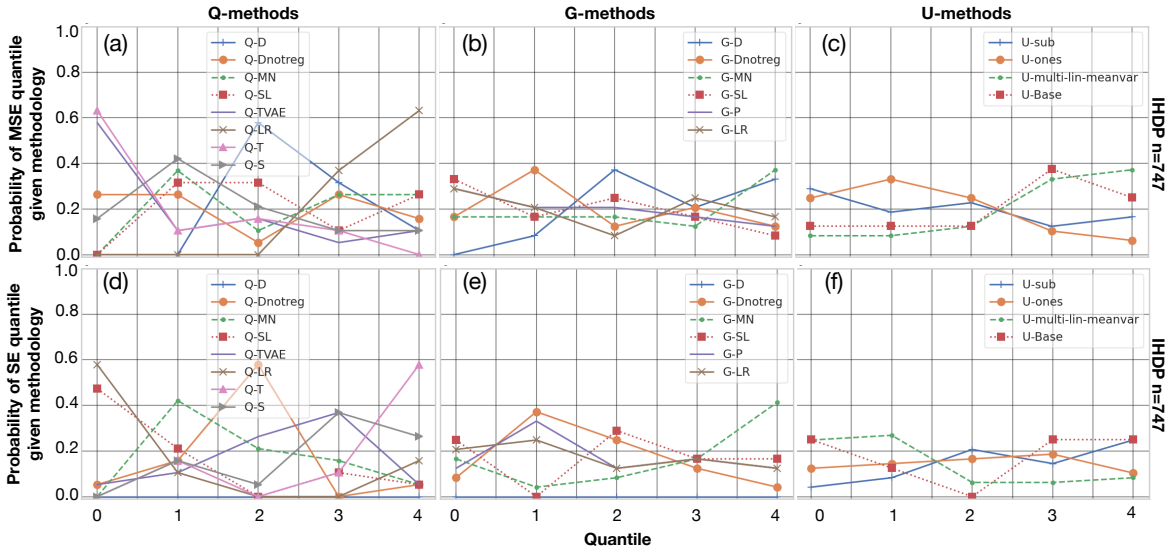

Figure 9: After recording the MSE and standard error (s.e.) of the 100 ATE estimates for the IHDP dataset and for each Q (outcome), G (propensity score), and U (update step) method combination, we rank order them (from lowest to highest), and calculate $p(O|M)$ where O is the MSE (top row) or s.e. (bottom row) quantile, and M is the method. For 5 quantiles, this enables us to find the probability of getting a MSE or s.e. in the best quantile given a particular method $p(O = 0|M = m)$. If a method performs well, we expect to have high probability of achieving an MSE and/or s.e. in the top first or second quantiles, and a low probability of achieving an MSE and/or s.e. in the last quantiles. Best viewed in colour.

6.3.5 G-Methods - s.e.

The s.e. results for the choice of propensity score G-method can be found in the central columns of Fig. 7 and Fig. 9e, and Figs. 8(k,n,q). As was found for the MSE results, the choice of G-method was not decisive, besides the poor performance of G-MN for IHDP dataset, and for the $n = 10000$ LF datasets. It is interesting to again find that the choice of G-method does not have a strong impact on the tightness of the estimates.

6.3.6 U-Methods - s.e.

The s.e. results for the choice of update U-method are presented in the right-hand column of Fig. 7 and Fig. 9f. In contrast to the choice of G-method, the choice of U-method had a significant impact on the tightness of the associated estimates, and the pattern of performance is similar to the pattern for MSE. For low sample sizes, it can be seen from both Figs. 7c and 7l that the tightest estimates are achieved using U-multi and U-Base, with U-sub yielding the least tight estimates. Increasing the sample size shifts the performance of U-sub and U-ones, making them competitive with the other methods.

For the 'Gen' dataset, Fig. 8(l,o,r) indicate that the base models perform well, whilst U-sub seems to be one of the least good performers. For the IHDP dataset, it can be seen in Fig. 9f that the choice of U-method had little impact on the tightness of the estimates, but the best performers were U-Base (*i.e.*, no update), and U-multi.

6.3.7 Q-, G-, U-Methods - Normality

The results evaluating the normality of the estimates are provided in Fig. 10 for the LF and Gen dataset variants, and Fig. 11 for IHDP. For the LF and Gen datasets, plots (a-i) provide the proportion of results from the respective method which yielded normally distributed estimates ($p > 0.01$) for each of the different dataset sizes $n = \{500, 5000, 10000\}$. In Fig. 10a it can be seen that most Q-methods performed well across all sample sizes with LF (v1), with the exception of Q-D which was less likely to yield normally

distributed estimates, and we observe a drop in performance for Q-MN as sample size increases. Once again, and as indicated by Fig. 10b, the choice of G-method was not found to impact the likelihood of normally distributed estimates. Figs. 10c indicates that the likelihood of U-Base and U-multi yielding normally distributed estimates dropped slightly with sample size, with U-ones yielding consistently normally distributed estimates regardless of sample size.

For LF (v2), the results in Figs. 10d-10f indicate more variability, possibly as a result of the additional non-linearity in the outcome model. When $n = 500$, the outcome Q-method most likely to yield normally distributed estimates was Q-T, followed by Q-D and Q-MN. However, for $n = 5000$ and $n = 10000$, the only methods not yielding consistently normally distributed results were Q-D and Q-MN. For the propensity score G-methods, the method most likely to yield normally distributed results with $n = 500$ was G-LR, followed by G-SL. The other methods did not perform well until the sample size was increased to $n = 5000$ or $n = 10000$ for which all methods performed equally well. For the U-methods, the best performing result across all sample sizes was U-ones, followed by U-sub, U-multi, and finally U-base.

For the Gen dataset, the results in Figs. 10g-10i indicate that Q-MN and Q-D were two of the methods least likely to yield normally distributed estimates, particularly as the sample size increased. On the other hand, the G-LR performed well as a propensity score method in this regard, and our MultiStep worked well as an update method.

Finally, the likelihood of achieving normally distributed estimates for the IHDP dataset are shown in Fig. 11. The sample size is fixed for this dataset, and the results for the Q-, G-, and U- methods are presented together (hence the different graph format). It can be seen that Q-D provided the highest likelihood of normally distributed estimates, with the other methods yielding comparable (and low) likelihood. Similarly, G-D yielded the highest likelihood of normally distributed estimates, with the other G-methods being relatively equal (and low). Finally, none of the U-methods provided a high likelihood of normally distributed estimates.

6.3.8 Summary of the Main Evaluation

Note that in some figures, certain methods may not have a monotonic probability which starts high and ends low, or vice versa. For example, in Fig. 6p, Q-LR has a u-shaped probability, suggesting that for some combinations of Q-LR with certain other G- and U-methods, its performance is good, and with others it is poor. In such cases it may be more informative to consult the full results in the Appendix, to attempt to understand whether there is any particular combination dependence.

Overall, the results were quick mixed across datasets and metrics, highlighting that no single method outperforms all others across datasets of different functional form, structure, and sample size.

MSE Summary: Our Q-MN performed well on the LF datasets, particularly in smaller samples. We found that both Q-LR and Q-SL also performed consistently across the different sample sizes, even with the introduction of non-linearity with LF (v2). Indeed, with the introduction of this non-linearity, we found Q-TVAE to yield good performance, and this competitive edge held up with IHDP as well. We did not find that the choice of G-method had a large impact on the results, although G-MN tended to do slightly worse. With smaller sample sizes $n = \{500, 5000\}$ and/or simpler datasets (LF v1), our U-multi performed the best as an update method. As sample size increased, we found that the onestep U-ones became the best performer, and similar behaviour has been found in other work (Neugebauer & van der Laan, 2005). For more complex datasets like IHDP, we found that U-ones and U-sub performed well.

Standard Error Summary: Once again, our Q-MN provided the tightest estimates, and did so consistently over all sample sizes and datasets except IHDP. The next best and most consistent estimator (including good performance on IHDP) in terms of the tightness of its estimates, was Q-SL. Once again, we did not find that the choice of G-method had a large impact on the results, but G-MN tended to do slightly worse than others. Our U-multi yielded consistently tight estimates across all datasets (including IHDP), although in general, the base models (without update steps) also performed well in this regard. As with the MSE results, U-ones and U-sub performed more competitively as the sample size increased.

Normality Summary: The choice of Q-method did not have a big impact on the likelihood of normally distributed estimates for the LF datasets, although Q-D performed poorly, and the performance of Q-

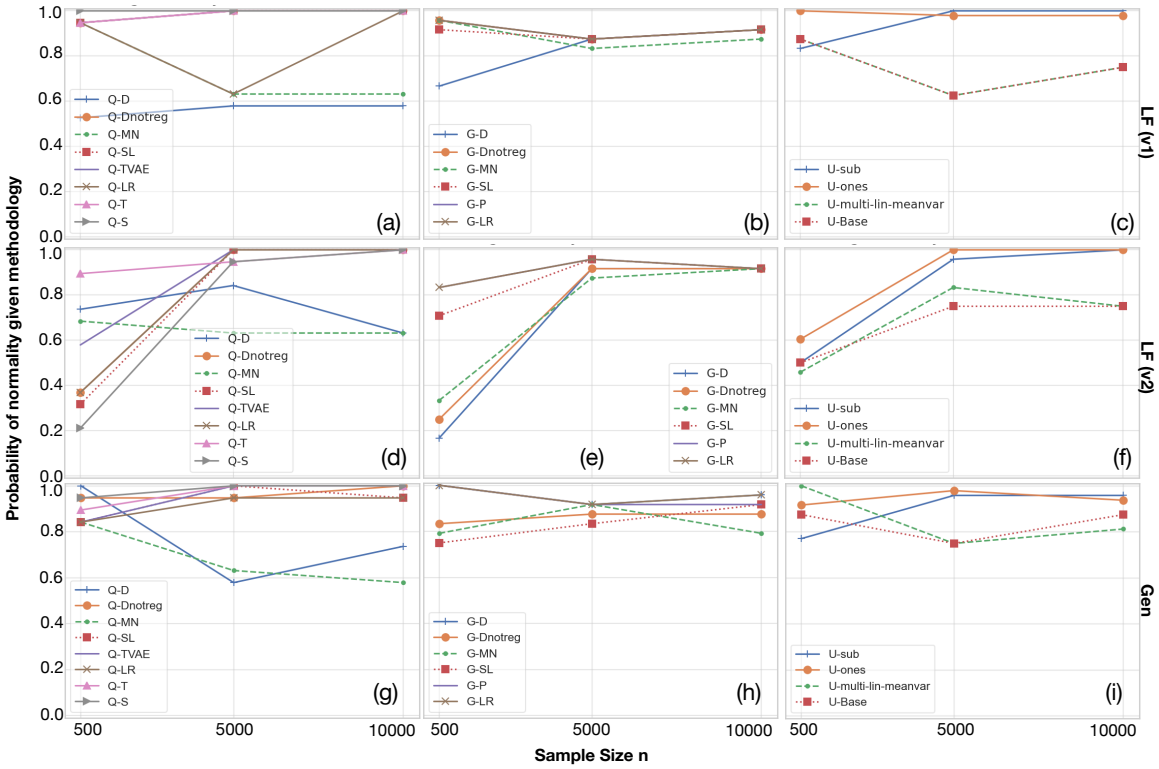

Figure 10: Probability of $p > 0.01$ for the Shapiro-Wilk test of normality for each Q (outcome), G (propensity score), and U (update step) method with the LF and general 'Gen' datasets $n = \{500, 5000, 10000\}$. Because we undertook all combinations of G and U, each point represents a marginalization over the other dimension(s). For instance, for the Q methods, Q-D (DragonNet) is an average probability result when combining Q-D with all possible other G and U methods. Best viewed in colour.

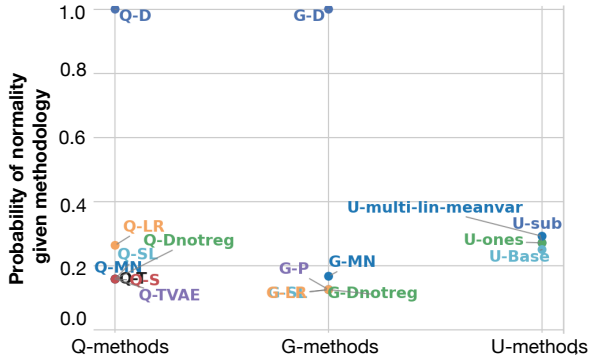

Figure 11: Probability of $p > 0.01$ for the Shapiro-Wilk test of normality for each Q (outcome), G (propensity score), and U (update step) method with the IHDP dataset. Because we undertook all combinations of G and U, each point represents a marginalization over the other dimension(s). For instance, for the Q methods, Q-D (DragonNet) is an average probability result when combining Q-D with all possible other G and U methods. Best viewed in color.

MN dropped as sample size increased. Surprisingly, these results reversed for the IHDP dataset, with Q-D providing the most frequently normally distributed estimates, with the other methods yielding generally poor performance. Both G-LR and G-SL worked well as propensity score models for the LF-datasets, yielding a high likelihood of normally distributed estimates. However, on IHDP only the propensity score estimates from G-D were found to work well. U-ones and U-sub were found to yield consistently normally distributed errors across the LF datasets, with our U-multi unfortunately yielding little advantage over the base model.

In some ways, the relatively disappointing results with respect to the normality of the estimates is not surprising. Benkeser et al. (2017) and van der Laan (2014) showed that the double-robustness property relating to a normal limiting distribution which is afforded by estimators satisfying the efficient influence function does not apply when data-adaptive estimators are used (such as superlearners). In order for the double-robustness property to hold (with respect to the normal limiting distribution) with data-adaptive estimators, additional conditions must be satisfied. The failure to yield normally distributed estimates for many of the evaluated methods in this work thus may well be due to some degree of misspecification in the treatment or outcome models (or, indeed, both). One would expect that using the additional update steps proposed by Benkeser et al. (2017) and van der Laan (2014) would yield improved results and this presents a promising direction for future evaluations and development.

6.4 Results of the Shapley Value Analysis

In addition to the presentation of the results given in Figs. 6-11, as well as those given below in Figs. 20-26, we also explored whether a meta-analysis using the Shapley value approach (Shapley, 1953; Lundberg et al., 2020; 2017; Lundberg & Lee, 2017) could provide additional insights into the performance variation across methods and datasets. Indeed, one of the limitations of the way the earlier results are presented is that they involve marginalization over one or more methodological components (*e.g.*, to obtain quantile probabilities for the Q-methods, we have to marginalize over all G- and U-methods). This can make it difficult to identify higher-order interactions or performance variability/sensitivity between different combinations of components and datasets.

The results presented in Figs. 12-14, as well as those in the Appendix in Table 6 and Figs. 17-19, represent the output from the SHapley Additive exPlanations (SHAP) machine learning explainability technique (Lundberg et al., 2020). The process is as follows: we take the full set of factorial results from our experiments (all combinations from Table 1), and use the choice dataset and choice/combination of Q-, G-, and U-methods as predictors in a regression with each of the MSE, ATE estimate standard error, and Shapiro-Wilk test p-value

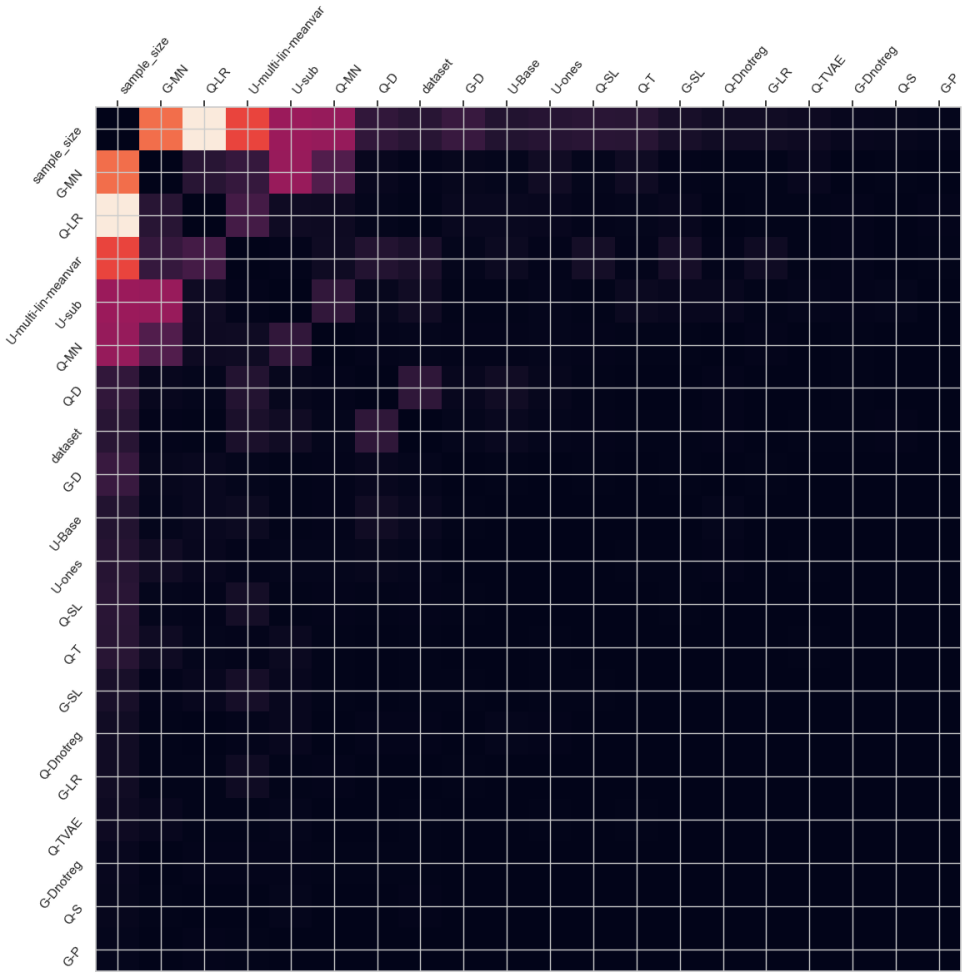

Figure 12: Shapley predictor interaction heatmap for MSE outcome.

results as different outcomes. We use a random forest algorithm (Breiman, 2001) as the regressor, with the default values in the scikit-learn implementation which have been shown to yield stable and consistent performance (Pedregosa et al., 2011; Probst et al., 2018). Then, the SHAP package conceives of the regression task as a game, with each predictor representing an agent which, in collaboration with other predictors, seek to maximise the performance of the regressor. The output of the Shapley analysis includes global predictor importances (which tell us, on average, how useful each predictor is in explaining the regressor output), the individual impact of each datapoint in the dataset on the regressor's output, and a quantification of the degree of interaction between predictors. These results therefore help us identify whether there exist strong interactions between the choice of methods and/or datasets.

Figs. 12, 13, and 14 show Shapley interaction heatmaps for the three outcomes MSE, ATE estimate standard error, and p-values, respectively. Brighter values indicate the presence of an interaction between predictors (where the combined information tells us something more about the likely value of the outcome than the individual predictors alone). A good method should arguably be one which does not exhibit strong interaction with the dataset and/or the sample-size. Any such interactions indicate that the method is otherwise sensitive to these factors, and therefore does not provide consistent performance.

For MSE in Fig. 12, we see interactions between the sample size, G-MN, Q-LR, U-Multi and, to a lesser extent, U-sub and Q-MN. Otherwise, practically no other methods show strong interactions with each other or the datasets apart from the combination of G-MN with U-Sub. For the ATE estimate standard error

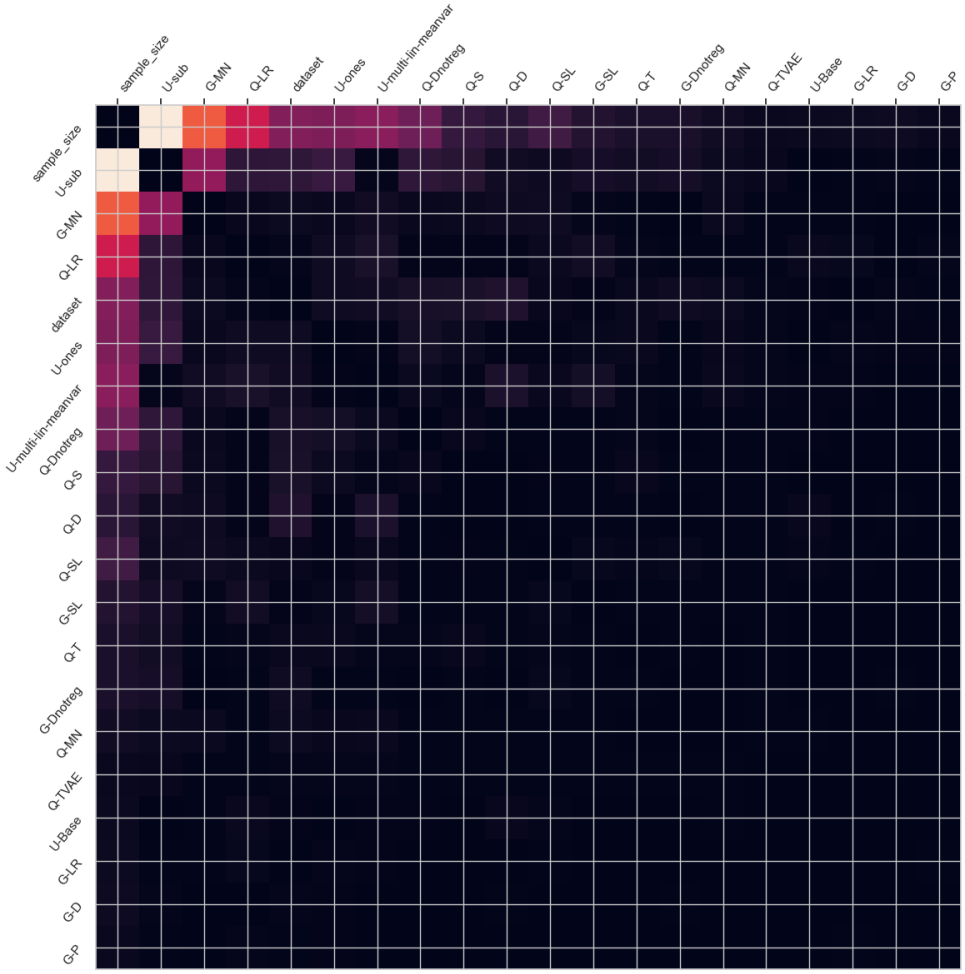

Figure 13: Shapley predictor interaction heatmap for ATE estimate standard error outcome.

results in Fig. 13, we see very similar results, indicating a strong interaction between the sample size and U-sub, G-MN and Q-LR, in particular. Otherwise, there are few notable interactions indicated by these results.

For the Shapiro-Wilk test for normality p-value results in Fig. 14, we predominantly the dataset and sample size playing a role, indicating that the structural and/or functional form of the dataset impacts the ability of a method to yield normally distributed results, particularly when used in combination with Q-D. Remember here that the p-value is an empirical test for the normality of the distribution of the estimates that we compute over the number of simulations, which is fixed across datasets/experiments.

Finally, the overall variable importance summaries in Figs. 17-19, provided in Appendix I, confirm the sensitivity of performance to the sample size and (for the p-value) the dataset.

7 Discussion

In this paper we have introduced some key aspects of semiparametric theory and have undertaken a comprehensive evaluation of the potential of semiparametric techniques to provide a 'free' performance improvement for existing estimators without needing more data, and without needing to retrain them. We also proposed a new pseudo-ensemble NN method 'MultiNet' for simulating an ensemble approach with a single network,

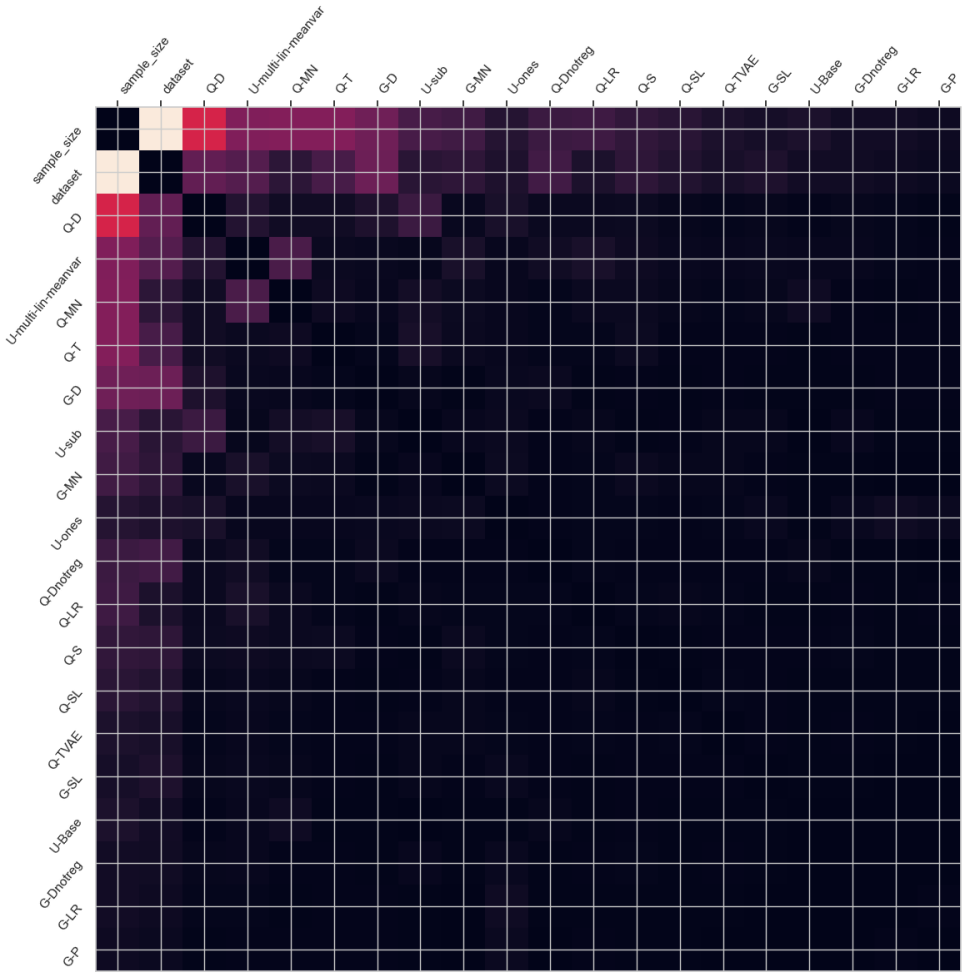

Figure 14: Shapley predictor interaction heatmap for Shapiro-Wilk p-value outcome.

a new update step variant 'MultiStep'. Our evaluation included a discussion of the choice of outcome 'Q' method, propensity score 'G' method, and the update 'U' method.

The summary of results indicates that the best results were subject to variation across datasets and sample size. This was particularly evident when comparing the results on the LF datasets with those of the IHDP dataset, and when reviewing the Shapley interaction results. Indeed, the Shapley results are quite telling, in that the most predictive variables for MSE, s.e. and normality were sample size and dataset structure/functional form. This highlights a dependence of the performance on the method-dataset combination which is difficult to alleviate. That said, some methods were more stable across datasets than others. A further review of the Shapley interaction plots can be useful to researchers in selecting a group of Q-, G-, and U-methods which are, at least based on these results, relatively insensitive to the dataset structure/functional form and sample size. Juggling the relativel performance metrics is difficult, however, because a method might perform well in terms of MSE, and be insensitive to sample size, for example, and yet fail to yield normally distributed errors. Such an estimator would be difficult to justify if one is interested in performing reliable statistical inference (such as that related to null-hypothesis significance testing).

More generally, dataset dependence was previously highlighted in the context of causal effect estimation by Curth et al. (2021b), and it is something which practitioners should be aware of, especially in the causal inference setting where we do not have access to ground-truth. Researchers developing such methods should also, of course, be aware of this issue, because it can significantly inform the evaluation design for testing and

comparing different methods. These caveats notwithstanding, we found our MultiNet method to perform well as an outcome method, yielding state of the art on a number of evaluations, and performing particularly well on datasets with smaller sample sizes. Indeed, to an extent these results conflict with those of Farrell et al. (2021), who showed that relatively basic neural networks were capable of excellent performance when combined with semiparametric techniques and evaluated on their own simulated data as well as data for a direct mail marketing campaign. Arguably, their conclusions and results are less mixed than ours, although it is worth remembering that we restricted the set of methods used for the main evaluation to those with already competitive performance, and the remaining spread of our 'mixed' results may already be reassuringly tight. Unfortunately, it is difficult to say what is an acceptable level of performance, although recent large-scale work by Gordon et al. (2022) suggests that the primary challenge will be in satisfying identifiability - that is, ensuring that our estimand can be expressed as a function of the observational data, that our model is structurally well-specified, and that no unobserved confounders exist which otherwise bias our estimates.

Many of the methods failed to yield normally distributed estimates. This is somewhat expected given that the double robustness guarantees do not apply to the nature of the limiting distribution. Benkeser et al. (2017) and van der Laan (2014) provide a means to augment the update step frameworks to include additional conditions which, when satisfied, extend the double robustness guarantees to the (normal) limiting distribution of the estimates. Furthermore, there are a number of conditions which must hold for the IF update techniques to work. Firstly, our estimator must be regular and asymptotically linear such that the second order remainder term $o_p(\cdot)$ tends in probability to zero sufficiently quickly. These properties concern the sample size, the smoothness of the estimator, and the quality of the models we are using to approximate the relevant factors of the distribution. Clearly, if our initial model(s) is(are) poor/misspecified then a linear path (or equivalently, a first order VME) will not be sufficient to model the residual distance from the estimand, and the update steps may actually worsen our initial estimate. As long as our initial estimator is 'good enough' (insofar as it is regular and asymptotically linear), we can describe any residual bias using IFs. Unfortunately, we are not currently aware of a way to assess 'good enough'-ness, particularly in the causal-inference setting, where explicit supervision is not available. There may exist a way to use the magnitude of the IF to assess the validity of the assumption of asymptotic normality, and use this as a proxy for model performance, but we leave this to future work.

Many open questions remain: a similar set of experiments should be undertaken for other estimands (such as the conditional ATE). Also, one may derive higher order IFs (Carone et al., 2014; van der Laan et al., 2021; van der Vaart, 2014; Robins et al., 2008) which introduce new challenges and opportunities. Additionally, it may be possible to use IFs to derive a proxy representing 'good enough'-ness, *i.e.*, whether the initial estimator is close enough to the target estimand for the remaining bias to be modelled linearly. This, in turn, may also provide a way to assess the performance of causal inference methods, which would be highly advantageous given that explicit supervision will rarely be available in real-world causal inference settings. The extensions of Benkeser et al. (2017) and van der Laan (2014) also represent an interesting avenue for further development, particularly in relation to the goal of undertaking valid statistical inference with nonparametric estimators.

7.1 Choosing Estimators

In practice, researchers are faced with a dilemma. Whilst the theory underpinning semiparametric and double-robust methods has not been called into question, we find that the real-world performance of the chosen estimators can vary across datasets, algorithmic choices (*e.g.*, which candidate learners to included in an machine learning ensemble estimator), and sample sizes. This means that applied researchers may be unable to simply check the recent literature to find and use the most recent, state-of-the-art methods, as determined by a finite set of experiments on benchmark datasets. Inevitably, there is no free lunch, and researchers are therefore forced to undertake their own evaluation process if they wish to maximise the chances that their methods will yield meaningful and robust results. The question then remains: What should be involved in this evaluation process to best expedite the selection of a method?

We discuss two scenarios: Firstly, the case when practitioners can feasibly simulate datasets with similar characteristics to the phenomenon under study. In this situation, practitioners can follow a similar process to the one documented in this work, and evaluate a set of candidate methods and select one or a handful

of methods which exhibit adequate or exemplary performance. They can then apply these methods to their problem, and, following a process of 'triangulation', identify convergence in the results suggested by these chosen methods. The benefit of this approach is that (a) one can hedge over multiple (of the already better performing) methods rather than risking it all with one potentially poor performer, and (b) one can get a feeling for the consistency and sensitivity of the results to the choice of method. Results which are, for example, widely inconsistent across methods indicate that only a subset or none of the methods are working well for the specific application, and more simulation and evaluation are needed to establish a set of suitable methods.

Secondly, we consider the case when practitioners have little-to-no understanding of the underlying phenomenon (in particular, no understanding of the non-linearity/complexity/functional form relating variables), and are therefore unable to perform meaningful simulations of the underlying phenomenon of interest. In this case, it is especially important that practitioners utilize multiple methods and, again, 'triangulate' in order to understand the sensitivity of their results to the chosen methodology. Until one sees reasonable consistency across methodologies, it is necessary for researchers to consider again their choice of methods. We advise that in this second case, it is also important that researchers evaluate a diverse range of methods, with varying degrees of complexity. A review of the results presented in this paper does confirm the well-established notion that smaller datasets can benefit from the use of less complex methods (thus helping to prevent over-fitting). The use of cross-validation techniques can also be useful in mitigating challenges associated with fitting highly-data-adaptive learners to small datasets.

Finally, some of the methods explored in this paper did not compete well, and these can *potentially* be ruled out in order to reduce the choice. We say this tentatively because these methods may perform well with the right set of hyperparameters, and one cannot rule out the possibility that a longer exploration of the hyperparameter space could change the order of results. The hyperparameter selection problem is yet another challenge which practitioners face. Altogether, the number of combinations of these hyperparameters, the choice of Q-, G-, and U-methods, and the choice of candidate learners included in the individual estimator ensembles reinforces the conclusion that practitioners may have to bite the bullet and undertake thorough evaluations of their methodologies, before deploying them on their real-world applications, wherever possible.

8 Broader Impact

It is always important to remember that the reliability of causal inference depends on strong, untestable assumptions (not least because there is rarely any access to ground-truth in the domain of causal inference). Given the variability of the performance of the evaluated methods across datasets, in particular with regards to the normality of the estimates (and therefore also the validity of subsequent inference) any practical application of causal inference methods must be undertaken with caution. Indeed, we recommend researchers establish the extent to which their inference depends on the methods used, by undertaking the same analysis with multiple approaches/estimators.

References

A.M. Alaa and M. van der Schaar. Validating causal inference models via influence functions. *ICLR*, 2019.

A.M. Alaa and M. van der Schaar. Discriminative jackknife: Quantifying uncertainty in deep learning via higher-order influence functions. *arXiv preprint*, arXiv:2007.13481v1, 2020.

N. S. Altman. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3):175–185, 1992. doi: 10.1080/00031305.1992.10475879.

Philipp Bach, Victor Chernozhukov, Malte S. Kurz, and Martin Spindler. DoubleML – An object-oriented implementation of double machine learning in Python. *Journal of Machine Learning Research*, 23(53): 1–6, 2022. URL `http://jmlr.org/papers/v23/21-0862.html`.

D. Benkeser, M. Carone, M.J. van der Laan, and et al. Doubly robust nonparametric inference on the average treatment effect. *Biometrika*, 104(4):863–880, 2017. doi: 10.1093/biomet/asx053.

R. Bhattacharya, R. Nabi, and I. Shpitser. Semiparametric inference for causal effects in graphical models with hidden variables. *arXiv:2003.12659v1*, 2020.

I. Bica, A.M. Alaa, C. Lambert, and M. van der Schaar. From real-world patient data to individualized treatment effects using machine learning: Current and future methods to address underlying challenges. *Clinical Pharmacology and Therapeutics*, 109(1):87–100, 2020. doi: 10.1002/cpt.1907.

P.J. Bickel, C.A.J. Klassen, Y. Ritov, and J.A. Wellner. *Efficient and Adaptive Estimation for Semiparametric Models*. Spinger-Verlag, New York, 1998.

M.J. Blanca, R. Alarcon, and R. Bono. Current practices in data analysis procedures in psychology: what has changed? *Frontiers in Psychology*, 2018. doi: 10.3389/fpsyg.2018.02558.

V. Borisov, T. Leeman, K. Sebler, and J. Haug. Deep neural networks and tabular data: A survey. *arXiv preprint*, arXiv:2110.01889v2, 2022.

L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001. doi: 10.1023/A:1010933404324.

M. Carone, I. Diaz, and M.J. van der Laan. Higher-order targeted minimum loss-based estimation. *U.C. Berkeley Division of Biostatistics Working Paper Series*, 2014.

H. Chen, T. Harinen, Lee J-L., M. Yung, and Z. Zhao. CausalML: Python package for causal machine learning. *arXiv preprint*, 2002.11631, 2020.

V. Chernozhukov, D. Chetverikov, M. Demirer, E. Duflo, C. Hansen, W. Newey, and J. Robins. Double/debiased machine learning for treatment and structural parameters. *Econometrics Journal*, 21(1): C1–C68, 2018. doi: 10.1111/ectj.12097.

A. Curth and M. van der Schaar. Nonparametric estimation of heterogeneous treatment effects: From theory to learning algorithms. *AISTATS*, 130, 2021.

A. Curth, A.M. Alaa, and M. van der Schaar. Estimating structural target functions using machine learning and influence functions. *arXiv preprint*, arXiv:2008.06461v3, 2021a.

A. Curth, D. Svensson, J. Weatherall, and M. van der Schaar. Really doing great at estimating CATE? a critical look at ML benchmarking practices in treatment effect estimation. *35th Conference onf Neural Information Processing Systems (NeurIPS 2021)*, 2021b.

M. Dehghani, Y. Tay, A.A. Gritsenko, Z. Zhao, N. Houlsby, F. Diaz, D. Metzler, and O. Vinyals. The benchmark lottery. *2021*, arXiv:2107.07002v1, arXiv preprint.

V. Dorie. Non-parametrics for causal inference. *https://github.com/vdorie/npci*, 2016.

B. Efron and G. Gong. A leisurely look at the boostrap, the jackkife, and cross-validation. *The American Statistician*, 37(1):36–48, 1983. doi: doi:10.2307/2685844.

R.J. Evans and T.S. Richardson. Smooth, identifiable supermodels of discrete DAG models with latent variables. *Bernoulli*, 25(2):848–876, 2019. doi: 10.3150/17-BEJ1005.

M. Ezzati, A.D. Lopez, and C.J.L. Murray (eds.). *Comparative Quantification of Health Risks: Global and Regional Burden of Disease Attributable to Selected Major Risk Factors*, chapter Effects of multiple interventions. World Health Organization, Geneva, 2004.

M.H. Farrell, T. Liang, and S. Misra. Deep neural networks for estimation and inference. *Econometrica*, 89 (1):181–213, 2021.

A. Fisher and E.H. Kennedy. Visually communicating and teaching intuition for influence functions. *arXiv:1810.03260v3*, 2019.

Y. Freund and R. Schapire. A decision-theoretic generalization of on-line learning and application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997. doi: 10.1006/jcss.1997.1504.

J. Friedman. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5), 2001.

Y. Gal and Z. Ghahramani. Dropout as a Bayesian approximation: representing model uncertainty in deep learning. *ICML*, 2016.

B.R. Gordon, R. Moakler, and F. Zettelmeyer. Close enough? a large-scale exploration of non-experimental approaches to advertising measurement. *arXiv preprint*, arXiv:2201.07055v1, 2022.

C. Guo, G. Pleiss, Y. Sun, and K.Q. Weinberger. On calibration of modern neural networks. *ICLR*, 2017.

R. Guo, J. Li, and H. Liu. Learning individual causal effects from networked observational data. *Association for Computing Machinery*, 2020.

F. R. Hampel. The influence curve and its role in robust estimation. *Journal of the American Statistical Association*, 69(346):383–393, 1974.

X. Han, B.C. Wallace, and Y. Tsvetkov. Explaining black box predictions and unveiling data artifacts through influence functions. *arXiv preprint*, arXiv:2005.06675v1, 2020.

L. Henckel, E. Perković, and M.H. Maathuis. Graphical criteria for efficient total effect estimation via adjustment in causal linear models. *arXiv preprint*, arXiv:1907.02435v2, 2020.

J. L. Hill. Bayesian nonparametric modeling for causal inference. *Journal of Computational and Graphical Statistics*, 20(1), 2011.

O. Hines, O. Dukes, K. Diaz-Oraz, and S. Vansteelandt. Demystifying statistical learning based on efficient influence functions. *arXiv preprint*, arXiv:2107.00681, 2021.

K. Hornik. Some new results on neural network approximation. *Neural Networks*, 6:1069–1072, 1993.

K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2:359–366, 1989. doi: 10.1016/0893-6080(89)90020-8.

Y. Huang and M. Valtorta. Pearl's calculus of intervention is complete. *Proceedings of the Twenty-Second Conference on Uncertainty in Artificial Intelligence*, arXiv:1206.6831:217–224, 2006. doi: 10.5555/3020419.3020446.

H. Ichimura and W. Newey. The influence function of semiparametric estimators. *arXiv preprint*, arXiv:1508.01378v2, 2021.

G.W. Imbens and D.B. Rubin. *Causal inference for statistics, social, and biomedical sciences. An Introduction.* Cambridge University Press, New York, 2015.

E. Jones, T. Oliphant, P. Petereson, and et al. SciPy: Open source scientific tools for Python. *http://www.scipy.org*, 2001.

Y. Jung, J. Tian, and E. Bareinboim. Estimating causal effects using weighting-based estimators. *The 34th AAAI Conference on Artificial Intelligence*, 2020.

A. Kadra, M. Lindauer, F. Hutter, and J. Grabocka. Regularization is all you need: simple neural nets can excel on tabular data. *NeurIPS*, 2021.

E.H. Kennedy. Semiparametric theory and empirical processes in causal inference. *arXiv:1510.04740v3*, 2016.

E.H. Kennedy. Optimal doubly robust estimation of hetereogeneous causal effects. *arXiv preprint*, arXiv:2004.14497v2, 2020.

D. P. Kingma and J. L. Ba. Adam: a method for stochastic optimization. *arXiv:1412.6980v9*, 2017.

P.W. Koh and P. Liang. Understanding black-box predictions via influence curves. *PMLR*, 2017.

N. Kreif and K. DiazOrdaz. Machine learning in policy evaluation: new tools for causal inference. *arXiv:1903.00402v1*, 2019.

S. R. Kunzel, J.S. Sekhon, P.J. Bickel, and B. Yu. Meta-learners for estimating heterogeneous treatment effects using machine learning. *arXiv preprint*, arXiv:1706.03461v6, 2019.

C.F. Kurz. Augmented inverse probability weighting and the double robustness property. *Medical Decision Making*, 2021. doi: 10.1177/0272989X211027181.

J. Levy. Tutorial: Deriving the efficient influence curve for large models. *arXiv:1903.01706v3*, 2019.

H. Li, S. Rosete, J. Coyle, R.V. Phillips, N.S. Hejazi, I. Malenica, B.F. Arnold, J. Benjamin-Chung, A. Mertens, J.M. Colford, M.J. van der Laan, and A.E. Hubbard. Evaluating the robustness of targeted maximum likelihood estimators via realistic simulations in nutrition intervention trials. *Statistics in Medicine*, 41(2), 2022. doi: https://doi.org/10.1002/sim.9348.

C. Louizos, U. Shalit, J. Mooij, D. Sontag, R. Zemel, and M. Welling. Causal effect inference with deep latent-variable models. *31st Conference on Neural Information Processing Systems*, 2017.

S.M. Lundberg and S-I. Lee. A unified approach to interpreting model predictions. *31st Conference on Neural Information Processing Systems*, 2017.

S.M. Lundberg, G.G. Erion, and S-I. Lee. Consistent individualized feature attribution for tree ensembles. *Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia*, 2017.

S.M. Lundberg, G. Erion, H. Chen, A. DeGrave, J.M. Prutkin, B. Nair, R. Katz, J. Himmelfarb, N. Bansal, and S-I. Lee. From local explanations to global understanding with explainable AI for trees. *Nature Machine Intelligence*, 2:56–67, 2020. doi: 10.1038/s42256-019-0138-9.

M.A. Luque-Fernandez, M. Schomaker, B. Rachet, and M.E. Schnitzer. Targeted maximum likelihood estimation for a binary treatment: A tutorial. *Statistics in Medicine*, 37(16):2530–2546, 2018. doi: 10.1002/sim.7628.

R. Neugebauer and M.J. van der Laan. Why prefer double robust estimates? illustration with causal point treatment studies. *Journal of Statistical Planning and Inference*, 129(1):405–426, 2005.

W. Newey. Semi-parametric efficicency bounds. *Journal of Applied Econometrics*, 5:99–135, 1990.

W. Newey. The asymptotic variance of semi-parametric estimators. *Econometrika*, 62:1349–82, 1994.

J. Pearl. *Causality*. Cambridge University Press, Cambridge, 2009.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, and B. et al. Thirion. Scikit-learn: Machine learning in Python. *JMLR*, 12:2825–2830, 2011.

M. Petersen, L. Balzer, D. Kwarsiima, N. Sang, G. Chamie, J. Ayieko, J. Kabami, A. Owaraganise, T. Liegler, F. Mwangwa, and K. Kadede. Association of implementation of a universal testing and treatment intervention with HIV diagnosis, receipt of antiretroviral therapy, and viral suppression in East Africa. *Journal of American Medical Association*, 317(21):2196–2206, 2017. doi: 10.1001/jama.2017.5705.

K.E. Porter, S. Gruber, M.J. van der Laan, and J.S. Sekhon. The relative performance of targeted maximum likelihood estimators. *International Journal of Biostatistics*, 7:1034, 2011.

P. Probst, M.N. Wright, and A.-L. Boulesteix. Hyperparameters and tuning strategies for random forest. *Wires Data Mining and Knowledge Discovery*, 2018. doi: 10.1002/widm.1301.

T.S. Richardson and P. Spirtes. Causal inference via ancestral graph models. In P. Green, N. Hjort, and S. Richardson (eds.), *Highly Structured Stochastic Systems*. Oxford University Press, Oxford, 2003.

T.S. Richardson, R.J. Evans, J.M. Robins, and I. Shpitser. Nested Markov properties for Acyclic Directed Mixed Graphs. *arXiv preprint*, arXiv:1701.06686v2, 2017.

J. Robins. A new approach to causal inference in mortality studies with a sustained exposure period - application to control of the healthy worker survivor effect. *Mathematical Modelling*, 7:1393–1512, 1986. doi: 10.1016/0270-0255(86)90088-6.

J.M. Robins, L. Li, E.J. Tchetgen, and A.W. van der Vaart. Higher order influence functions and minimax estimation of nonlinear functionals. *Probability and Statistics: Essays in Honor of David A. Freedman*, pp. 335–421, 2008.

A. Rotnitzky and E. Smucler. Efficient adjustment sets for population average treatment effect estimation in non-parametric causal graphical models. *JMLR*, 21(188), 2020.

D. B. Rubin. Causal inference using potential outcomes: Design, modeling, decisions. *Journal of the American Statistical Association*, 100(469):322–331, 2005. doi: 10.1198/016214504000001880.

N. Sani, J. Lee, R. Nabi, and I. Shpitser. A semiparametric approach to interpretable machine learning. *arXiv preprint*, arXiv:2006.04732 Search... arXiv:2006.04732 Search... arXiv:2006.04732, 2020.

U. Shalit, F. D. Johansson, and D. Sontag. Estimating individual treatment effect: generalization bounds and algorithms. *ICML*, 2017.

S.S. Shapiro and M.B. Wilk. An analysis of variance test for normality (complete samples). *Biometrika*, 52 (3-4):591–611, 1965. doi: 10.1093/biomet/52.3-4.591.

L.S. Shapley. A value for n-person games. *Contributions to the Theory of Games*, 2(28):307–317, 1953.

C. Shi, D. M. Blei, and V. Veitch. Adapting neural networks for the estimation of treatment effects. *33rd Conference on Neural Information Processing Systems*, 2019.

C. Shi, T. Xu, and W. Bergsma. Double generative adversarial networks for conditional independence testing. *arXiv:2006.02615v1*, 2020.

I. Shpitser and J. Pearl. Identification of joint interventional distributions in recursive semi-Markovian causal models. *Proceedings of the National Conference on Artificial Intelligence*, 21:1219–1226, 2006.

R. Shwartz-Ziv and A. Armon. Tabular data: Deep learning is not all you need. *Information Fusion*, 81: 84–90, 2021. doi: 10.1016/j.inffus.2021.11.011.

B. Siegerink, W. den Hollander, M. Zeegers, and R. Middelburg. Causal inference in law: an epidemiological perspective. *European Journal of Risk Regulation*, 7(1):175–186, 2016. doi: 10.1017/S1867299X0000547X.

C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *CVPR*, 2015.

J. Tian and J. Pearl. A general identification condition for causal effects. *AAAI*, 2002.

A. Tsiatis. *Semiparametric Theory and Missing Data*. Spinger, New York, 2006.

M. J. van der Laan and S. Gruber. Targeted minimum loss based estimation of causal effects of multiple time point interventions. *Int. J. Biostat*, 8: Art 9(41), 2012.

M. J. van der Laan and S. Rose. *Targeted Learning - Causal Inference for Observational and Experimental Data*. Springer International, New York, 2011.

M. J. van der Laan and R. J. C. M. Starmans. Entering the era of data science: targeted learning and the integration of statistics and computational data analysis. *Advances in Statistics*, 2014.

M. J. van der Laan, Z. Wang, and L. van der Laan. Higher order targeted maximum likelihood estimation. *arXiv:2101.06290v3*, 2021.

M.J. van der Laan. Targeted estimation of nuisance parameters to obtain valid statistical inference. *International Journal on Biostatistics*, 10:29–57, 2014.

M.J. van der Laan and D.B. Rubin. Targeted maximum likelihood learning. *The International Journal of Biostatistics*, 2(1), 2006. doi: 10.2202/1557-4679.1043.

M.J. van der Laan, E.C. Polley, and A.E. Hubbard. Super Learner. *Statistical Applications of Genetics and Molecular Biology*, 6(25), 2007. doi: 10.2202/1544-6115.1309.

A.W. van der Vaart. Higher order tangent spaces and influence functions. *Statistical Science*, 29(4):679–686, 2014.

T. Verma and J. Pearl. Equivalence and synthesis of causal models. *Proc. 6th Conf. on Uncertainty in Artificial Intelligence*, 1990.

M. J. Vowels. Misspecification and unreliable interpretations in psychology and social science. *Psychological Methods*, 2021. doi: 10.1037/met0000429.

M. J. Vowels, N.C. Camgoz, and R. Bowden. Targeted VAE: Structured inference and targeted learning for causal parameter estimation. *IEEE SMDS*, 2021.

Y. Wen, D. Tran, and J. Ba. BatchEnsemble: An alternative approach to efficient ensemble and lifelong learning. *ICLR*, 2020.

D.H. Wolpert and W.G. Macready. No free lunch theorems for optimization. *IEEE Transacions on Evolutionary Computation*, 1(67), 1997. doi: 10.1109/4235.585893.

P.A. Wu and K. Fukumizu. Causal mosaic: cause-effect inference via nonlinear ICA and ensemble method. *AISTATS*, 108, 2020.

P.A. Wu and K. Fukumizu. Intact-VAE: Estimating treatment effects under unobserved confounding. *ICLR*, 2022.

L. Yao, S. Li, Y. Li, M. Huai, J. Gao, and A. Zhang. Representation learning for treatment effect estimation from observational data. *32nd Conference on Neural Information Processing Systems (NeurIPS)*, 2018.

L. Yao, Z. Chu, S. Li, Y. Li, J. Gao, and A. Zhang. A survey on causal inference. *ACM Transactions on Knowledge Discovery from Data*, 15(5):1–46, 2020. doi: 10.1145/3444944.

J. Yoon, J. Jordan, and M. van der Schaar. GANITE: Estimation of individualized treatment effects using generative adversarial nets. *ICLR*, 2018.

Q. Zhong and J-L Wang. Neural networks for partially linear quantile regression. *arXiv preprint*, arXiv:2106.06225, 2021.

H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *J. R. Statist. Soc.*, 67(2): 301–320, 2005.

A Causal Assumptions

The causal quantity can be estimated in terms of observational (and therefore statistical) quantities if a number of strong (but common: Yao et al., 2020; Guo et al., 2020; Rubin, 2005; Imbens & Rubin, 2015; Vowels et al., 2021) assumptions hold: (1) Stable Unit Treatment Value Assumption (SUTVA): the potential outcomes for each individual or data unit are independent of the treatments assigned to all other individuals. (2) Positivity: the assignment of treatment probabilities are non-zero and non-deterministic $P(T = t_i|\mathbf{X} = \mathbf{x}_i) > 0$, $\forall\, t, \mathbf{x}$. (3) Ignorability/Unconfoundedness/Conditional Exchangeability: There are no unobserved confounders, such that the likelihoods of treatment for two individuals with the same covariates are equal, and the potential outcomes for two individuals with the same latent covariates are also equal s.t. $T \perp\!\!\!\perp (Y(1), Y(0))|\mathbf{X}$.

B Statistical Inference with Influence Functions

Machine learning techniques, while parameterized *per se*, are often non-parametric insofar as the estimates they yield are not directly parameterizable as (*e.g.*) a Gaussian with a mean and a variance. However, such parameterization is extremely helpful in facilitating statistical inference, including the ubiquitous null hypothesis significance test, which is straightforward when the parameter being tested is normally distributed. Whilst other approaches to statistical inference with non-parametric methods exist, such as the bootstrap approach (Efron & Gong, 1983), influence functions have been recognized as a valid approach for some time (Bickel et al., 1998; Tsiatis, 2006).

Following van der Laan & Rose (2011, p.75) we can derive 95% confidence intervals from the influence function to be (assuming normal distribution):

$$
\begin{aligned}
&\widehat{\mathrm{Var}}(\phi) = \frac{1}{n}\sum_{i}^{n}\left[\phi(\mathbf{z}_i) - \frac{1}{n}\sum_{j}^{n}\phi(\mathbf{z}_j)\right]^2, \\
&\widehat{\mathrm{se}} = \sqrt{\frac{\widehat{\mathrm{Var}}(\phi)}{n}}, \\
&\Psi^*(\hat{\mathcal{P}}_n) \pm 1.96\widehat{\mathrm{se}}, \\
&p_{val} = 2\left[1 - \Phi\left(\left|\frac{\Psi^*(\hat{\mathcal{P}}_n)}{\widehat{\mathrm{se}}}\right|\right)\right],
\end{aligned}
\tag{8}
$$

where $\Psi^*(\hat{\mathcal{P}}_n)$ is the estimated target quantity after bias correction has been applied, Φ is the CDF of a normal distribution, $\widehat{\mathrm{se}}$ is the standard error, and p_{val} is the p-value.

C IFs for General Graphical Models

In this paper, we focus on the estimation of average treatment effect in the setting of Fig 1a. However, the methods discussed in this paper can be applied for more complex estimands with an arbitrary causal graph structure, as long as the estimand at hand is causally *identifiable* from the observed data. In this section, we discuss the derivation of IFs for a general form of an estimand in a general graphical model.

C.1 Influence Function of an Interventional Distribution

The causal identification of interventional distributions is well-studied in the literature. In the case of full observability, any interventional distribution is identifiable using (extended) g-formula (Ezzati et al., 2004; Robins, 1986). If some variables of the causal system are unobserved, all interventional distributions are not necessarily identifiable. Tian & Pearl (2002) and Shpitser & Pearl (2006) provided necessary and sufficient

conditions of identifiability in such models. The causal identification problem in DAGs with unobserved (latent) variables can equivalently be defined on *acyclic directed mixed graphs* (ADMGs) (Richardson & Spirtes, 2003; Richardson et al., 2017; Evans & Richardson, 2019). ADMGs are acyclic mixed graphs with directed and bidirected edges, that result from a DAG through a latent projection operation onto a graph over the observable variables (Verma & Pearl, 1990).

Pearl's do-calculus is shown to be complete for the identification of interventional distributions (Huang & Valtorta, 2006). Let $\mathbf{V}$ denote the set of all observed variables. Starting with an identifiable interventional distribution $P(\mathbf{y}|do(\mathbf{T}=\mathbf{t}'))$, an identification functional of the following form is derived using do-calculus:

$$\mathcal{P}(\mathbf{y}|do(\mathbf{T}=\mathbf{t}')) = \sum\!\!\!\!\!\!\int_{\mathbf{S}} \frac{\Pi_i \mathcal{P}(\mathbf{a_i}|\mathbf{b_i})}{\Pi_j \mathcal{P}(\mathbf{c_j}|\mathbf{d_j})}, \tag{9}$$

where $\mathbf{a_i}, \mathbf{b_i}, \mathbf{c_j}$, and $\mathbf{d_j}$ are realizations of $\mathbf{A_i}, \mathbf{B_i}, \mathbf{C_j}$, and $\mathbf{D_j}$, respectively, and $\mathbf{A_i}, \mathbf{B_i}, \mathbf{C_j}, \mathbf{D_j}, \mathbf{S}$ are subsets of variables such that for each i and j, $\mathbf{A_i} \cap \mathbf{B_i} = \varnothing$ and $\mathbf{C_j} \cap \mathbf{D_j} = \varnothing$. Note that the sets $\mathbf{B_i}$ and $\mathbf{D_j}$ might be empty. The $\sum\!\!\!\!\!\!\int$ symbol in Eq. 9 indicates a summation over the values of the set of variables $\mathbf{S}$ in the discrete case, and an integration over these values in the continuous setting. To derive the influence function of Eq. 9, we begin with a conditional distribution of the form $\mathcal{P}(\mathbf{a}|\mathbf{b})$. If $\mathbf{b} \neq \varnothing$, we can write

$$\begin{aligned}
\mathcal{P}_\epsilon(\mathbf{v}) &= (1-\epsilon)\mathcal{P}(\mathbf{v}) + \epsilon\delta_{\tilde{\mathbf{v}}}(\cdot), \\
\mathcal{P}_\epsilon(\mathbf{a}|\mathbf{b}) &= \frac{\mathcal{P}_\epsilon(\mathbf{a},\mathbf{b})}{\mathcal{P}_\epsilon(\mathbf{b})}, \\
\left.\frac{d\mathcal{P}_\epsilon(\mathbf{a}|\mathbf{b})}{d\epsilon}\right|_{\epsilon=0} &= \frac{\delta_{\tilde{\mathbf{a}},\tilde{\mathbf{b}}}(\mathbf{a},\mathbf{b}) - \mathcal{P}(\mathbf{a},\mathbf{b})}{\mathcal{P}(\mathbf{b})} - \frac{\mathcal{P}(\mathbf{a},\mathbf{b})[\delta_{\tilde{\mathbf{b}}}(\mathbf{b}) - \mathcal{P}(\mathbf{b})]}{\mathcal{P}^2(\mathbf{b})} \\
&= \mathcal{P}(\mathbf{a}|\mathbf{b}) \cdot \left(\frac{\delta_{\tilde{\mathbf{a}},\tilde{\mathbf{b}}}(\mathbf{a},\mathbf{b})}{\mathcal{P}(\mathbf{a},\mathbf{b})} - \frac{\delta_{\tilde{\mathbf{b}}}(\mathbf{b})}{\mathcal{P}(\mathbf{b})} \right),
\end{aligned} \tag{10}$$

where $\tilde{\mathbf{v}}$ is the point that we compute the influence function at, and $\tilde{\mathbf{a}}, \tilde{\mathbf{b}}$ are the values of sets of variables $\mathbf{A}, \mathbf{B} \subseteq \mathbf{V}$ that are consistent with $\tilde{\mathbf{v}}$. For an empty $\mathbf{b}$, using similar arguments, we have:

$$\left.\frac{d\mathcal{P}_\epsilon(\mathbf{a})}{d\epsilon}\right|_{\epsilon=0} = \mathcal{P}(\mathbf{a}) \cdot \left(\frac{\delta_{\tilde{\mathbf{a}}}(\mathbf{a})}{\mathcal{P}(\mathbf{a})} - 1 \right). \tag{11}$$

With slight abuse of notation, for $\mathbf{b} = \varnothing$, we define $\frac{\delta_{\tilde{\mathbf{b}}}(\mathbf{b})}{\mathcal{P}(\mathbf{b})} = 1$. Using Eq. 10 and Eq. 11, we can now derive the IF of Eq. 9.

$$\begin{aligned}
&\phi(\tilde{\mathbf{v}}, \mathcal{P}) = \left.\frac{d((1-\epsilon)\mathcal{P} + \epsilon\delta_{\tilde{\mathbf{v}}})}{d\epsilon}\right|_{\epsilon=0} = \\
&\sum\!\!\!\!\!\!\int_{\mathbf{S}} \frac{\Pi_i \mathcal{P}(\mathbf{a_i}|\mathbf{b_i})}{\Pi_j \mathcal{P}(\mathbf{c_j}|\mathbf{d_j})} \cdot \left[\sum_i \left(\frac{\delta_{\tilde{\mathbf{a}_i},\tilde{\mathbf{b}_i}}(\mathbf{a_i},\mathbf{b_i})}{\mathcal{P}(\mathbf{a_i},\mathbf{b_i})} - \frac{\delta_{\tilde{\mathbf{b}_i}}(\mathbf{b_i})}{\mathcal{P}(\mathbf{b_i})} \right) - \sum_j \left(\frac{\delta_{\tilde{\mathbf{c}_j},\tilde{\mathbf{d}_j}}(\mathbf{c_j},\mathbf{d_j})}{\mathcal{P}(\mathbf{c_j},\mathbf{d_j})} - \frac{\delta_{\tilde{\mathbf{d}_j}}(\mathbf{d_j})}{\mathcal{P}(\mathbf{d_j})} \right) \right].
\end{aligned} \tag{12}$$

Note that we used $\frac{d}{d\epsilon}\frac{1}{\mathcal{P}_\epsilon(\mathbf{c}|\mathbf{d})} = -\frac{\frac{d}{d\epsilon}\mathcal{P}_\epsilon(\mathbf{c}|\mathbf{d})}{\mathcal{P}^2_\epsilon(\mathbf{c}|\mathbf{d})}$. Equation 12 is the foundation to the approach that shall be discussed in the following section for deriving the IF of a general class of estimands.

C.2 Influence Function of a General Estimand

We have so far discussed the influence function of a causal effect of the form $\mathcal{P}(\mathbf{y}|do(\mathbf{T}=\mathbf{t}'))$. In this section, we show how IFs can be derived for any general estimand of the form:

$$\Psi(\mathcal{P}) = \mathbb{E}_{\mathcal{P}}[\kappa(\mathcal{P})], \tag{13}$$

where $\kappa(\cdot)$ is a functional. Then we have:

$$\begin{aligned}
\mathcal{P}_\epsilon &= \epsilon\hat{\mathcal{P}}_n + (1-\epsilon)\mathcal{P}, \\
\Psi(\mathcal{P}_\epsilon) &= \int \kappa(\mathcal{P}_\epsilon)\mathcal{P}_\epsilon d\mathbf{v}, \\
\left.\frac{d\Psi(\mathcal{P}_\epsilon)}{d\epsilon}\right|_{\epsilon=0} &= \int \left(\frac{d\mathcal{P}_\epsilon}{d\epsilon}\cdot\kappa(\mathcal{P}_\epsilon) + \frac{d\kappa}{d\mathcal{P}_\epsilon}\cdot\frac{d\mathcal{P}_\epsilon}{d\epsilon}\cdot\mathcal{P}_\epsilon\right)\Big|_{\epsilon=0} d\mathbf{v} \\
&= \int \left(\kappa(\mathcal{P}) + \frac{d\kappa}{d\mathcal{P}}\cdot\mathcal{P}\right)\cdot\left.\frac{d\mathcal{P}_\epsilon}{d\epsilon}\right|_{\epsilon=0} d\mathbf{v} \\
&= \int \kappa(\mathcal{P})\cdot\left.\frac{d\mathcal{P}_\epsilon}{d\epsilon}\right|_{\epsilon=0} d\mathbf{v} + \mathbb{E}_{\mathcal{P}}\left[\frac{d\kappa}{d\mathcal{P}}\cdot\left.\frac{d\mathcal{P}_\epsilon}{d\epsilon}\right|_{\epsilon=0}\right].
\end{aligned} \tag{14}$$

The value of $\left.\frac{d\mathcal{P}_\epsilon}{d\epsilon}\right|_{\epsilon=0}$ can be plugged into Eq. 14, which completes the derivation of the IF for the estimand in Eq. 13. As an example, if the queried estimand is the average density of a variable Y, that is, κ is the identity functional, then:

$$\begin{aligned}
\Psi(\mathcal{P}) &= \int \mathcal{P}^2(y)dy, \\
\left.\frac{d\Psi(\mathcal{P}_\epsilon)}{d\epsilon}\right|_{\epsilon=0} &= \int (\mathcal{P} + 1\cdot\mathcal{P})\cdot\left.\frac{d\mathcal{P}_\epsilon}{d\epsilon}\right|_{\epsilon=0} dy \\
&= \int 2\mathcal{P}(y)\cdot\left.\frac{d\mathcal{P}_\epsilon}{d\epsilon}\right|_{\epsilon=0} dy.
\end{aligned}$$

Algorithm 1 summarises the steps of our proposed automated approach to derive the influence function of an estimand of the form presented in Eq. 13, given a general graphical model. Note that if the effect is identifiable, this algorithm outputs the analytic influence function, and otherwise, throws a failure. A demonstrative example can be found in the associated code repository in the form of a notebook, and/or in the attached supplementary code.

Algorithm 1 IF of an identifiable effect.

input: An estimand $\Psi(\mathcal{P})$ of the form of Eq. 13, an interventional distribution $\mathcal{P}$, causal graph $\mathcal{G}$
output: The analytic IF of $\Psi(\mathcal{P})$ if $\mathcal{P}$ is identifiable, fail o.w.

if $\mathcal{P}$ is identifiable **then**
  $\tilde{\mathcal{P}} \leftarrow$ the identification functional of $\mathcal{P}$ (Eq. 9) using do-calculus
  $\phi \leftarrow$ the IF of $\mathcal{P}$ as in Eq. 12
  $\left.\frac{d\Psi(\mathcal{P}_\epsilon)}{d\epsilon}\right|_{\epsilon=0} \leftarrow$ the formulation as in Eq. 14
  $\Phi \leftarrow$ Plug ϕ into $\left.\frac{d\Psi(\mathcal{P}_\epsilon)}{d\epsilon}\right|_{\epsilon=0}$
  return Φ
else
  return FAIL

D Influence Function Update Methods

D.1 One-Step and Submodel Approach

Using the *one-step* approach, the original estimator $\Psi(\hat{\mathcal{P}}_n)$ can be improved by a straightforward application of the Von Mises Expansion (VME) of Eq. 3 - one takes the initial estimator and adds to it the estimate of the IF to yield an updated estimator which accounts for the 'plug-in bias'. In the case of the ATE, this yields the augmented inverse propensity weighted (AIPW) estimator (Hines et al., 2021; Neugebauer & van der Laan, 2005; Kurz, 2021).

The second *submodel* approach updates the initial estimate by solving $\sum_{i=1}^{n} \phi(\mathbf{z}_i, \hat{\mathcal{P}}_n) = 0$, which can be done by estimating the degree to which the principal estimator Q is being biased, and correcting for it. This approach works by first constructing a parametric submodel in terms of the plug in estimator $Q(t, \mathbf{X})$ and a function H of the propensity score G (which represents the biasing quantity), and with these can be used to derive an updated plug-in estimator $Q^*(t, \mathbf{x})$. In the expressions which follow, we assume binary treatment T and have replaced the Dirac delta functions with indicator functions. First, the updated estimator $Q^*(t, \mathbf{x})$ can be expressed in terms of a biasing quantity H as follows:

$$\begin{aligned} \hat{Q}^*(T=1, \mathbf{x}_i) &= \hat{Q}(T=1, \mathbf{x}_i) + \hat{\gamma} H(\mathbf{z}_i, T=1), \\ \text{where } \; H(\mathbf{z}_i, T=1) &= \frac{\mathbb{1}_{t_i}(1)}{\hat{G}(\tilde{\mathbf{x}})}. \\ \text{The equivalent for when treatment is 0 can be expressed as:} & \\ \hat{Q}^*(T=0, \mathbf{x}_i) &= \hat{Q}(T=0, \mathbf{x}_i) + \hat{\gamma} H(\mathbf{z}_i, T=0), \\ \text{where } \; H(\mathbf{z}_i, T=0) &= -\frac{1 - \mathbb{1}_{t_i}(0)}{1 - \hat{G}(\tilde{\mathbf{x}})}. \end{aligned} \tag{15}$$

$H(\mathbf{z}_i, t_i)$ is known as the clever covariate. The parameter $\hat{\gamma}$ is estimated as the coefficient in the associated intercept-free 'maximum-likelihood linear regression' reflected in the first line of Eq. 15 above. When the updated $\hat{Q}^*$ is substituted into Eq.4, one solves in one update what is known as the *efficient influence function*, and following the update, the mean of the IF, as well as the residual bias will be zero. In practice, the two methods yield different results with finite samples (Porter et al., 2011; Benkeser et al., 2017). In particular, the one-step / AIPW estimator may yield estimates outside of the range of values allowed according to the parameter space, and be more sensitive to near-positivity violations (*i.e.*, when the probability of treatment is close to zero) owing to the first term on the RHS of Eq. 4 (Luque-Fernandez et al., 2018). In contrast, the submodel approach will not, because it is constrained by the intercept-free regression step - for instance, if the outcome Y is binary, the logistic link function will prevent any 'out-of-bounds' convergence behavior which might otherwise result in an unstable estimate for the coefficient $\hat{\gamma}$. Thus, both the one-step and submodel approach achieve the same aim: solving for the efficient influence function. However, the one-step approach does so naively according to the VME, and the second does so according to an additional regression stage (but nonetheless still only requires a single update to the original outcome model Q).

D.2 Targeted Regularization

Targeted regularization has, to the best of our knowledge, only been used twice in the NN literature, once in DragonNet (Shi et al., 2019), and once in TVAE (Vowels et al., 2021), both of which were applied to the task of causal inference. The idea is to solve the efficient influence curve *during* NN training, similarly to Eq. 15, on a per-batch basis. The parameter $\hat{\gamma}$ in Eq. 15 is treated as a learnable parameter, trained as part of the optimization of the NN. The submodel update in Eq. 15 is thereby recast as a regularizer which influences the weights and biases of the outcome model $\hat{Q}(t, \mathbf{x})$. In total, then, the training objective is to minimize the sum of the negative log-likelihood (NLL) of the outcome model $\hat{Q}(t, \mathbf{x})$ which has parameters θ (which comprises NN weights and biases), and the NLL of the updated outcome model $\hat{Q}^*(t, \tilde{\mathbf{x}})$, which is parameterized by both θ and $\hat{\gamma}$. As the second NLL term involves the clever covariate H, which in turn involves the plug-in estimator for the propensity score $G(Z)$, we also need a model for the treatment which may be trained via another NLL objective, or integrated into the same NN as the one for the outcome model. Due to the paucity of theoretical analysis for NNs, it is not clear whether targeted regularization provides similar guarantees (debiasing, double-robustness, asymptotic normality) to the one-step and submodel approaches, and this is something we explore empirically.

E LF (v1 and v2) and Gen Dataset Details

The generating equations for LF(v1) are:

$$
\begin{aligned}
&X_1 \sim Be(0.5), \quad X_2 \sim Be(0.65), \\
&X_3 \sim \text{int}[U(0,4)], \quad X_4 \sim \text{int}[U(0,5)], \\
&T \sim Be(p_T), \text{ where} \\
&p_T = \sigma(-5 + 0.05X_2 + 0.25X_3 + 0.6X_4 + 0.4X_2X_4), \\
&Y_1 = \sigma(-1 + 1 - 0.1X_1 + 0.35X_2 + 0.25X_3 + 0.2X_4 + 0.15X_2X_4), \\
&Y_0 = \sigma(-1 + 0 - 0.1X_1 + 0.35X_2 + 0.25X_3 + 0.2X_4 + 0.15X_2X_4),
\end{aligned}
\tag{16}
$$

where int[.] is an operator which rounds the sample to the nearest integer, Be is a Bernoulli distribution, U is a uniform distribution, σ is the sigmoid function, and Y_1 and Y_0 are the counterfactual outcomes when $T = 1$ and $T = 0$, respectively. Covariate X_1 represents biological sex, X_2 represents age category, X_3 represents cancer stage, and X_4 represents comorbidities.

We create a variant (v2) of this DGP by introducing non-linearity into the outcome, and then into the treatment assignment as follows:

$$
Y_1 = \sigma(exp[-1 + 1 - 0.1X_1 + 0.35X_2 + 0.25X_3 + 0.2X_4 + 0.15X_2X_4]). \tag{17}
$$

Figs. 15 and 16 provide information on the propensity scores for the v1 and v2 variants (the second version has the same propensity score generating model as v1).

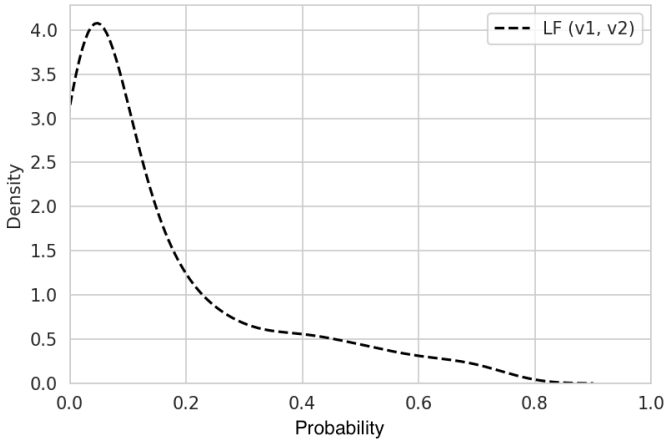

Figure 15: Marginal propensity scores for the LF (v1) and LF (v2) datasets. Note that the minimum probability of treatment in a random draw from the DGP is 0.007. The datasets are intentionally designed such that certain subgroups are unlikely to receive treatment, resulting in near-positivity violations.

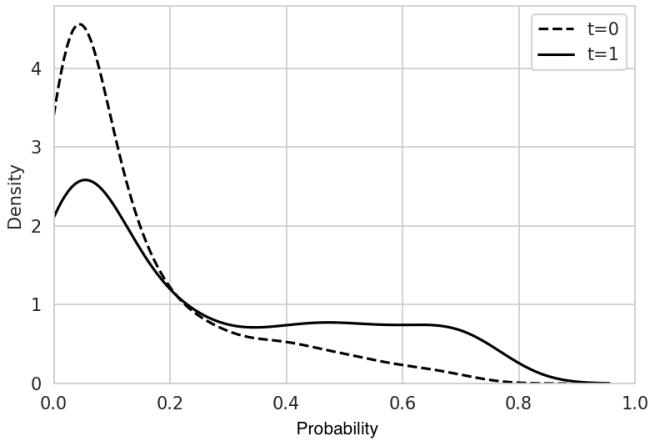

Figure 16: Propensity scores by treatment assignment for a sample from the LF (v1) dataset.

Finally, the generating equations for the generalized version of the LF dataset, which we refer to as 'Gen', are:

$$
\begin{aligned}
&X_1 \sim Be(0.5), \quad X_2 \sim Be(0.65), \\
&X_3 \sim \text{int}[U(0,4)], \quad X_4 \sim \text{int}[U(0,5)], \\
&U_{X_5} \sim N(0,1), \quad X_5 = 0.2X_1 + U_{X_5}, \\
&I_1 \sim N(0,1), \quad I_2 \sim N(0,1) \\
&R_1 \sim N(0,1), \quad R_2 \sim N(0,1) \\
&U_T \sim N(0,1), \quad T \sim Be(p_T), \text{ where} \\
&p_T = \sigma(-5 + 0.05X_2 + 0.25X_3 + 0.6X_4 + 0.4X_2X_4 + 0.15X_5 + 0.1I_1 + 0.15I_2 + 0.1U_T) \\
&U_M \sim N(0,1), \quad M_1 = 0.8 + 0.15U_M, \quad M_0 = 0.15U_M \\
&Y_1 = \sigma(exp[-1 + M_1 - 0.1X_1 + 0.35X_2 + 0.25X_3 + 0.2X_4 + 0.15X_2X_4 + R_1 + R_2]), \\
&Y_0 = \sigma(-1 + M_0 - 0.1X_1 + 0.35X_2 + 0.25X_3 + 0.2X_4 + 0.15X_2X_4 + R_1 + R_2), \\
&Y = Y_1X + Y_0(1 - X) \\
&U_C \sim N(0,1), \quad C = 0.6Y + 0.4X + 0.4U_C,
\end{aligned}
\tag{18}
$$

where int[.] is an operator which rounds the sample to the nearest integer, Be is a Bernoulli distribution, U is a uniform distribution, N is a normal distribution, σ is the sigmoid function, I_1 and I_2 are instrumental variables, R_1 and R_2 are risk variables, M_1 and M_0 are the counterfactual mediators and Y_1 and Y_0 are the counterfactual outcomes when $T = 1$ and $T = 0$, respectively. C is a collider, and finally, X_1-X_5 are confounders.

F Hyperparameters

The LR and SL approaches are implemented using the default algorithms in the scikit-learn package (Pedregosa et al., 2011), whilst the the DragonNet, S-learner, T-learner, and P-learner, are implemented using the CausalML package (Chen et al., 2020). For DragonNet the number of neurons per layer was set to 200, the learning rate set to 1×10^{-1}, number of epochs = 30, and batch size = 64. For TVAE the dimensionality of all latent variables was set to 5, the number of layers set to 2, batch size = 200, number of epochs = 100, learning rate = 5×10^{-4}, and targeted regularization weight of 0.1.

For CFR and MN, we undertake a Monte-Carlo train-test split hyperparameter search with 15 trials, for every one of the 100 samples from the DGP. The best performing set of hyperparameters is then used to

Table 2: Hyperparameter search space for CFR and MN based methods.

Parameter	Min	Max
Batch size	10	64
L2 Weight Penalty	1e-5	1e-3
No. of Iterations	2000	10000
Learning Rate	1e-5	1e-2
No. Layers	2	14
Dropout Prob.	0.1	0.5
No. Neurons per Layer	5	200

train CFR and MN on the full dataset. For the hyperparameter search itself, we undertake 15 trials on a train/test split for each of the 100 samples from the DGP, and additional, separate hyperparameter searches are undertaken for methods using targeted regularization. The hyperparameters explored for CFR and MN based methods are shown in Table 2. Note that the iteration count is not in terms of epochs - it represents the number of batches sampled randomly from the dataset. The number of iterations can be multiplied by the batch size and divided by the dataset size to approximately determine the equivalent number of epochs this represents.

Note that, unlike in traditional supervised learning tasks, using the full data with causal inference is possible because the target estimand is not the same quantity as the quantity used to fit the algorithms (Farrell et al., 2021). Indeed, whilst cross-fitting is used for the hyperparameter search, subsequent use of the full data has been shown to be beneficial, especially in small samples (Curth & van der Schaar, 2021). It is reassuring to note that overfitting is likely to worsen our recorded estimates, rather than misleadingly improve them. Similarly, even though the SL is trained and the corresponding weights derived using a hold-out set, the final algorithm is trained on the full dataset for estimation. Logistic regression is simply trained on the full dataset without any data splitting. This is what motivated us to ask the question as to whether or not it is possible to have a 'free lunch' with IFs. Indeed, if no additional data is required, but we can nonetheless improve our estimates and achieve valid statistical inference (for the purposes, for example, of null hypothesis significance tests), then this would represent a valuable gain. For all treatment models, we bound predictions to fall in the range $[.025, .975]$ (Li et al., 2022).

G Things that Did Not Work

G.1 Calibration

One of the initial possibilities that we considered which might explain why some methods (*e.g.*, CFR) were not performing as well as others, was that the calibration of the output might be poor (Guo et al., 2017). However, we tried calibrating the trained outcome and treatment model networks using temperature scaling. We found it to be unsuccessful, and we leave an exploration of why it failed to future work.

G.2 Restricted Hyperparameter Search

Additionally, we tried only performing hyperparameter search with a held-out test set *once* at the beginning of the 100 subsequent simulations for each model and dataset variant, rather than performing it for every single simulation. This did not work, and we found that if the first network 'designed' through hyperparameter search happened to be degenerate with respect to its performance as a plug-in estimator (notwithstanding its potentially adequate performance as an outcome model), then it will be degenerate for all simulations, and yield incredibly biased results. However, performing hyperparameter search for every simulation more accurately represents the use of these algorithms in practice.

This problem also highlights the importance of fitting multiple neural networks on the same data. As supervision is not available, the usual metrics for hyperparameter search (based on *e.g.*, held out data loss scores) can be a poor indicator for the efficacy of the network as a plug-in estimator. By re-performing hyperparameter search, even on the same data (put perhaps, with different splits), one can effectively bootstrap to average out the variability associated with the hyperparameter search itself. Indeed, as the

results show, the average estimates for the ATE using CFR net are close to the true ATE, even if the variance of the estimation is relatively high. We leave a comparison of the contribution of variance from hyperparameter search to further work.

G.3 MultiStep Update Variants

Relating to our proposed MultiStep objective, we also tried a non-linear, generalized variant with an objective which still attempts to minimize the mean and variance of the influence function but such that the update step is parameterized as follows:

$$\hat{Q}(t, \mathbf{x}_i) + g_\theta(\nu_1 \hat{Q}(t, \mathbf{x}_i), \nu_2 H(\mathbf{z}_i)) \tag{19}$$

It can be seen that instead of optimizing over the domain of $\hat{\gamma} \in \Gamma$ in Eq. 6, we instead optimize over $\theta \in \Theta$, where θ are the parameters of a shallow NN function g. Here, $\nu_1 \in \{0, 1\}$ and $\nu_2 \in \{0, 1\}$ are hyperparameters determining whether the NN function g_θ should be taken over just the clever covariate H, or over both the clever covariate and the outcome model Q.

In practice however, this approach did not yield good estimates. Furthermore, we found that MultiStep update step given in Eq. 5, setting $\alpha_1 = 0$ (*i.e.*, no mean-zero penalty) also did not work well. This result was surprising because a similar approach in Neugebauer & van der Laan (2005), which did not include a mean-zero penalty, yielded an improvement. However, it is also intuitive that if the two properties of the Efficient Influence Function are (1) mean-zero and (2) minimum variance, then it makes sense that an optimization objective should benefit from the inclusion of both of these conditions.

H Initial Evaluation

Table 3: Initial results over a restricted set of model variations. All update steps use the same propensity score G- algorithm as their Q-model algorithm, unless indicated by 'w/ G-SL', which indicates the use of a SuperLearner. Mean Squared Errors (MSE) and standard error (s.e.) (lower is better) and Shapiro-Wilk test p-values for normality (higher is better) for 100 simulations. Best results are those competing across all three dimensions. **Bold** indicates the best result for each algorithm. Multiple methods may perform approximately equally well.

Dataset	**Q Model**	**U-Base**			**U-ones**			**U-sub**			**Treg**			**Treg+U-sub**			**U-ones w/ G-SL**			**U-sub w/ G-SL**		
		p	MSE	s.e.	p	MSE	s.e.	p	MSE	s.e.	p	MSE	s.e.	p	MSE	s.e.	p	MSE	s.e.	p	MSE	s.e.
	LR	.001	.0004	.002	**.276**	**.0007**	**.003**	**.248**	**.0008**	**.003**	-	-	-	-	-	-	**.378**	**.0006**	**.003**	**.591**	**.0008**	**.003**
	SL	.001	.0004	.002	**.53**	**.0008**	**.003**	**.651**	**.0009**	**.003**	-	-	-	-	-	-	-	-	-	-	-	-
	CFR	.0	.0114	.008	.001	.0042	.004	.01	.01	.003	.07	.0113	.008	.0	.0105	.002	**.396**	**.0006**	**.003**	.909	.0015	.003
LF (v1)	MN-Inc	.052	.0008	.003	**.78**	**.0007**	**.003**	.394	.001	.003	.729	.0012	.003	.681	.001	.003	**.639**	**.0008**	**.003**	.329	.001	.003
	MN-Inc+LM	.135	.0009	.003	**.141**	**.0007**	**.003**	.578	.0009	.003	.0	.0017	.004	.957	.0011	.003	**.969**	**.0008**	**.003**	.786	.0009	.003
	MN-Casc	.0	.0018	.004	.231	.0014	.002	.0	.0018	.003	.083	.0086	.007	.702	.0045	.004	**.831**	**.0007**	**.003**	.339	.0009	.003
$n = 5000$	MN-Casc+LM	.053	.0058	.006	.018	.002	.003	.204	.0037	.003	.0	.0091	.008	.74	.0036	.003	**.747**	**.0007**	**.003**	.625	.001	.003
	LR	.066	.0024	.002	**.752**	**.0007**	**.003**	**.497**	**.0008**	**.003**	-	-	-	-	-	-	**.785**	**.0007**	**.003**	**.867**	**.0009**	**.003**
	SL	.349	.0017	.003	**.938**	**.0008**	**.003**	**.92**	**.0009**	**.003**	-	-	-	-	-	-	-	-	-	-	-	-
	CFR	.0	.0185	.01	.0	.006	.005	.0	.0151	.002	.0	.035	.01	.008	.0162	.002	**.623**	**.0007**	**.003**	.065	.0015	.003
LF (v2)	MN-Inc	.119	.001	.003	**.204**	**.0006**	**.003**	.211	.0008	.003	.002	.0009	.002	**.029**	**.0008**	**.003**	.058	.0007	.003	**.049**	**.0008**	**.003**
	MN-Inc+LM	.0	.0011	.003	**.438**	**.0009**	**.003**	.813	.0011	.003	.139	.0071	.005	.678	.0026	.003	**.959**	**.0005**	**.002**	**.949**	**.0009**	**.003**
	MN-Casc	.0	.002	.004	.013	.0033	.002	.892	.0043	.003	.77	.014	.006	.365	.0101	.002	**.272**	**.0007**	**.003**	.264	.0011	.003
$n = 5000$	MN-Casc+LM	.257	.0113	.007	.349	.0032	.003	.001	.0083	.002	.066	.0295	.007	.0	.0112	.002	**.897**	**.0006**	**.003**	.241	.0013	.003
	LR	.022	.1818	.019	.0	.0576	.035	.0	.0461	.044	-	-	-	-	-	-	.0	.1322	.019	**.0**	**.0597**	**.03**
	SL	.0	.0466	.032	**.0**	**.0311**	**.033**	.0	.0346	.034	-	-	-	-	-	-	-	-	-	-	-	-
	CFR	.0	.7709	.098	.0	.2865	.074	**.0**	**.0439**	**.052**	.0	25.5	.3	.0	.0604	.051	.0	.2626	.063	.0	1.7	.114
IHDP	MN-Inc	.0	.0324	.042	**.0**	**.0297**	**.044**	.0	8.7	.299	.0	.0482	.042	.0	30.8	.537	**.0**	**.0243**	**.044**	.0	.0425	.042
	MN-Inc+LM	.0	.0393	.045	**.0**	**.0259**	**.043**	.0	.9849	.099	.0	.1332	.038	.0	1.9	.138	**.0**	**.0243**	**.044**	.0	.0327	.042
	MN-Casc	.0	.1977	.046	.0	.0737	.04	.0	.064	.04	.0	2.9	.115	.0	.102	.042	.0	.0816	.042	**.0**	**.0383**	**.047**
$n = 747$	MN-Casc+LM	.0	4.7	.158	.0	1.4	.093	.0	.2118	.049	.0	23.9	.164	**.0**	**.1824**	**.06**	.0	1.1	.079	.0	4.7	.202

H.1 Initial Evaluation

We share initial results in Table 3. These results were used to inform a subsequent set of experiments with a restricted set of variants. Specifically, we used these to select the most successful variant of MultiNet.

For **LF (v1)**, we see that the base CFR performs significantly worse in all considered metrics than LR and SL. Base LR and base SL achieved the best results in terms of MSE and s.e., although note that none of

the base algorithms achieve asymptotic normality. Notice that LR's base MSE performance on LF (v1) is actually better than its MSE performance using the one-step and submodel updates. Such behaviour has been noted before by Luque-Fernandez et al. (2018), and occurs when the base learner is already close and/or when both outcome and treatment models are misspecified. Unlike CFR, our *MN-Inc* and *MN-Casc* variants worked well as either outcome or treatment models, yielding the best results with the one-step update. The other two of our *MN-* variants also performed well with the one-step and submodel updates but required a SL treatment model to do so.

The potential improvements for LR in combination with update steps is more striking for **LF (v2)**. Here, the LR base outcome model is misspecified (LF v2 has an exponential outcome model). Combining the LR with the SL one-step and submodel update processes enabled the LR method to perform well in spite of the non-linearity of the outcome. This is a demonstration of double-robustness - even though the outcome model is misspecified, the treatment model is not (or at least, it is sufficiently correctly specified), owing to the use of a SL, and the estimates are improved. As with the LF (v1) dataset, combining CFR with IFs resulted in a substantial improvement, especially when using an SL treatment model, yielding a competitive MSE, s.e., and normally distributed estimates (thus amenable to statistical inference). These results demonstrate the power of semiparametric methods for improving our estimation with NNs, and again illustrate the double-robustness property: the CFR outcome model was poorly specified, but was able to recover with an SL treatment model. Similar performance for our *MN-* variants on LF (v1) was observed with LF (v2).

Unfortunately, no method variant yielded normally distributed estimates with the **IHDP** dataset. The worst performing estimator across any combination of semiparametric techniques was LR. This makes sense given the non-linearity in the IHDP outcome process (Curth et al., 2021b). The SL with the one-step or submodel updates performed equally (poorly) as the best CFR and *MN-Casc* variants, although the SL provided a smaller s.e.. Overall, the best methods were our *MN-Inc* and *MN-Inc+LM* variants in combination with either a one-step update, or a one-step update using a SL treatment model.

The MultiNet variant which performed the best and most consistently across **all datasets** was our *MN-Inc* (or equally, *MN-Inc+LM*) with the one-step update. Whereas other methods benefited from the help of a SL treatment model, *MN-Inc* worked well as both an outcome and a treatment model, making it the best all-rounder across datasets, as well as the least dependent on the SL for correction. For all NN based approaches, targeted regularization made little difference, and sometimes resulted in instability and high MSEs. Further work is required to investigate this, although it may relate to which treatment model is used, and the associated sensitivity to positivity violations. A prior application also described the potential for the regularization to be inconsistent (Shi et al., 2019).

For all base learners, we observe the potential for improvement using the semiparameteric techniques, primarily for improving the associated MSE. It is also worth noting that in general, the base CFR method has consistently higher (*i.e.*, worse) s.e. than the *MN-variants*, although combining CFR with an udpdate step (*e.g.*, one-step w/ SL) significantly tightened the s.e..

I Additional Results/Analysis

In Table 4 we also provide a set of results (without any marginalization over any of the G-, U-, or Q-dimensions) for a subset of the methods considered in the full-factorial design. Note that whilst we have tried to highlight the best results in bold and underline, many of the results are close/competitive and illustrate (again) that the performance is dataset and combination dependent, as well as that there exist multiple possible 'best' options for a given situation.

In Table 5 we provide results for the absolute difference between the true average treatment effect and the estimated average treatment effect (aeATE) over the $r = 100$ simulations. We define this as $\sum_i^r |\tau_i - \hat{\tau}_i|$. Note that the results here were conducted over different simulations to the results in the other tables (hence why we report again the p-values and standard errors).

Table 4: Unmarginalized results over a restricted set of model variations including the Double Machine Learning (DML) method (Chernozhukov et al., 2018). Mean Squared Errors (MSE) and standard error (s.e.) (lower is better) and Shapiro-Wilk test p-values for normality (higher is better) are provided ad computed over 100 simulations. Best results are those competing across all three dimensions. **Bold** indicates the best result for each algorithm. Multiple methods may perform approximately equally well.

Dataset	**Q Model**	**U-Base**			**G-SL + U-multi**			**G-SL+U-sub**			**G-MN+U-multi**			**G-MN+U-sub**		
		p	MSE	s.e.	p	MSE	s.e.	p	MSE	s.e.	p	MSE	s.e.	p	MSE	s.e.
LF (v1) ($n = 500$)	Q-MN	**0.985**	**0.0024**	**0.054**	**0.959**	**0.0023**	**0.054**	0.027	0.0104	0.109	**0.945**	**0.0024**	**0.044**	0.125	0.0080	0.093
	Q-SL	0.030	0.0032	0.057	0.034	0.0036	0.060	0.023	0.0091	0.102	0.043	0.0035	0.060	0.093	0.0081	0.095
	Q-TVAE	0.617	0.0053	0.050	0.386	0.0042	0.052	0.074	0.0104	0.108	**0.716**	**0.0046**	**0.052**	0.049	0.0083	0.093
	Q-T	0.700	0.0059	0.064	0.772	0.0056	0.065	0.495	0.0084	0.099	**0.843**	**0.0057**	**0.064**	0.059	0.0073	0.089
	DML	0.113	0.0055	0.064												
LF (v1) ($n = 5000$)	Q-MN	0.000	0.0006	0.025	0.0	0.0005	0.024	**0.599**	**0.0009**	**0.031**	0.0	0.0006	0.025	0.583	0.0307	0.030
	Q-SL	0.024	0.0004	0.022	0.055	0.0005	0.023	**0.536**	**0.0009**	**0.030**	0.012	0.0004	0.023	0.224	0.0008	0.029
	Q-TVAE	0.444	0.0007	0.024	0.182	0.0006	0.025	**0.655**	**0.0009**	**0.031**	0.368	0.0007	0.024	0.414	0.0008	0.030
	Q-T	**0.975**	**0.0010**	**0.032**	0.989	0.0010	0.032	**0.935**	**0.0010**	**0.033**	0.823	0.0010	0.032	0.854	0.0010	0.033
	DML	0.717	0.0012	0.028												
LF (v1) ($n = 10000$)	Q-MN	0.0	0.0008	0.027	0.0	0.0007	0.025	0.366	0.0004	0.020	0.0	0.0006	0.023	**0.648**	**0.0006**	**0.022**
	Q-SL	1.0	0.0002	0.015	**0.993**	**0.0002**	**0.016**	0.625	0.0004	0.019	**0.997**	**0.0002**	**0.015**	**0.996**	**0.0004**	**0.019**
	Q-TVAE	**0.666**	**0.0004**	**0.019**	0.956	0.0004	0.019	0.392	0.0004	0.020	**0.782**	**0.0004**	**0.018**	**0.861**	**0.0004**	**0.020**
	Q-T	0.341	0.0004	0.020	0.331	0.0004	0.021	**0.326**	**0.0004**	**0.021**	0.141	0.0004	0.021	0.122	0.0004	0.021
	DML	0.777	0.0005	0.019												
LF (v2) ($n = 500$)	Q-MN	**0.439**	**0.0018**	**0.043**	0.149	0.0018	0.044	0.014	0.0119	0.114	0.075	0.0019	0.044	0.0	0.0102	0.108
	Q-SL	0.0	0.0027	0.054	0.001	0.0029	0.057	0.025	0.0097	0.103	0.0	0.0029	0.057	0.0	0.0095	0.103
	Q-TVAE	0.097	0.0075	0.043	0.031	0.0059	0.045	0.046	0.0125	0.114	0.016	0.0067	0.045	0.0	0.0112	0.112
	Q-T	**0.817**	**0.0066**	**0.064**	**0.721**	**0.0062**	**0.065**	0.047	0.0094	0.102	**0.753**	**0.0064**	**0.065**	0.003	0.0110	0.108
	DML	0.017	0.0050	0.057												
LF (v2) ($n = 5000$)	Q-MN	0.0	0.0011	0.029	0.0	0.0009	0.027	0.925	0.0010	0.031	0.0	0.0011	0.027	**0.662**	**0.0010**	**0.030**
	Q-SL	**0.765**	**0.0008**	**0.028**	0.669	0.0008	0.028	**0.921**	**0.0009**	**0.030**	0.661	0.0007	0.028	0.678	0.0008	0.030
	Q-TVAE	0.607	0.0007	0.028	0.795	0.0007	0.027	**0.912**	**0.0009**	**0.031**	**0.799**	**0.0006**	**0.027**	0.672	0.0008	0.029
	Q-T	**0.793**	**0.0010**	**0.032**	0.644	0.0010	0.032	**0.724**	**0.0011**	**0.033**	**0.755**	**0.0010**	**0.032**	0.0	0.0010	0.037
	DML	0.824	0.0012	0.032												
LF (v2) ($n = 10000$)	Q-MN	0.0	0.0009	0.027	0.0	0.0008	0.025	**0.707**	**0.0005**	**0.020**	0.0	0.0011	0.025	**0.783**	**0.0009**	**0.026**
	Q-SL	**0.859**	**0.0004**	**0.019**	0.649	0.0004	0.019	**0.964**	**0.0005**	**0.020**	0.594	0.0004	0.019	0.676	0.0004	0.021
	Q-TVAE	0.0	0.0177	0.412	0.448	0.0004	0.021	**0.981**	**0.0004**	**0.020**	0.240	0.0004	0.021	0.227	0.0004	0.021
	Q-T	0.0	0.0124	0.430	**0.719**	**0.0005**	**0.022**	0.566	0.0005	0.022	0.574	0.0005	0.022	0.285	0.0005	0.022
	DML	0.800	0.0005	0.022												
Gen ($n = 500$)	Q-MN	0.673	0.0031	0.0449	0.7375	0.0026	0.0434	0	0.0086	0.0928	0.6498	0.0025	0.0429	0.0001	0.0053	0.074
	Q-SL	0.9082	0.0035	0.0436	0.8841	0.0035	0.0458	0	0.0062	0.0779	0.4463	0.0035	0.0453	0.0001	0.005	0.0725
	Q-TVAE	0.8044	0.0015	0.0356	**0.8349**	**0.0013**	**0.0373**	0	0.0101	0.1013	0.6077	0.0014	0.0368	0	0.0055	0.076
	Q-T	0.5179	0.0025	0.0496	0.5521	0.0024	0.0502	0.0069	0.0058	0.0795	0.5266	0.0025	0.0499	0.0044	0.0045	0.0696
	DML	0.7614	0.0024	0.0446												
Gen ($n = 5000$)	Q-MN	0	0.0019	0.0347	0	0.0017	0.0311	0.0133	0.0007	0.0271	0	0.0015	0.0309	0.7184	0.0004	0.0227
	Q-SL	**0.5309**	**0.0003**	**0.0186**	**0.2679**	**0.0003**	**0.0188**	0.4612	0.0006	0.0249	**0.2474**	**0.0003**	**0.0189**	0.9423	0.0005	0.0235
	Q-TVAE	0.2001	0.0005	0.0218	0.3553	0.0004	0.0208	0.3302	0.0006	0.0259	0.487	0.0005	0.0216	0.9806	0.0005	0.0244
	Q-T	0.748	0.0011	0.0246	0.7697	0.001	0.025	0.8191	0.0011	0.0254	0.5491	0.001	0.0244	0.9	0.0011	0.0253
	DML	0.7733	0.0005	0.0198												
Gen ($n = 10000$)	Q-MN	0	0.0015	0.0243	0	0.0013	0.0223	0.101	0.0004	0.0193	0	0.0011	0.0221	0.0522	0.0003	0.0168
	Q-SL	**0.7922**	**0.0001**	**0.0123**	0.4591	0.0002	0.013	0.4056	0.0003	0.0167	**0.5275**	**0.0001**	**0.0127**	0.0091	0.0003	0.0164
	Q-TVAE	0.5489	0.0003	0.0163	0.8935	0.0002	0.0151	0.4025	0.0003	0.0171	0.3575	0.0002	0.0156	0.0491	0.0003	0.0172
	Q-T	0.0717	0.0005	0.0174	0.0515	0.0005	0.0177	0.0653	0.0005	0.0176	0.0636	0.0005	0.0176	0.04	0.0005	0.0175
	DML	0.8017	0.0003	0.0137												
IHDP ($n = 747$)	Q-MN	0.0	0.0647	0.409	0.0	0.0451	0.405	0.0	0.0390	0.400	0.0	0.0515	0.411	0.0	0.614	0.844
	Q-SL	0.0	0.0440	0.317	0.0	0.0353	0.346	0.0	0.0340	0.344	0.0	0.0390	0.346	0.0	0.625	0.846
	Q-TVAE	0.0	0.0178	0.412	0.0	0.0132	0.404	0.0	0.0134	0.405	0.0	0.0142	0.408	0.0	0.191	0.582
	Q-T	0.0	0.0124	0.430	0.0	0.0123	0.422	**0.0**	**0.0118**	**0.424**	0.0	0.0125	0.426	0.0	0.0293	0.446
	DML	0.0	0.0129	0.441												

I.1 Shapley Results

In addition to the interaction plots given in the text, here we provide results for the regressor performance and the predictor impacts.

I.1.1 Regressor Performance

Table 6 show the performance of the random forest regressor in predicting the three outcomes (MSE, ATE estimate standard error, and p-value). These results help us understand whether there is any information in the set of predictors which is useful in predicting the outcome (and therefore, in turn, whether there exist any potentially meaningful patterns). We provide results for the fraction of explained variance, R^2, and MSE. The results are averages over a 10-fold cross-validated evaluation. It is useful to interrogate this table first to understand whether any further investigation is needed - if the predictive performance is poor,

Table 5: Unmarginalized results over a restricted set of model variations including the Double Machine Learning (DML) method (Chernozhukov et al., 2018). Average absolute error on the ATE (aeATE) and standard error (s.e.) (lower is better) and Shapiro-Wilk test p-values for normality (higher is better) are provided ad computed over 100 simulations. Best results are those competing across all three dimensions, although aeATE is prioritised. **Bold** indicates the best result for each algorithm. Multiple methods may perform approximately equally well.

Dataset	**Q Model**	**U-Base**			**G-SL + U-multi**			**G-SL+U-sub**			**G-MN+U-multi**			**G-MN+U-sub**		
		p	aeATE	s.e.	p	aeATE	s.e.	p	aeATE	s.e.	p	aeATE	s.e.	p	aeATE	s.e.
	Q-MN	**0.3357**	**0.0385**	**0.0286**	**0.251**	**0.0379**	**0.0275**	0.0323	0.0799	0.0613	**0.1368**	**0.0378**	**0.0269**	0.0045	0.0768	0.0605
	Q-SL	0.0318	0.0429	0.0375	0.0354	0.0452	0.0391	0.0204	0.0759	0.0578	0.0375	0.0443	0.0388	0.0202	0.0758	0.0577
LF (v1)	Q-TVAE	0.0698	0.0642	0.0354	0.1008	0.0552	0.0349	0.0728	0.081	0.0632	0.1043	0.0584	0.0346	0.0029	0.0806	0.059
(n=500)	Q-T	0.7005	0.063	0.0442	0.7785	0.0613	0.0434	0.4219	0.0737	0.0552	0.7178	0.0618	0.0429	0.1244	0.0734	0.0526
	DML	0.1225	0.0605	0.0433												
	Q-MN	0.0349	0.0207	0.0164	**0.0037**	**0.0178**	**0.0145**	0.6143	0.0248	0.0173	**0.011**	**0.0182**	**0.0139**	0.5692	0.0237	0.0179
	Q-SL	**0.0216**	**0.0163**	**0.013**	**0.0519**	**0.0173**	**0.0135**	0.5245	0.0237	0.0175	**0.0657**	**0.0166**	**0.0129**	0.2467	0.0217	0.0164
LF (v1)	Q-TVAE	0.2875	0.0228	0.0179	0.2168	0.0202	0.0168	0.6859	0.0243	0.018	0.16	0.0213	0.0173	0.465	0.0227	0.018
(n=5000)	Q-T	0.9751	0.0241	0.0193	0.9719	0.0246	0.0189	0.9311	0.0254	0.0198	0.9949	0.0245	0.0194	0.6863	0.0254	0.0199
	DML	0.7166	0.0278	0.0199												
	Q-MN	0	0.0192	0.0188	0	0.0175	0.0189	0.3163	0.0163	0.0109	0	0.0184	0.0189	0.2524	0.0189	0.0135
	Q-SL	**0.9998**	**0.012**	**0.0095**	**0.9372**	**0.0129**	**0.0097**	0.613	0.016	0.011	**0.9926**	**0.0118**	**0.0092**	0.3251	0.0149	0.0108
LF (v1)	Q-TVAE	0.2716	0.0153	0.0118	0.3701	0.0143	0.0095	0.5634	0.0161	0.0112	0.6847	0.0148	0.0104	0.1576	0.0154	0.0112
(n=10000)	Q-T	0.3408	0.0172	0.0113	0.3254	0.0179	0.0117	0.3335	0.0177	0.0116	0.3688	0.0177	0.0117	0.4915	0.0171	0.0116
	DML	0.777	0.0174	0.0133												
	Q-MN	**0.4521**	**0.0325**	**0.0248**	**0.1313**	**0.0326**	**0.0241**	0.015	0.0803	0.0669	**0.2789**	**0.0346**	**0.0238**	0.0009	0.0724	0.0588
	Q-SL	0.0001	0.0366	0.0328	0.0001	0.0384	0.0342	0.0086	0.0723	0.0572	0.0002	0.0385	0.0345	0.0001	0.0708	0.0564
LF (v2)	Q-TVAE	0.0806	0.0708	0.0354	0.0662	0.061	0.0342	0.0035	0.0813	0.0679	0.0203	0.0655	0.0355	0	0.076	0.0601
(n=500)	Q-T	0.8333	0.0609	0.0402	0.735	0.0586	0.0394	0.0103	0.0693	0.0587	0.8407	0.0598	0.0393	0.0079	0.0703	0.0585
	DML	0.0167	0.0604	0.0368												
	Q-MN	0	0.0277	0.0247	0	0.0247	0.0225	0.8417	0.0251	0.0181	0	0.0283	0.0222	0.6407	0.0262	0.0203
	Q-SL	0.7856	0.0229	0.0159	0.9039	0.0219	0.0158	0.9209	0.0249	0.0179	0.702	0.0227	0.0158	0.5783	0.0244	0.0184
LF (v2)	Q-TVAE	**0.908**	**0.0225**	**0.0159**	**0.6827**	**0.0218**	**0.0153**	0.7311	0.0242	0.0184	0.4674	0.0224	0.0147	0.4977	0.0233	0.018
(n=5000)	Q-T	0.7928	0.0249	0.0192	0.8973	0.0252	0.0195	0.7337	0.0259	0.0197	0.7734	0.0257	0.0198	0.0001	0.0271	0.0254
	DML	0.824	0.0273	0.0206												
	Q-MN	0	0.0287	0.0206	0	0.0266	0.0224	0.8711	0.018	0.0131	0	0.0293	0.0192	0.1258	0.0246	0.0185
	Q-SL	**0.8728**	**0.0152**	**0.0113**	0.7803	0.0159	0.0117	0.9735	0.0175	0.0125	0.8779	0.0156	0.0115	0.321	0.0173	0.0118
LF (v2)	Q-TVAE	0.5869	0.0163	0.0116	0.3849	0.0158	0.0113	0.9698	0.016	0.0115	**0.3413**	**0.0155**	**0.0108**	0.2403	0.0165	0.011
(n=10000)	Q-T	0.647	0.0178	0.0128	0.7031	0.0184	0.0131	0.5627	0.018	0.0131	0.3893	0.0179	0.0127	0.4665	0.0186	0.0133
	DML	0.7998	0.0175	0.0136												
	Q-MN	0.0012	0.0433	0.0379	0	0.0417	0.0381	0.0004	0.0712	0.0559	0	0.0405	0.0369	0	0.0512	0.0527
	Q-SL	0.9273	0.0512	0.0355	0.892	0.0501	0.0365	0	0.0619	0.0444	0.7328	0.0496	0.0362	0	0.0503	0.046
Gen	Q-TVAE	0.0383	0.0321	0.0201	**0.0579**	**0.0303**	**0.0182**	0.0003	0.0754	0.0619	0.103	0.0322	0.0195	0	0.0583	0.0513
(n=500)	Q-T	0.0595	0.0369	0.0253	0.0753	0.0369	0.0247	0.1224	0.056	0.0436	0.0515	0.0375	0.0248	0.0002	0.0513	0.0405
	DML	0.6547	0.0387	0.0248												
	Q-MN	0	0.037	0.0307	0	0.0349	0.0304	0.7922	0.0206	0.0155	0	0.0312	0.0291	0.2339	0.0174	0.0137
	Q-SL	**0.514**	**0.0131**	**0.0107**	0.5157	0.0137	0.0111	0.4922	0.0187	0.0148	0.2609	0.0133	0.0111	0.8739	0.0175	0.0138
Gen	Q-TVAE	0.9298	0.0187	0.0139	0.9347	0.0166	0.0134	0.4052	0.0185	0.015	0.8204	0.0176	0.0139	0.6559	0.0187	0.0145
(n=5000)	Q-T	0.748	0.0276	0.0178	0.8857	0.0267	0.0178	0.8492	0.0275	0.0183	0.8432	0.0269	0.0176	0.0702	0.0291	0.0206
	DML	0.868	0.0188	0.0127												
	Q-MN	0	0.0329	0.0323	0	0.0306	0.0329	0.0109	0.0123	0.0144	0	0.027	0.032	0.3189	0.01	0.0071
	Q-SL	**0.9635**	**0.0083**	**0.0068**	0.9686	0.0085	0.0063	0.2335	0.0111	0.0104	**0.9994**	**0.0082**	**0.0062**	0.6388	0.0102	0.0077
Gen	Q-TVAE	0.6881	0.0134	0.0092	0.5343	0.0116	0.0086	0.6199	0.0117	0.0111	0.6207	0.0113	0.0076	0.7079	0.0115	0.0083
(n=10000)	Q-T	0.3724	0.0196	0.0105	0.3895	0.0184	0.0106	0.4392	0.0191	0.0103	0.5978	0.0188	0.0106	0.3361	0.019	0.0105
	DML	0.9949	0.0143	0.0091												
	Q-MN	0	0.2142	0.3337	0	0.2062	0.2972	0	0.1573	0.2141	0	0.244	0.475	0	0.3005	0.7104
	Q-SL	0	0.1474	0.1489	0	0.1292	0.1233	0	0.1344	0.1277	0	0.1543	0.2658	0	0.1832	0.2403
IHDP	Q-TVAE	0	0.1123	0.0772	0	0.0993	0.0671	0	0.0912	0.0707	0	0.1289	0.1893	0	0.2085	0.2932
(n=747)	Q-T	0	0.0895	0.0668	0	0.0889	0.0667	**0**	**0.0877**	**0.0655**	0	0.1118	0.1742	0	0.1328	0.1681
	DML	0	0.0919	0.0667												

there is no point explaining the regressor's behaviour; if it is good, then it is worth investigating what the regressor is using to achieve that performance.

From Table 6 it can be seen that all outcomes were somewhat well predicted, with the arguable exception of the MSE, which had large standard errors for the R^2 and fraction of explained variance suggesting that test data in some of the folds in the 10-fold cross-validation scheme were poorly predicted, whilst others were predicted well. Even though R^2 and explained variance are not reliable metrics in non-linear regression tasks, they can nonetheless be seen that a relatively high R2 is achieved, indicating that information about the MSE is predictable from the set of predictors. This was especially true for the prediction of the Shapiro-Wilk test p-values and ATE estimate standard error, which both had average fraction of explained variance and average R^2 greater than 0.7.

Table 6: Meta-analysis results for random forest regression performance for MSE, standard error of the estimates (ATE s.e.) and p-values as the outcomes. Results for R^2, explained variance, and MSE are given as the mean ± the standard derivation across the 10-fold cross-validation procedure.

Outcomes:	MSE	ATE s.e.	p-value
R^2	0.47±0.667	0.97±0.027	0.47±0.667
Explained Variance	0.48±0.658	0.97±0.027	0.48±0.658
MSE	0.0005±0.0006	0.0003±0.00004	0.0005±0.0006

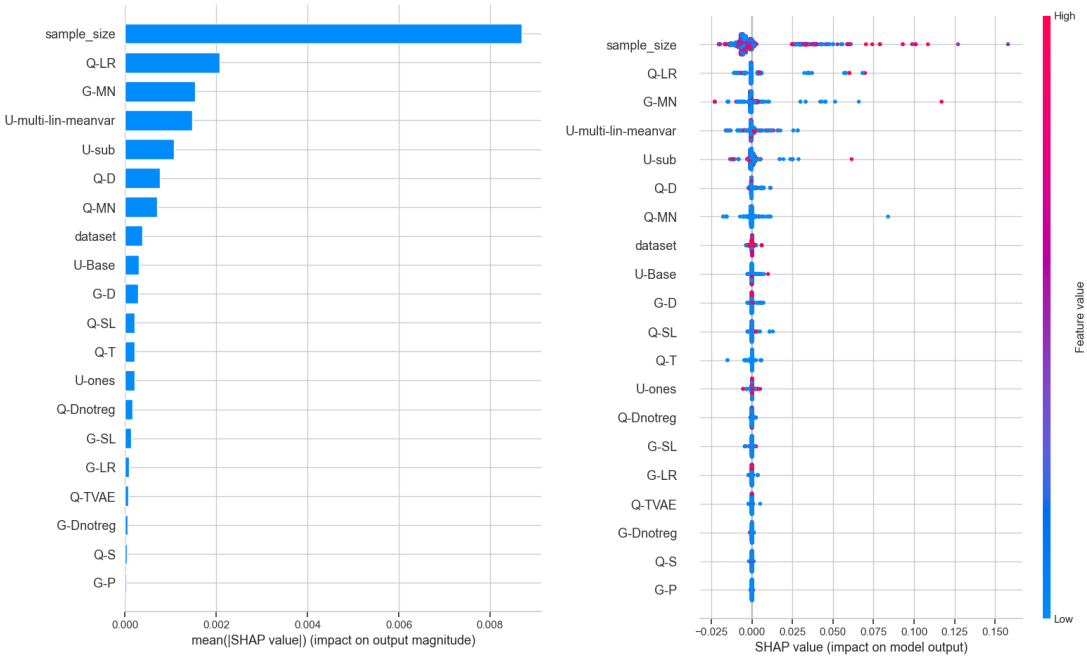

Figure 17: Shapley predictor random forest regressor impact results for the MSE.

I.1.2 Shapley Predictor Impact

Given the regression performance results indicate some patterns exist in the data, let us now turn to the Shapley explainability results. For each of the three respective outcomes (MSE, ATE estimate standard error, and p-values), Figs. 17, 18, and 19 depict the global predictor impacts on the regressor (left-hand plots) as well as the per-predictor, per-datapoint impact on the regressor (right-hand plots).

The right-hand side plots in Figs. 17-19 are useful in visualising the spread of impact, and in which direction this impact is (*i.e.*, does it push the prediction of the outcome up or down in value). For example, in Fig. 17 one can see that the dataset is the most important predictor for predicting MSE, followed by Q-CFR, G-MN, and U-sub. We see that most of the impact (for all predictors) is clustered around 0, which indicates that they are stable in terms of their relationship to MSE. However, the right-hand side plot shows that some outliers exist, with certain combinations yielding higher (or, to a lesser extent, lower) MSEs.

In Fig. 18 for ATE estimate standard error we again see dataset as the most important predictor, followed by Q-CFR and sample size. There is a larger variance in the impact these predictors have on the outcome than for MSE. Finally, looking at Fig. 19 for the p-values, we see heavy dependence on sample size. Note that the p-value outcome here is a function of a *different* sample size to the one used in the regression - it is computed based on the number of simulations which is fixed for all experiments. Thus it is still interesting to note that sample size was important as a determinant of the Shapiro-Wilk test for normality p-value.

Perhaps more interesting than these predictor impact results are the interaction plots, which are given in Section 6.4.

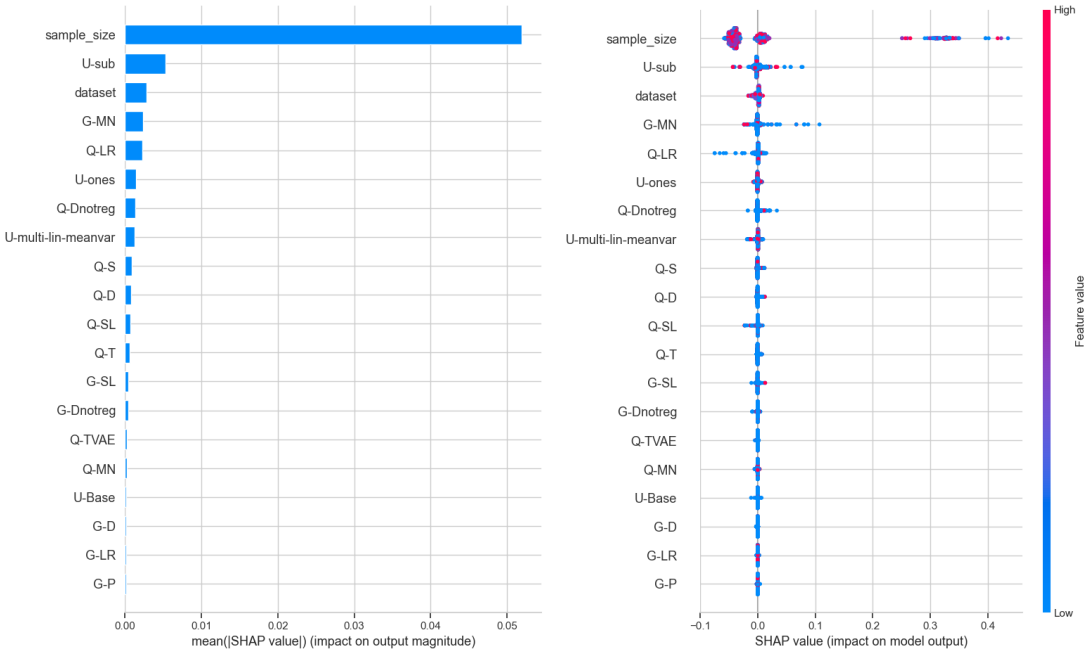

Figure 18: Shapley predictor random forest regressor impact results for the ATE estimate standard error.

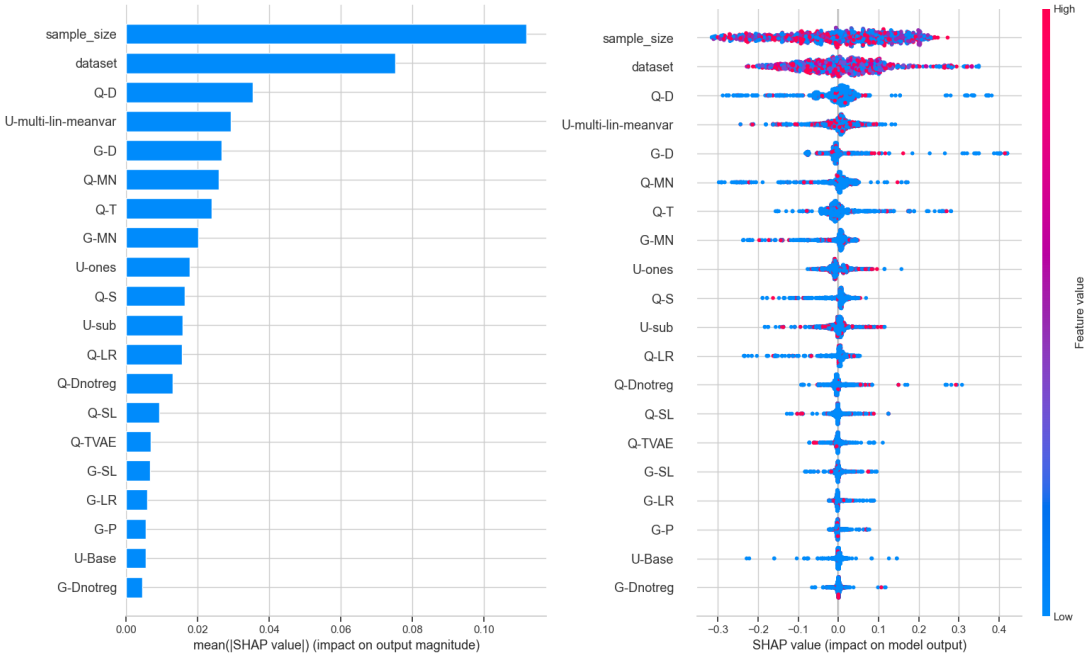

Figure 19: Shapley predictor random forest regressor impact results for the Shapiro-Wilk test p-values.

I.2 Alternative Perspectives

In the main text we provided summary results by estimating the probability that a particular Q (outcome), G (propensity), or U (update step) method would result in a performance advantage. This was done because the number of results was large, making it difficult to judge the efficacy of a method in isolation. In Figs. 20-26 we provide the complete results for each of the seven dataset variants: LF (v1) with $n = \{500, 5000, 10000\}$,

LF (v2) with $n = \{500, 5000, 10000\}$, and the IHDP dataset $n = 747$. For each figure we provide the comparison of each Q-method with each G- and U-method, and include a red dashed line to include the base method (just the Q-method without the IF update step) for comparison.

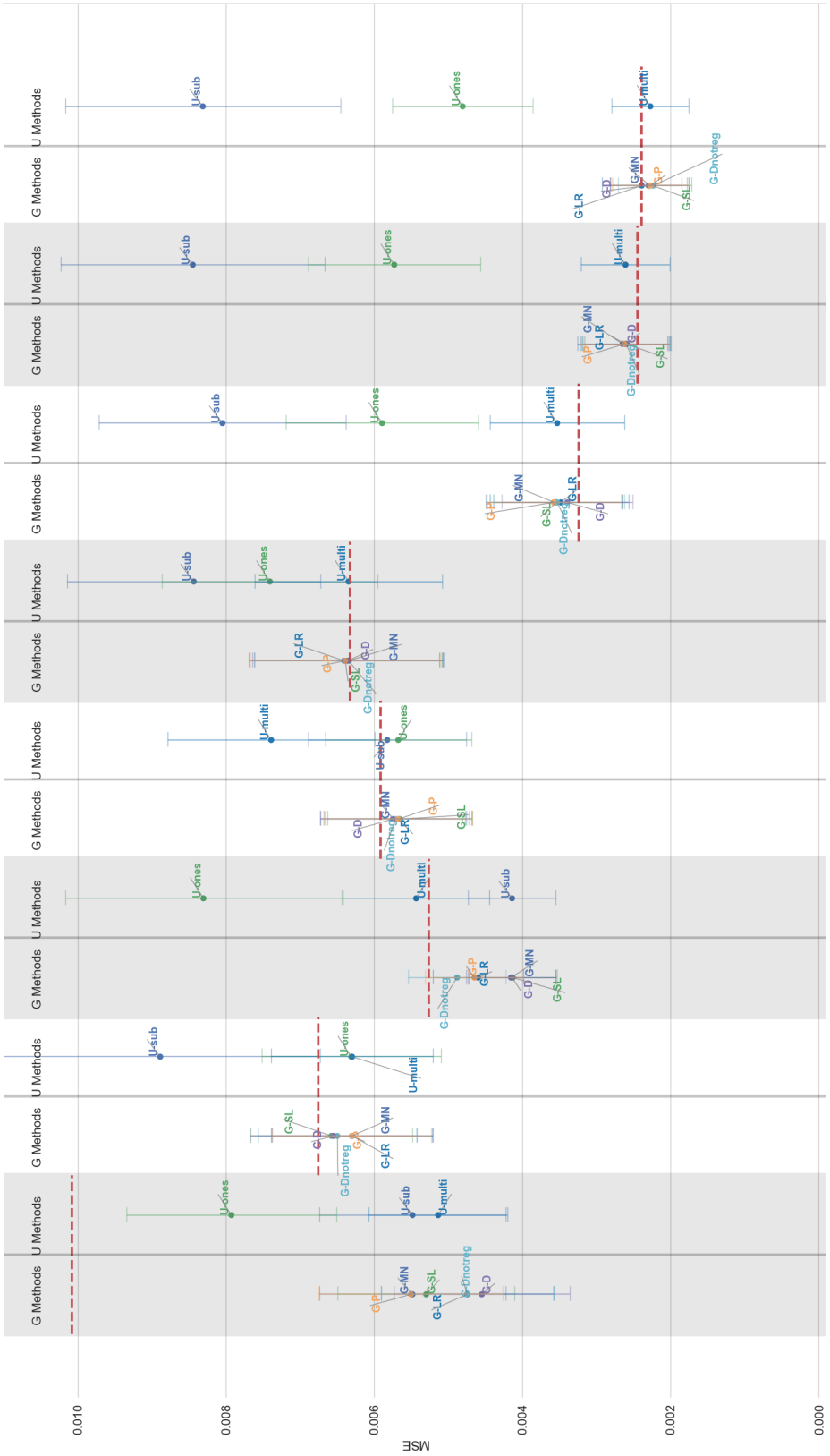

Figure 20: LF (v1) $n = 500$ results. For each outcome model Q (x-axis) we plot the corresponding Mean Squared Error (y-axis) for each of the possible propensity models G (left sub-column) and each of the possible update methods U. The base performance (no update step and therefore no G or U) is given as a horizontal dashed red line. Because we undertook all combinations of G and U, each point represents a marginalization over the other dimension. For instance, for Q-D (DragonNet), the 'U-ones' point is an average result for the onestep update process, using all possible propensity models G. Graph best viewed in color.

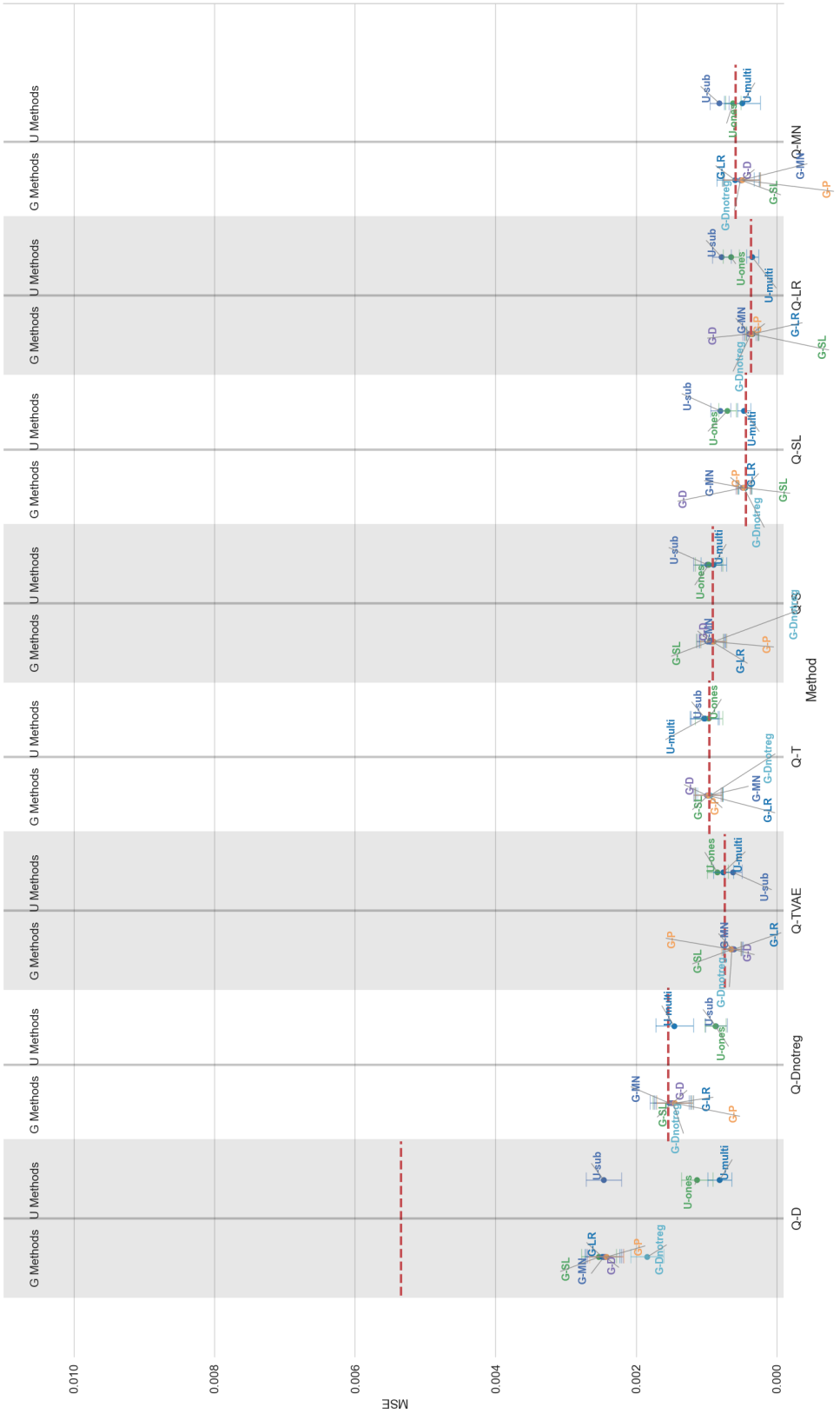

Figure 21: LF (v1) $n = 5000$ results. For each outcome model Q (x-axis) we plot the corresponding Mean Squared Error (y-axis) for each of the possible propensity models G (left sub-column) and each of the possible update methods U. The base performance (no update step and therefore no G or U) is given as a horizontal dashed red line. Because we undertook all combinations of G and U, each point represents a marginalization over the other dimension. For instance, for Q-D (DragonNet), the 'U-ones' point is an average result for the onestep update process, using all possible propensity models G. Graph best viewed in color.

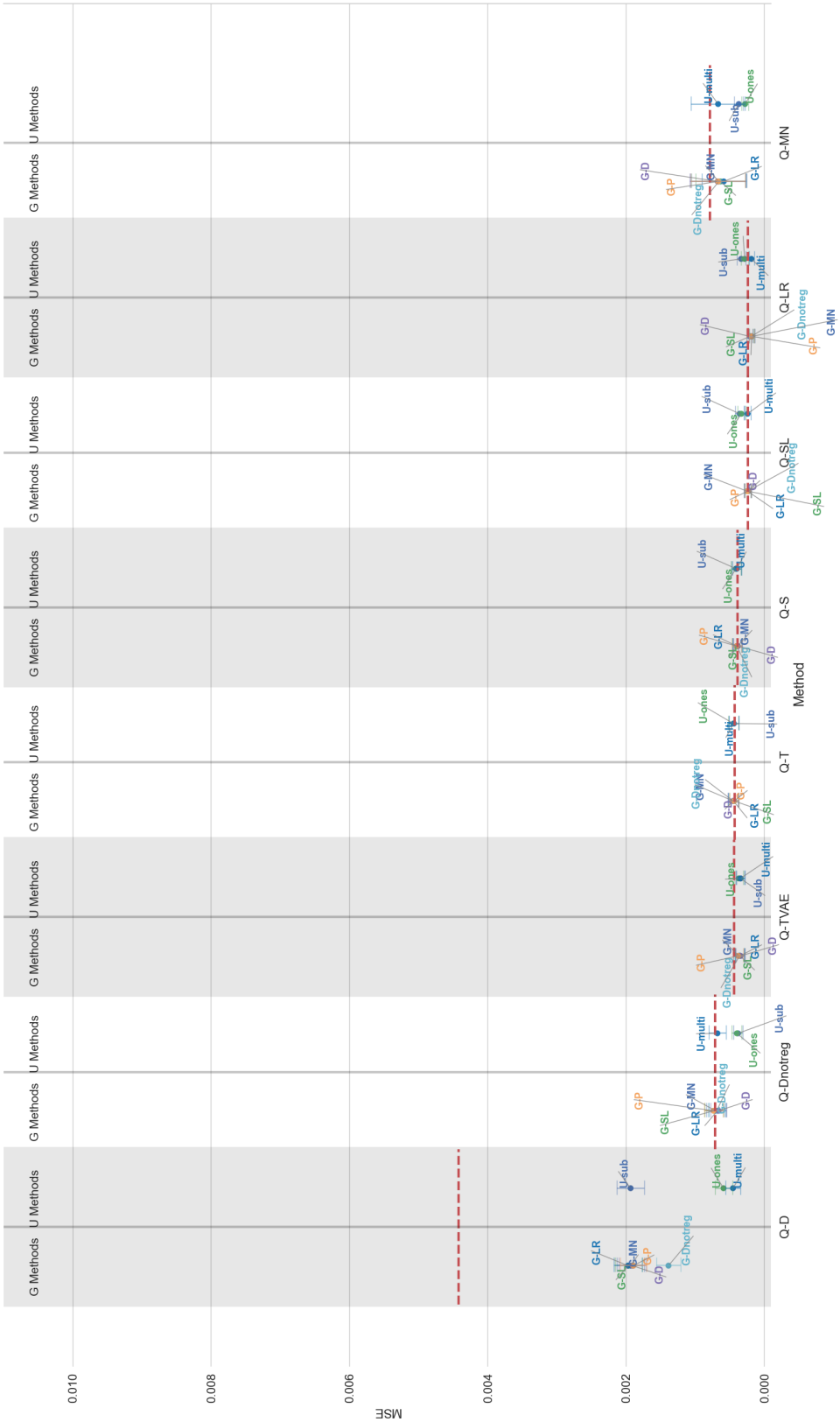

Figure 22: LF (v1) $n = 10000$ results. For each outcome model Q (x-axis) we plot the corresponding Mean Squared Error (y-axis) for each of the possible propensity models G (left sub-column) and each of the possible update methods U. The base performance (no update step and therefore no G or U) is given as a horizontal dashed red line. Because we undertook all combinations of G and U, each point represents a marginalization over the other dimension. For instance, for Q-D (DragonNet), the 'U-ones' point is an average result for the onestep update process, using all possible propensity models G. Graph best viewed in color.

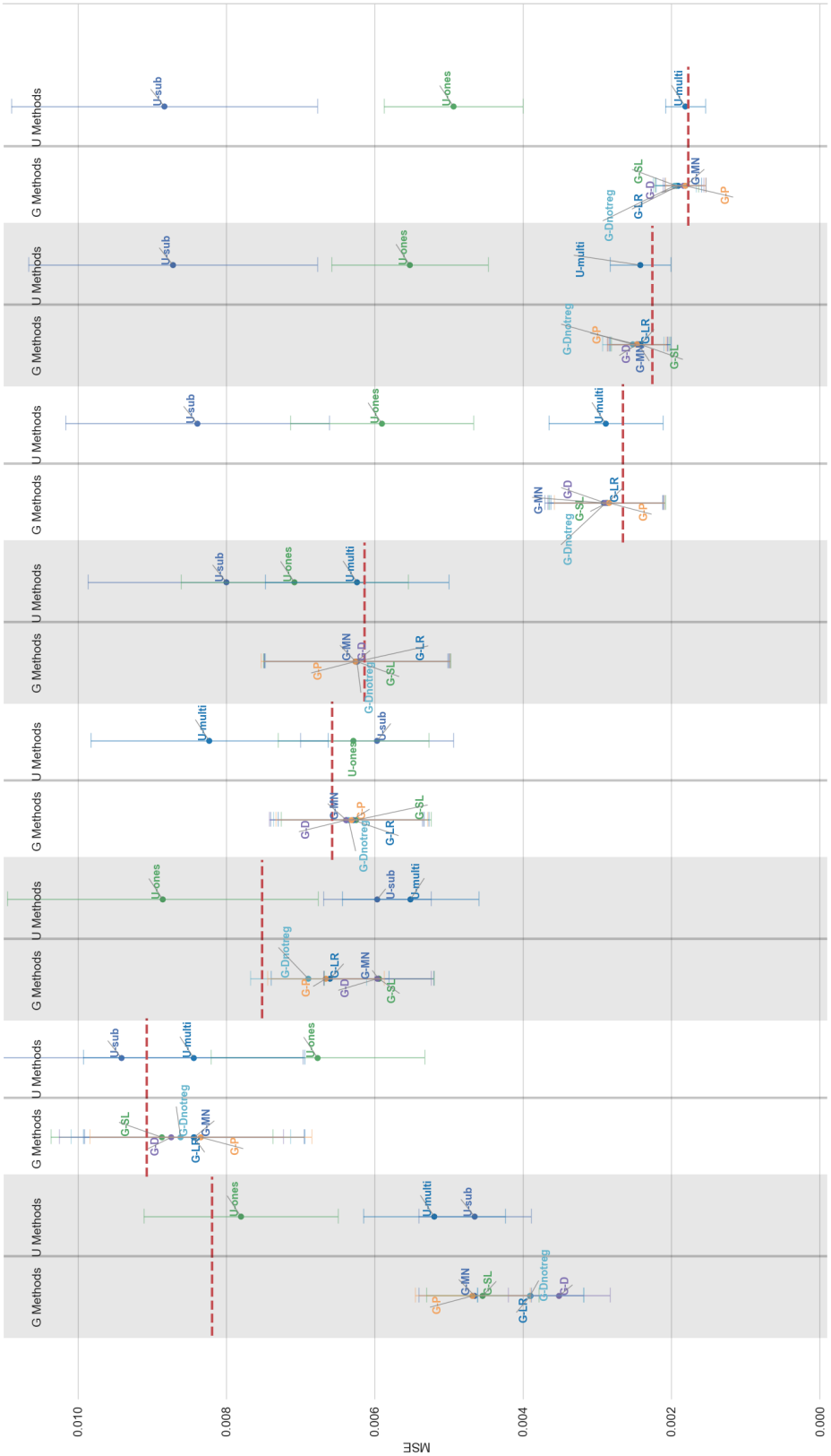

Figure 23: LF (v2) $n = 500$ results. For each outcome model Q (x-axis) we plot the corresponding Mean Squared Error (y-axis) for each of the possible propensity models G (left sub-column) and each of the possible update methods U. The base performance (no update step and therefore no G or U) is given as a horizontal dashed red line. Because we undertook all combinations of G and U, each point represents a marginalization over the other dimension. For instance, for Q-D (DragonNet), the 'U-ones' point is an average result for the onestep update process, using all possible propensity models G. Best viewed in color.

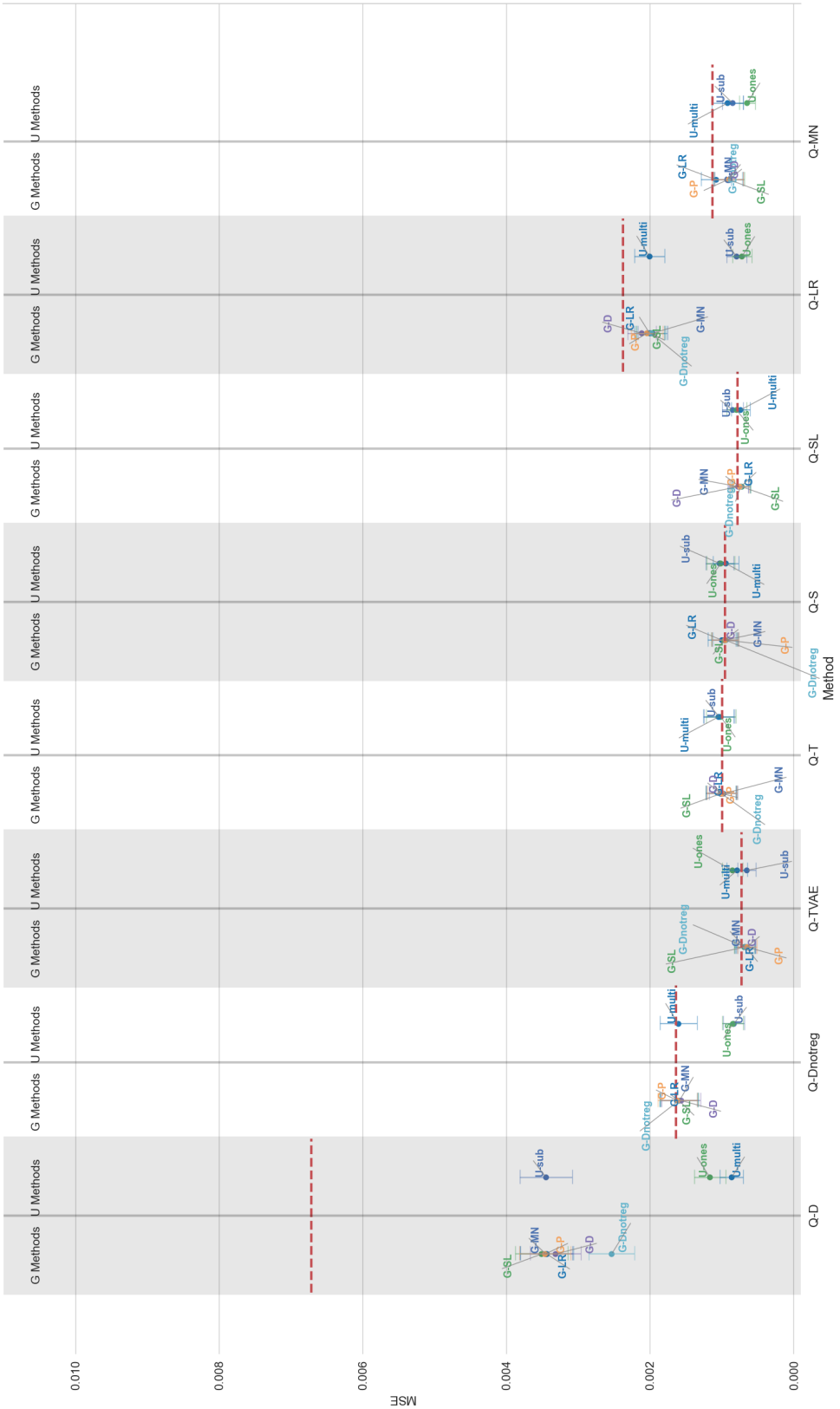

Figure 24: LF (v2) $n = 5000$ results. For each outcome model Q (x-axis) we plot the corresponding Mean Squared Error (y-axis) for each of the possible propensity models G (left sub-column) and each of the possible update methods U. The base performance (no update step and therefore no G or U) is given as a horizontal dashed red line. Because we undertook all combinations of G and U, each point represents a marginalization over the other dimension. For instance, for Q-D (DragonNet), the 'U-ones' point is an average result for the onestep update process, using all possible propensity models G. Graph best viewed in color.

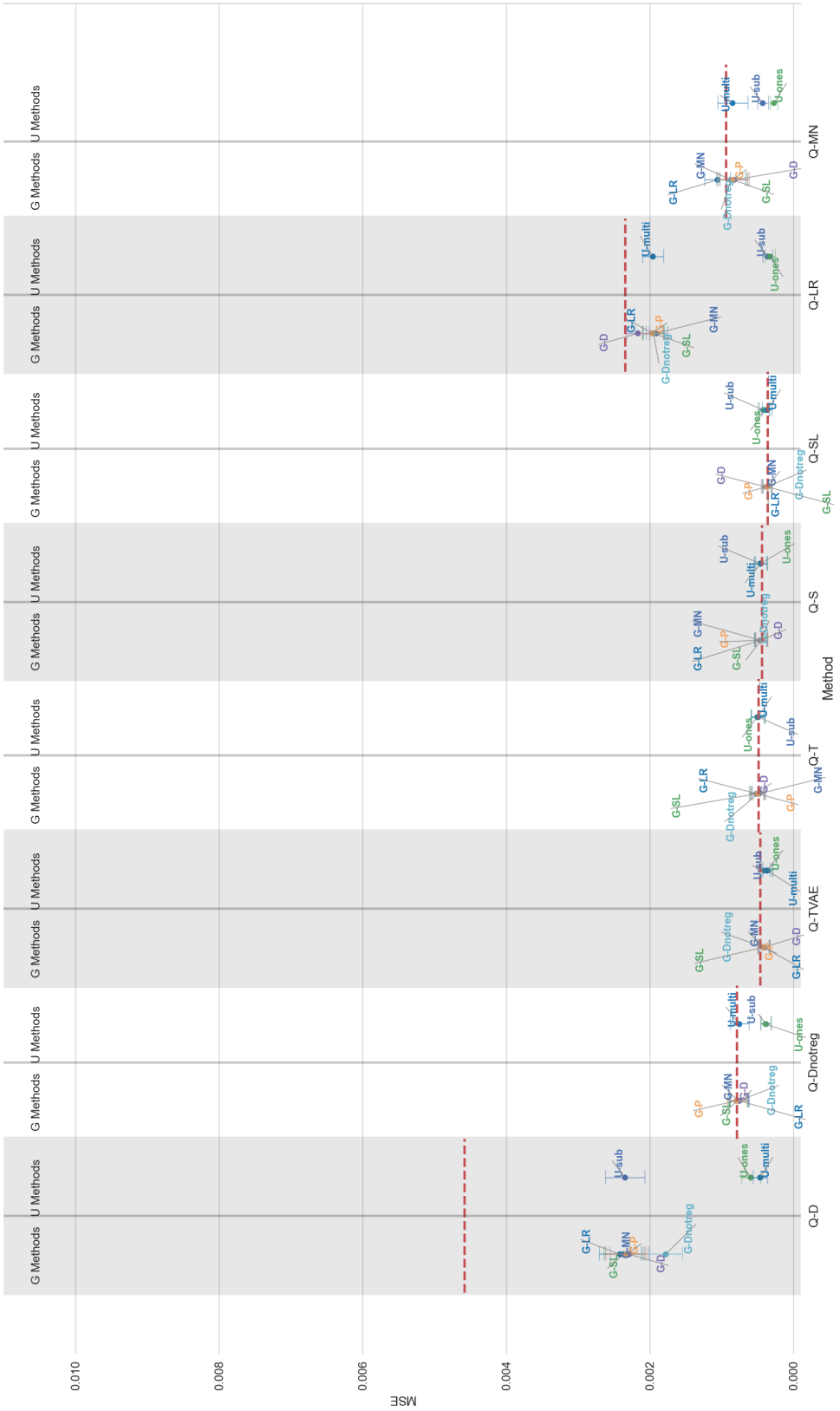

Figure 25: LF (v2) $n = 10000$ results. For each outcome model Q (x-axis) we plot the corresponding Mean Squared Error (y-axis) for each of the possible propensity models G (left sub-column) and each of the possible update methods U. The base performance (no update step and therefore no G or U) is given as a horizontal dashed red line. Because we undertook all combinations of G and U, each point represents a marginalization over the other dimension. For instance, for Q-D (DragonNet), the 'U-ones' point is an average result for the onestep update process, using all possible propensity models G. Best viewed in color.

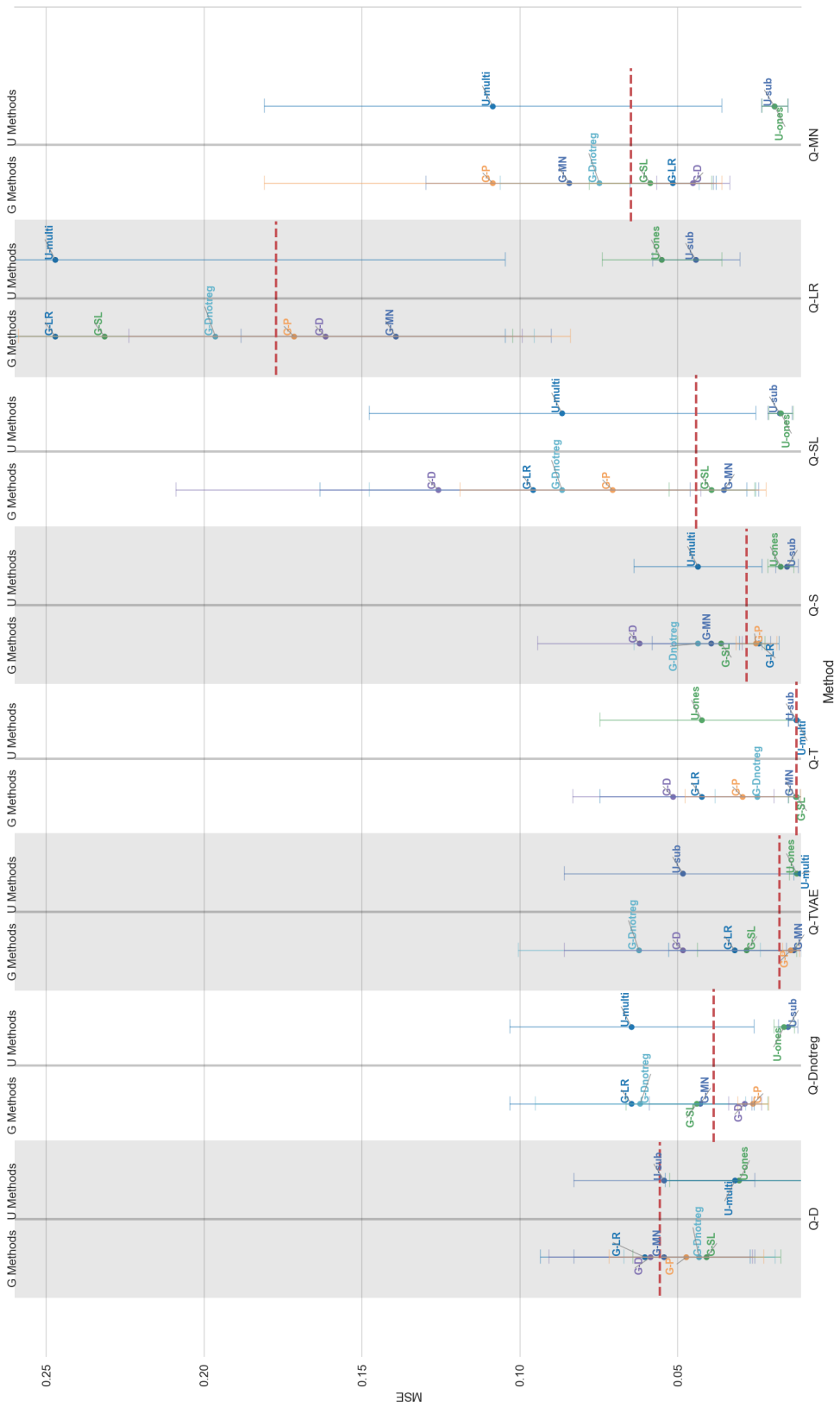

Figure 26: IHDP $n = 747$ results. For each outcome model Q (x-axis) we plot the corresponding Mean Squared Error (y-axis) for each of the possible propensity models G (left sub-column) and each of the possible update methods U. The base performance (no update step and therefore no G or U) is given as a horizontal dashed red line. Because we undertook all combinations of G and U, each point represents a marginalization over the other dimension. For instance, for Q-D (DragonNet), the 'U-ones' point is an average result for the onestep update process, using all possible propensity models G. Best viewed in colour.

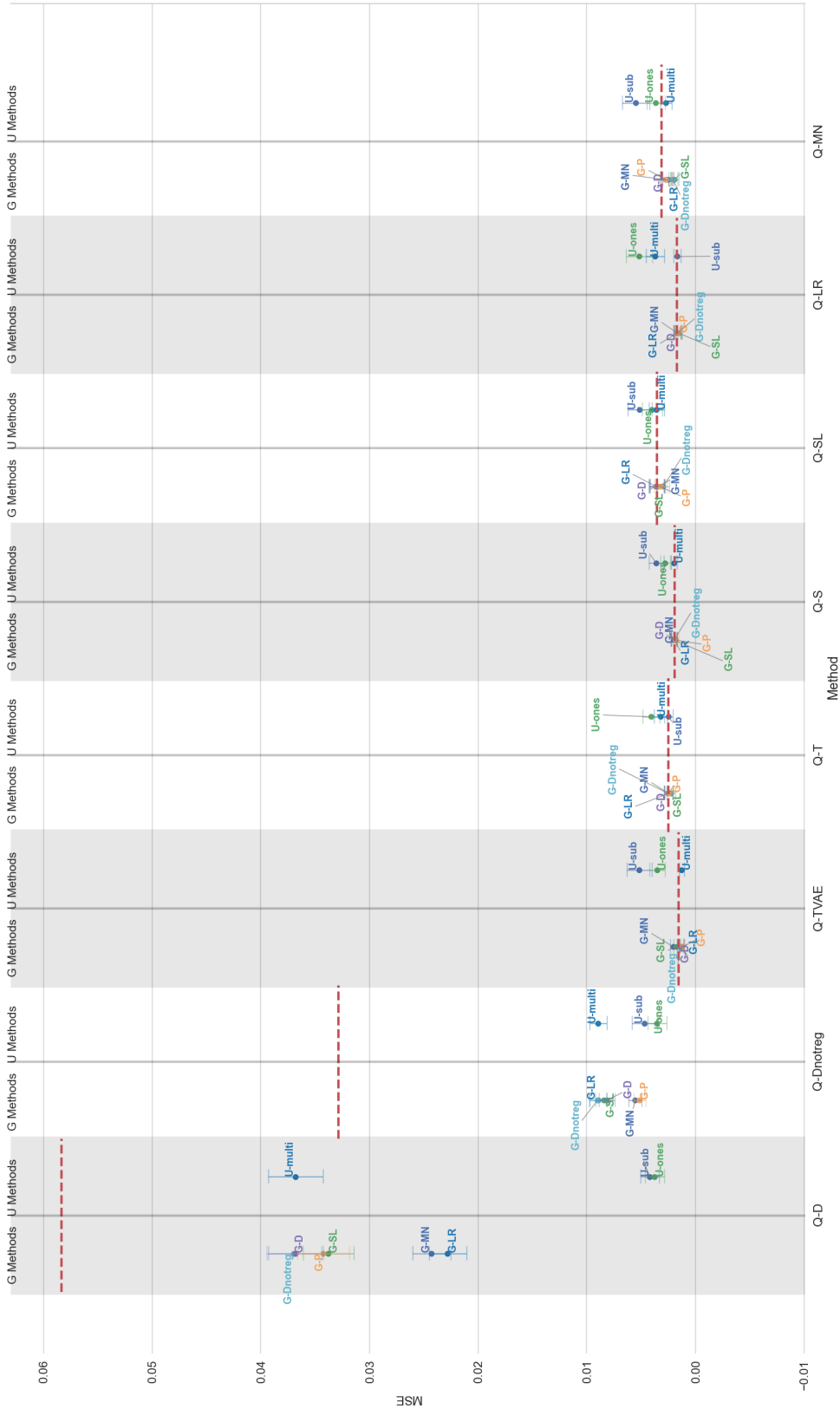

Figure 27: General graph $n = 500$ results. For each outcome model Q (x-axis) we plot the corresponding Mean Squared Error (y-axis) for each of the possible propensity models G (left sub-column) and each of the possible update methods U. The base performance (no update step and therefore no G or U) is given as a horizontal dashed red line. Because we undertook all combinations of G and U, each point represents a marginalization over the other dimension. For instance, for Q-D (DragonNet), the 'U-ones' point is an average result for the onestep update process, using all possible propensity models G. Best viewed in color.

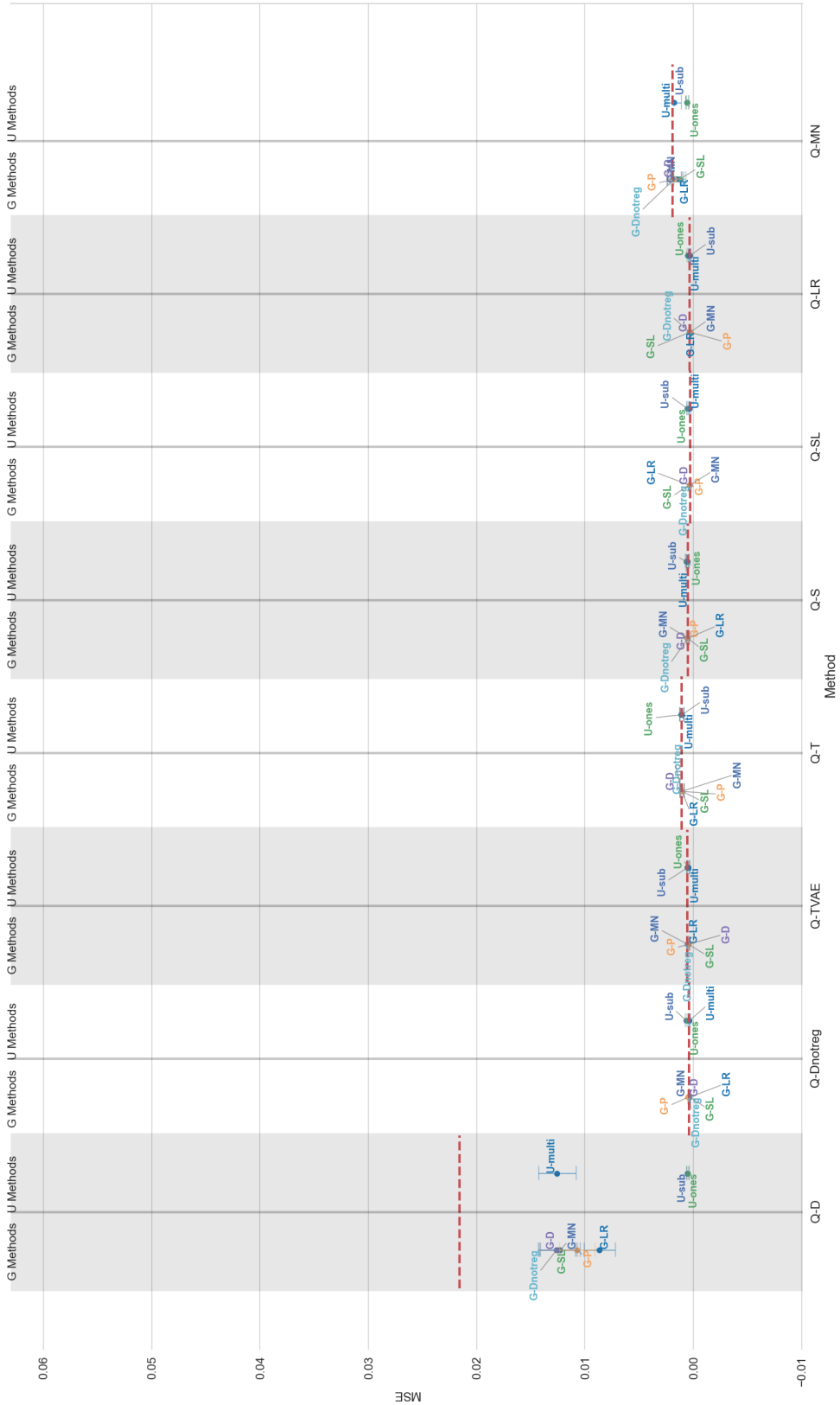

Figure 28: General graph $n = 5000$ results. For each outcome model Q (x-axis) we plot the corresponding Mean Squared Error (y-axis) for each of the possible propensity models G (left sub-column) and each of the possible update methods U. The base performance (no update step and therefore no G or U) is given as a horizontal dashed red line. Because we undertook all combinations of G and U, each point represents a marginalization over the other dimension. For instance, for Q-D (DragonNet), the 'U-ones' point is an average result for the onestep update process, using all possible propensity models G. Graph best viewed in color.

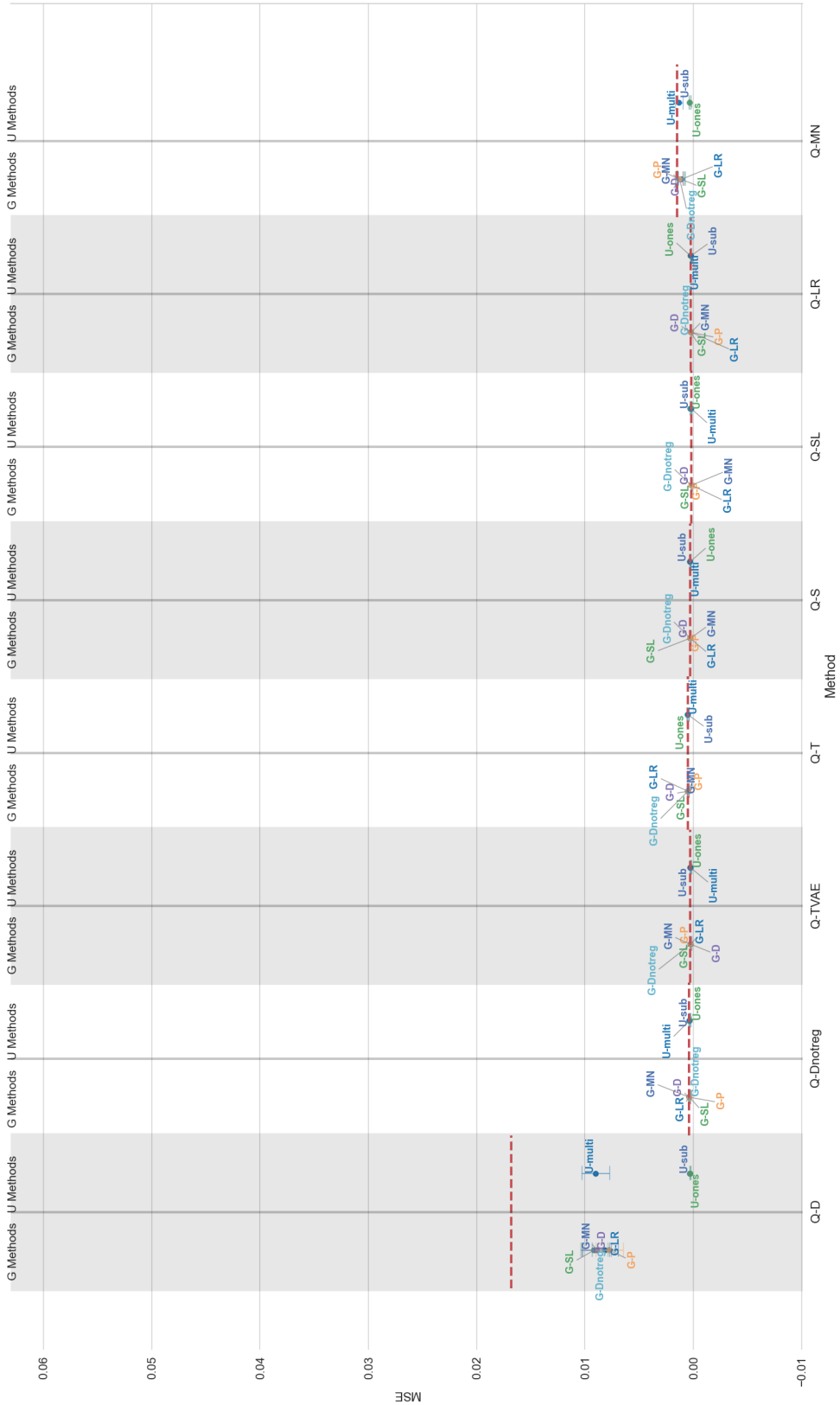

Figure 29: General graph $n = 10000$ results. For each outcome model Q (x-axis) we plot the corresponding Mean Squared Error (y-axis) for each of the possible propensity models G (left sub-column) and each of the possible update methods U. The base performance (no update step and therefore no G or U) is given as a horizontal dashed red line. Because we undertook all combinations of G and U, each point represents a marginalization over the other dimension. For instance, for Q-D (DragonNet), the 'U-ones' point is an average result for the onestep update process, using all possible propensity models G. Best viewed in color.