Deep Neural Networks for Estimation and Inference[*]

Max H. Farrell   Tengyuan Liang   Sanjog Misra

University of Chicago, Booth School of Business

September 19, 2019

Abstract

We study deep neural networks and their use in semiparametric inference. We establish novel rates of convergence for deep feedforward neural nets. Our new rates are sufficiently fast (in some cases minimax optimal) to allow us to establish valid second-step inference after first-step estimation with deep learning, a result also new to the literature. Our estimation rates and semiparametric inference results handle the current standard architecture: fully connected feedforward neural networks (multi-layer perceptrons), with the now-common rectified linear unit activation function and a depth explicitly diverging with the sample size. We discuss other architectures as well, including fixed-width, very deep networks. We establish nonasymptotic bounds for these deep nets for a general class of nonparametric regression-type loss functions, which includes as special cases least squares, logistic regression, and other generalized linear models. We then apply our theory to develop semiparametric inference, focusing on causal parameters for concreteness, such as treatment effects, expected welfare, and decomposition effects. Inference in many other semiparametric contexts can be readily obtained. We demonstrate the effectiveness of deep learning with a Monte Carlo analysis and an empirical application to direct mail marketing.

Keywords: Deep Learning, Neural Networks, Rectified Linear Unit, Nonasymptotic Bounds, Convergence Rates, Semiparametric Inference, Treatment Effects, Program Evaluation, Treatment Targeting.

1 Introduction

Statistical machine learning methods are being rapidly integrated into the social and medical sciences. Economics is no exception, and there has been a recent surge of research that applies and explores machine learning methods in the context of econometric modeling, particularly in "big data" settings. Furthermore, theoretical properties of these methods are the subject of intense recent study. This has netted several breakthroughs both theoretically, such as robust, valid inference following machine learning, and in novel applications and conclusions. Our goal in the present work

is to study a particular statistical machine learning technique which is widely popular in industrial applications, but less frequently used in academic work and largely ignored in recent theoretical developments on inference: deep neural networks. To our knowledge we provide the first inference results using deep learning methods.

Neural networks are estimation methods that model the relationship between inputs and outputs using layers of connected computational units (neurons), patterned after the biological neural networks of brains. These computational units sit between the inputs and output and allow data-driven learning of the appropriate model, in addition to learning the parameters of that model. Put into terms more familiar in nonparametric econometrics: neural networks can be thought of as a (complex) type of sieve estimation where the basis functions are flexibly learned from the data. Neural networks are perhaps not as familiar to economists as other methods, and indeed, were out of favor in the machine learning community for several years, returning to prominence only very recently in the form of deep learning. Deep neural nets contain many hidden layers of neurons between the input and output layers, and have been found to exhibit superior performance across a variety of contexts. Our work aims to bring wider attention to these methods and to take the first step toward filling the gaps in the theoretical understanding of inference using deep neural networks. Our results can be used in many economic contexts, including selection models, games, consumer surplus, and dynamic discrete choice.

Before the recent surge in attention, neural networks had taken a back seat to other methods (such as kernel methods or forests) largely because of their modest empirical performance and challenging optimization. However, the availability of scalable computing and stochastic optimization techniques (LeCun et al., 1998; Kingma and Ba, 2014) and the change from smooth sigmoid-type activation functions to rectified linear units (ReLU), $x \mapsto \max(x, 0)$ (Nair and Hinton, 2010), have seemingly overcome optimization hurdles and now this form of deep learning matches or sets the state of the art in many prediction contexts (Krizhevsky et al., 2012; He et al., 2016). Our theoretical results speak directly to this modern implementation of deep learning: we explicitly model the depth of the network as diverging with the sample size and focus on the ReLU activation function.

Further back in history, before falling out of favor, neural networks were widely studied and applied, particularly in the 1990s. In that time, *shallow* neural networks with smooth activation functions were shown to have many good theoretical properties. Intuitively, neural networks are

1

a form of sieve estimation, wherein basis functions of the original variables are used to approximate unknown nonparametric objects. What sets neural nets apart is that the basis functions are themselves learned from the data by optimizing over many flexible combinations of simple functions. It has been known for some time that such networks yield universal approximations (Hornik et al., 1989). Comprehensive theoretical treatments are given by White (1992) and Anthony and Bartlett (1999). Of particular relevance in this strand of theoretical work is Chen and White (1999), where it was shown that single-layer, sigmoid-based networks could attain sufficiently fast rates for semiparametric inference (see Chen (2007) for more references).

We explicitly depart from the extant literature by focusing on the modern setting of deep neural networks with the rectified linear (ReLU) activation function. We provide nonasymptotic bounds for nonparametric estimation using deep neural networks, immediately implying convergence rates. The bounds and convergence rates appear to be new to the literature and are one of the main theoretical contributions of the paper. We provide results for a general class of smooth loss functions for nonparametric regression style problems, covering as special cases generalized linear models and other empirically useful contexts. In our application to causal inference we specialize our results to linear and logistic regression as concrete illustrations. Our proof strategy employs a localization analysis that uses scale-insensitive measures of complexity, allowing us to consider richer classes of neural networks. This is in contrast to analyses which restrict the networks to have bounded parameters for each unit (discussed more below) and to the application of scale sensitive measures such as metric entropy (used by Chen and White, 1999, for example). These approaches would not deliver our sharp bounds and fast rates. Recent developments in approximation theory and complexity for deep ReLU networks are important building blocks for our results.

Our second main result establishes valid inference on finite-dimensional parameters following first-step estimation using deep learning. We focus on causal inference for concreteness and wide applicability, as well as to allow direct comparison to the literature. Program evaluation with observational data is one of the most common and important inference problems, and has often been used as a test case for theoretical study of inference following machine learning (e.g., Belloni et al., 2014; Farrell, 2015; Belloni et al., 2017; Athey et al., 2018). Causal inference as a whole is a vast literature; see Imbens and Rubin (2015) for a broad review and Abadie and Cattaneo (2018) for a recent review of program evaluation methods, and further references in both. Deep neural

networks have been argued (experimentally) to outperform the previous state-of-the-art in causal inference (Westreich et al., 2010; Johansson et al., 2016; Shalit et al., 2017; Hartford et al., 2017). To the best of our knowledge, ours are among the first theoretical results that explicitly deliver inference using deep neural networks.

We give specific results for average treatment effects, counterfactual expected utility/profits from treatment targeting strategies, and decomposition effects. Our results allow planners (e.g., firms or medical providers) to compare different strategies, either predetermined or estimated using auxiliary data, and recognizing that targeting can be costly, decide which strategy to implement. An interesting, and potentially useful, point we make in this context is that the selection on observables framework yields identification of counterfactual average outcomes without additional structural assumptions, so that, e.g., expected profit from a counterfactual treatment rule can be evaluated.

The usefulness of our deep learning results is of course not limited to causal inference. In particular, our results yield inference on essentially any estimand that admits a locally robust estimator (Chernozhukov et al., 2018c) that depends only on target functions within our class of loss function (under appropriate regularity conditions). Our aim is not to innovate at the semiparametric step, for example by seeking weaker conditions on the first stage, but rather, we aim to utilize such results. Prior work has verified the high-level conditions for other first-stage estimators, such as traditional kernels or series/sieves, lasso methods, sigmoid-based shallow neural networks, and others (under suitable assumptions for each method). Our work contributes directly to this area of research by showing that deep nets are a valid and useful first-step estimator, in particular, attaining a rate of $o(n^{-1/4})$ under appropriate smoothness conditions. Finally, we do not rely on sample splitting or cross fitting. In particular, we use localization explicitly to directly verify conditions required for valid inference, which may be a novel application of this proof method that is useful in future semiparametric inference problems.

We numerically illustrate our results, and more generally the utility of deep learning, with a detailed simulation study and an empirical study of a direct mail marketing campaign. Our data come from a large US consumer products retailer and consists around to three hundred thousand consumers with one hundred fifty covariates. Hitsch and Misra (2018) recently used this data to study various estimators, both traditional and modern, of heterogeneous treatment effects. We refer the reader to that paper for a more complete description of the data as well as results using other

estimators (see also Hansen et al. (2017)). We study the effect of catalog mailings on consumer purchases, and moreover, compare different targeting strategies (i.e. to which consumers catalogs should be mailed). The cost of sending out a single catalog can be close to one dollar, and with millions being set out, carefully assessing the targeting strategy is crucial. Our results suggest that deep nets are at least as good as (and sometimes better than) the best methods in Hitsch and Misra (2018).

The remainder of the paper proceeds as follows. Next, we briefly review the related theoretical literature. Section 2 introduces deep ReLU networks and states our main theoretical results: nonasymptotic bounds and convergence rates for general nonparametric regression-type loss functions. The semiparametric inference problem is set up in Section 3 and asymptotic results are presented in Section 4. The empirical application is presented in Section 5. Results of a simulation study are reported in Section 6. Section 7 concludes. All proofs are given in the appendix.

1.1 Related Theoretical Literature

Our paper contributes to several rapidly growing literatures, and we can not hope to do justice to each here. We give only those citations of particular relevance; more references can be found within these works. First, there has been much recent study of the statistical properties of the machine learning tools as an end in itself. Many studies have focused on the lasso and its variants (Bickel et al., 2009; Belloni et al., 2011, 2012; Farrell, 2015) and tree/forest based methods (Wager and Athey, 2018), though earlier work studied shallow (typically with a single hidden layer) neural networks with smooth activation functions (White, 1989, 1992; Chen and White, 1999). We fill the gap in this literature by studying *deep* neural networks with the non-smooth ReLU activation.

A second, intertwined strand of literature focuses on inference following the use of machine learning methods, often with a focus on average causal effects. Initial theoretical results were concerned with obtaining valid inference on a coefficient in a high-dimensional regression, following model selection or regularization, with particular focus on the lasso (Belloni et al., 2012; Javanmard and Montanari, 2014; van de Geer et al., 2014). Intuitively, this is a semiparametric problem, where the coefficient of interest is estimable at the parametric rate and the remaining coefficients are collectively a nonparametric nuisance parameter estimated using machine learning methods. Building on this intuition, many have studied the semiparametric stage directly, such as obtaining

novel, weaker conditions easing the application of machine learning methods (Belloni et al., 2014; Farrell, 2015; Chernozhukov et al., 2018c; Belloni et al., 2018, and references therein). Conceptually related to this strand are targeted maximum likelihood (van der Laan and Rose, 2001) and the higher-order influence functions (Robins et al., 2008, 2017). Our work builds on this work, employing conditions therein, and in particular, verifying them for deep ReLU nets.

Finally, our convergence rates build on, and contribute to, the recent theoretical machine learning literature on deep neural networks. Because of the renaissance in deep learning, a considerable amount of study has been done in recent years. Of particular relevance to us are Yarotsky (2017, 2018) and Bartlett et al. (2017); a recent textbook treatment, containing numerous other references, is given by Goodfellow et al. (2016).

2   Deep Neural Networks

In this section we will give our main theoretical results: nonasymptotic bounds and associated convergence rates for deep neural network estimation. The utility of these results for second-step semiparametric causal inference (the downstream task), for which our rates are sufficiently rapid, is demonstrated in Section 4. We view our results as an initial step in establishing both the estimation and inference theory for modern deep learning, i.e. neural networks built using the multi-layer perceptron architecture (described below) and the nonsmooth ReLU activation function. This combination is crucial: it has demonstrated state of the art performance empirically and can be feasibly optimized. This is in contrast with sigmoid-based networks, either shallow (for which theory exists, but may not match empirical performance) or deep (which are not feasible to optimize), and with shallow ReLU networks, which are not known to approximate broad classes functions.

As neural networks are perhaps less familiar to economists and other social scientists, we first briefly review the construction of deep ReLU nets. Our main focus will be on the fully connected feedfoward neural network, frequently referred to as a multi-layer perceptron, as this is the most commonly implemented network architecture and we want our results to inform empirical practice. However, our results are more general, accommodating other architectures provided they are able to yield a universal approximation (in the appropriate function class), and so we review neural nets

more generally and give concrete examples.

Our goal is to estimate an unknown, assumed-smooth function $f_*(\boldsymbol{x})$, that relates the covariates $\boldsymbol{X} \in \mathbb{R}^d$ to an outcome Y as the minimizer of the expectation of the per-observation loss function. Collecting these random variables into the vector $\boldsymbol{Z} = (Y, \boldsymbol{X}')' \in \mathbb{R}^{d+1}$, with $\boldsymbol{z} = (y, \boldsymbol{x}')'$ denoting a realization, we write

$$f_* = \arg\min \mathbb{E}\left[\ell\left(f, \boldsymbol{Z}\right)\right].$$

We allow for any loss function that is Lipschitz in f and obeys a curvature condition around f_*. Specifically, for constants c_1, c_2, and C_ℓ that are bounded and bounded away from zero, we assume that $\ell(f, \boldsymbol{z})$ obeys

$$|\ell(f, \boldsymbol{z}) - \ell(g, \boldsymbol{z})| \le C_\ell |f(\boldsymbol{x}) - g(\boldsymbol{x})|,$$

$$c_1 \mathbb{E}\left[(f - f_*)^2\right] \le \mathbb{E}[\ell(f, \boldsymbol{Z})] - \mathbb{E}[\ell(f_*, \boldsymbol{Z})] \le c_2 \mathbb{E}\left[(f - f_*)^2\right]. \tag{2.1}$$

Our results will be stated for a general loss obeying these two conditions.[1] We give a unified localization analysis of all such problems. This family of loss function covers many interesting problems. Two leading examples, used in our application to causal inference, are least squares and logistic regression, corresponding to the outcome and propensity score models respectively. For least squares, the target function and loss are

$$f_*(\boldsymbol{x}) := \mathbb{E}[Y|\boldsymbol{X} = \boldsymbol{x}] \qquad \text{and} \qquad \ell\left(f, \boldsymbol{z}\right) = \frac{1}{2}(y - f(\boldsymbol{x}))^2, \tag{2.2}$$

respectively, while for logistic regression these are

$$f_*(\boldsymbol{x}) := \log \frac{\mathbb{E}[Y|\boldsymbol{X} = \boldsymbol{x}]}{1 - \mathbb{E}[Y|\boldsymbol{X} = \boldsymbol{x}]} \qquad \text{and} \qquad \ell\left(f, \boldsymbol{z}\right) = -yf(\boldsymbol{x}) + \log\left(1 + e^{f(\boldsymbol{x})}\right). \tag{2.3}$$

Lemma 8 verifies, with explicit constants, that (2.1) holds for these two. Losses obeying (2.1) extend beyond these cases to other generalized linear models, such as count models, and can even cover multinomial logistic regression (multiclass classification), as shown in Lemma 9.

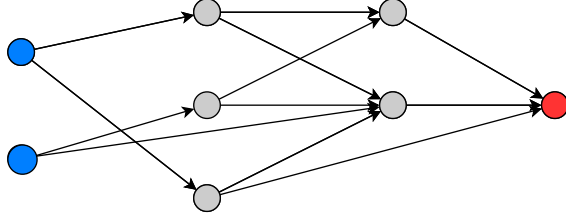[1] We thank an anonymous referee for suggesting this approach.

Figure 1: Illustration of a feedforward neural network with $W = 18$, $L = 2$, $U = 5$, and input dimension $d = 2$. The input units are shown in blue at left, the output in red at right, and the hidden units in grey between them.

2.1 Neural Network Constructions

For any loss, we estimate the target function using a deep ReLU network. We will give a brief outline of their construction here, paying closer attention to the details germane to our theory; complete introductions, and further references, are given by Anthony and Bartlett (1999) and Goodfellow et al. (2016).

The crucial choice is the specific network architecture, or class. In general we will call this $\mathcal{F}_{\mathrm{DNN}}$. From a theoretical point of view, different classes have different complexity and different approximating power. We give results for several concrete examples below. We will focus on *feedforward neural networks*. An example of a feedforward network is shown in Figure 1. The network consists of d input units, corresponding to the covariates $\boldsymbol{X} \in \mathbb{R}^d$, one output unit for the outcome Y. Between these are U hidden units, or computational nodes or neurons. These are connected by a directed acyclic graph specifying the architecture. The key graphical feature of a feedforward network is that hidden units are grouped in a sequence of L layers, the *depth* of the network, where a node is in layer $l = 1, 2, \ldots, L$, if it has a predecessor in layer $l - 1$ and no predecessor in any layer $l' \geq l$. The *width* of the network at a given layer, denoted H_l, is the number of units in that layer. The network is completed with the choice of an *activation function* $\sigma : \mathbb{R} \mapsto \mathbb{R}$ applied to the output of each node as described below. In this paper, we focus on the popular ReLU activation function $\sigma(x) = \max(x, 0)$, though our results can be extended (at notational cost) to cover piecewise linear activation functions (see also Remark 3).

An important and widely used subclass is the one that is *fully connected* between consecutive layers but has *no* other connections and each layer has number of hidden units that are of the same order of magnitude. This architecture is often referred to as a *Multi-Layer Perceptron* (MLP) and
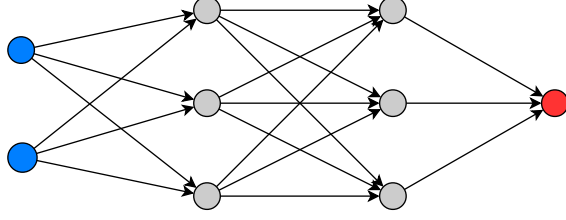
Figure 2: Illustration of multi-layer perceptron $\mathcal{F}_{\mathrm{MLP}}$ with $H = 3$, $L = 2$ ($U = 6$, $W = 25$), and input dimension $d = 2$.

we denote this class as $\mathcal{F}_{\mathrm{MLP}}$. See Figure 2, cf. Figure 1. We will assume that all the width of all layers share a common asymptotic order H, implying that for this class $U \asymp LH$.

We will allow for generic feedforward networks in our results, but we present special results for the MLP case, as it is widely used in empirical practice. As we will see below, the architecture, through its complexity, and more importantly, approximation power, plays a crucial role in the final convergence rate. In particular, we find only a suboptimal rate for the MLP case, but our upper bound is still sufficient for semiparametric inference. As a note on exposition, while our main results are in fact nonasymptotic bounds that hold with high probability, for simplicity we will refer to them as "rates" in most discussion.

To build intuition on the computation, and compare to other nonparametric methods, let us focus on least squares for the moment, i.e. Equation (2.2), with a continuous outcome using a multilayer perceptron with constant width H. Each hidden unit u receives an input in the form of a linear combination $\tilde{\boldsymbol{x}}'\boldsymbol{w} + b$, and then returns $\sigma(\tilde{\boldsymbol{x}}'\boldsymbol{w} + b)$, where the vector $\tilde{\boldsymbol{x}}$ collects the output of all the units with a directed edge into u (i.e., from prior layers), $\boldsymbol{w}$ is a vector of weights, and b is a constant term. (The constant term is often referred to as the "bias" in the deep learning literature, but given the loaded meaning of this term in inference, we will largely avoid referring to b as a bias.) The final layer's output is simply $\tilde{\boldsymbol{x}}'\boldsymbol{w} + b$ in the least squares case. The collection, over all nodes, of $\boldsymbol{w}$ and b, constitutes the parameters θ which are optimized in the final estimation. We denote W as the total number of parameters of the network. For the MLP, $W = (d+1)H + (L-1)(H^2 + H) + H + 1$. In general, W, U, L, and H, may change with n, but we suppress this in the notation.

Optimization proceeds layer-by-layer using (variants of) stochastic gradient descent, with gradients of the parameters calculated by back-propagation (implementing the chain rule) induced by the network structure. To see this, let $\tilde{x}_{h,l}$ denote the scalar output of a node $u = (h, l)$, for

$h = 1, \ldots H$, $l = 1, \ldots L$, and let $\tilde{\boldsymbol{x}}_l = (\tilde{x}_{1,l}, \ldots, \tilde{x}_{H,l})'$ for layer $l \leq L$. Each node thus computes $\tilde{x}_{h,l} = \sigma(\tilde{\boldsymbol{x}}'_{l-1} \boldsymbol{w}_{h,l-1} + b_{h,l-1})$ and the final output is $\hat{y} = \tilde{\boldsymbol{x}}'_L \boldsymbol{w}_L + b_L$. Once we recall that the network begins with the original observation $\boldsymbol{x}$, we can view $\tilde{\boldsymbol{x}}_L = \tilde{\boldsymbol{x}}_L(\boldsymbol{x})$, and thus the final output may be seen as a basis function approximation (albeit a complex and random one) written as $\hat{f}_{\mathrm{MLP}}(\boldsymbol{x}) = \tilde{\boldsymbol{x}}_L(\boldsymbol{x})' \boldsymbol{w}_L + b_L$, which is reminiscent of a traditional series (linear sieve) estimator. If all layers save the last were fixed, we could simply optimize using least squares directly: $(\boldsymbol{w}_L, b_L) = \arg\min_{\boldsymbol{w}, b} \|y_i - \tilde{\boldsymbol{x}}'_L \boldsymbol{w} - b\|_n^2$.

The crucial distinction is that the basis functions $\tilde{\boldsymbol{x}}_L(\cdot)$ are learned from the data. The "basis" is $\tilde{\boldsymbol{x}}_L = (\tilde{x}_{1,L}, \ldots, \tilde{x}_{H,L})'$, where each $\tilde{x}_{h,L} = \sigma(\tilde{\boldsymbol{x}}'_{L-1} \boldsymbol{w}_{h,L-1} + b_{h,L-1})$. Therefore, "before" we can solve the least squares problem above, we would have to estimate $(\boldsymbol{w}'_{h,L-1}, b_{h,L-1})$, $h = 1, \ldots, H$, anticipating the final estimation. These in turn depend on the prior layer, and so forth back to the original inputs $\boldsymbol{X}$. Measuring the gradient of the loss with respect to each layer of parameters uses the chain rule recursively, and is implemented by back-propagation. This is simply a sketch of course; for further introduction, see Hastie et al. (2009) and Goodfellow et al. (2016).

To further clarify the use of deep nets, it is useful to make explicit analogies to more classical nonparametric techniques, leveraging the form $\hat{f}_{\mathrm{MLP}}(\boldsymbol{x}) = \tilde{\boldsymbol{x}}_L(\boldsymbol{x})' \boldsymbol{w}_L + b_L$. For a traditional series estimator, say smoothing splines, the two choices for the practitioner are the spline basis (the shape and the degree) and the number of terms (knots), commonly referred to as the smoothing and tuning parameters, respectively. In kernel regression, these would respectively be the shape of the kernel (and degree of local polynomial) and the bandwidth(s). For neural networks, the same phenomena are present: the architecture as a *whole* (the graph structure and activation function) are the smoothing parameters while the width and depth play the role of tuning parameters for a set architecture.

The architecture plays a crucial role in that it determines the approximation power of the network, and it is worth noting that because of the relative complexity of neural networks, such approximations, and comparisons across architectures, are not simple. It is comparatively obvious that quartic splines are more flexible than cubic splines (for the same number of knots) as is a higher degree local polynomial (for the same bandwidth). At a glance, it may not be clear what function class a given network architecture (width, depth, graph structure, and activation function) can approximate. As we will show below, the MLP architecture is not yet known to yield an optimal

9

approximation (for a given width and depth) and therefore we are only able to prove a bound with slower than optimal rate. As a final note, computational considerations are important for deep nets in a way that is not true conventionally; see Remarks 1, 2, and 3.

Just as for classical nonparametrics, for a fixed architecture, it is the tuning parameter choices that determine the rate of convergence (for a fixed smoothness of the underlying function). The recent wave of theoretical study of deep learning is still in its infancy. As such, there is no understanding yet of optimal architecture(s) or tuning parameters. Choices of both are quite difficult, and only preliminary research has been done (see e.g., Daniely, 2017; Telgarsky, 2016; Safran and Shamir, 2016; Mhaskar and Poggio, 2016a; Raghu et al., 2017, and references therein). Further exploration of these ideas is beyond the current scope. It is interesting to note that in some cases, a good approximation can be obtained even with a fixed width H, provided the network is deep enough, a very particular way of enriching the "sieve space" $\mathcal{F}_{\mathrm{DNN}}$; see Corollary 2.

In sum, for a user-chosen architecture $\mathcal{F}_{\mathrm{DNN}}$, encompassing the choices $\sigma(\cdot)$, U, L, W, and the graph structure, the final estimate is computed using observed samples $\boldsymbol{z}_i = (y_i, \boldsymbol{x}_i')'$, $i = 1, 2, \ldots, n$, of $\boldsymbol{Z}$, by solving

$$\widehat{f}_{\mathrm{DNN}} := \underset{\substack{f_\theta \in \mathcal{F}_{\mathrm{DNN}} \\ \|f_\theta\|_\infty \leq 2M}}{\arg\min} \sum_{i=1}^{n} \ell\left(f, \boldsymbol{z}_i\right). \tag{2.4}$$

Recall that θ collects, over all nodes, the weights and constants $\boldsymbol{w}$ and b. When (2.4) is restricted to the MLP class we denote the resulting estimator $\widehat{f}_{\mathrm{MLP}}$. The choice of M may be arbitrarily large, and is part of the definition of the class $\mathcal{F}_{\mathrm{DNN}}$. This is neither a tuning parameter nor regularization in the usual sense: it is not assumed to vary with n, and beyond being finite and bounding $\|f_*\|_\infty$ (see Assumption 1), no properties of M are required. This is simply a formalization of the requirement that the optimizer is not allowed to diverge on the function level in the l_∞ sense– the weakest form of constraint. It is important to note that while typically regularization will alter the approximation power of the class, that is not the case with the choice of M as we will assume that the true function $f_*(\boldsymbol{x})$ is bounded, as is standard in nonparametric analysis. With some extra notational burden, one can make the dependence of the bound on M explicit, though we omit this for clarity as it is not related to statistical issues.

Remark 1. In applications it is common to apply some form of regularization to the optimization of (2.4). However, in theory, the role of explicit regularization is unclear and may be unnecessary, as stochastic gradient descent presents good, if not better, solutions empirically (see Section 6 and Zhang et al., 2016). Regularization may improve empirical performance in low signal-to-noise ratio problems. A detailed investigation is beyond the scope of the current work, though we do investigate this numerically in Sections 5 and 6. There are many alternative regularization methods, including L_1 and L_2 (weight decay) penalties, drop out, and others. ⌋

2.2 Bounds and Convergence Rates for Multi-Layer Perceptrons

We can now state our main theoretical results: bounds and convergence rates for deep ReLU networks. All proofs appear in the Appendix. We study neural networks from a nonparametric point of view (e.g., White, 1989, 1992; Schmidt-Hieber, 2017; Liang, 2018; Bauer and Kohler, 2017, in specific scenarios). Chen and Shen (1998) and Chen and White (1999) share our goal, fast convergence rates for use in semiparametric inference, but focus on shallow, sigmoid-based networks compared to our deep, ReLU-based networks, though they consider dependent data which we do not. Our theoretical approach is quite different. In particular, Chen and White (1999) obtain sufficiently fast rates by following the approach of Barron (1993) in using Maurey's method (Pisier, 1981) for approximation, but applying the refinement of Makovoz (1996). Our analysis of deep nets instead employs localization methods (Koltchinskii and Panchenko, 2000; Bartlett et al., 2005; Koltchinskii, 2006, 2011; Liang et al., 2015), along with the recent approximation work of Yarotsky (2017, 2018) and complexity results of Bartlett et al. (2017).

The regularity conditions we require are collected in the following.

Assumption 1. *Assume that $\boldsymbol{z}_i = (y_i, \boldsymbol{x}_i')', 1 \leq i \leq n$ are i.i.d. copies of $\boldsymbol{Z} = (Y, \boldsymbol{X}) \in \mathcal{Y} \times [-1, 1]^d$, where X is continuously distributed. For an absolute constant $M > 0$, assume $\|f_*\|_\infty \leq M$ and $\mathcal{Y} \subset [-M, M]$.*

This assumption is fairly standard in nonparametrics. The only restriction worth mentioning is that the outcome is bounded. In many cases this holds by default (such as logistic regression, where $\mathcal{Y} = \{0, 1\}$) or count models (where $\mathcal{Y} = \{0, 1, \ldots, M\}$, with M limited by real-world constraints). For continuous outcomes, such as least squares regression, our restriction is not

substantially more limiting than the usual assumption of a model such as $Y = f_*(\boldsymbol{X}) + \varepsilon$, where $\boldsymbol{X}$ is compact-supported, f_* is bounded, and the stochastic error ε possesses many moments. Indeed, in many applications such a structure is only coherent with bounded outcomes, such as the common practice of including lagged outcomes as predictors. Next, the assumption of continuously distributed covariates is quite standard. From a theoretical point of view, covariates taking on only a few values can be conditioned on and then averaged over, and these will, as usual, not enter into the dimensionality which curses the rates. Discrete covariates taking on many values may be more realistically thought of as continuous, and it may be more accurate to allow these to slow the convergence rates. Our focus on $L_2(\boldsymbol{X})$ convergence allows for these essentially automatically. Finally, from a practical point of view, deep networks handle discrete covariates seamlessly and have demonstrated excellent empirical performance, which is in contrast to other more classical nonparametric techniques that may require manual adaptation.

Proceeding now to our results, we begin with the most important network architecture, the multi-layer perceptron. This is the most widely used network architecture in practice and an important contribution of our work is to cover this directly, along with ReLU activation. MLPs are now known to approximate smooth functions well, leading to our next assumption: that the target function f_* lies in a Sobolev ball with certain smoothness. Discussion of Sobolev spaces, and comparisons to Hölder and Besov spaces, can be found in Gine and Nickl (2016).

Assumption 2. *Assume f_* lies in the Sobolev ball $\mathcal{W}^{\beta,\infty}([-1,1]^d)$, with smoothness $\beta \in \mathbb{N}_+$,*

$$f_*(x) \in \mathcal{W}^{\beta,\infty}([-1,1]^d) := \left\{ f : \max_{\alpha, |\alpha| \le \beta} \operatorname*{ess\,sup}_{x \in [-1,1]^d} |D^\alpha f(x)| \le 1 \right\},$$

where $\alpha = (\alpha_1, \ldots, \alpha_d)$, $|\alpha| = \alpha_1 + \ldots + \alpha_d$ and $D^\alpha f$ is the weak derivative.

Under Assumptions 1 and 2 we obtain the following result, which, to the best of our knowledge, is new to the literature. In some sense, this is our main result for deep learning, as it deals with the most common architecture. We apply this in Sections 4 and 5 for semiparametric inference.

Theorem 1 (Multi-Layer Perceptron)**.** *Suppose Assumptions 1 and 2 hold. Let $\widehat{f}_{\mathrm{MLP}}$ be the deep MLP-ReLU network estimator defined by* (2.4), *restricted to $\mathcal{F}_{\mathrm{MLP}}$, for a loss function obeying* (2.1), *with width $H \asymp n^{\frac{d}{2(\beta+d)}} \log^2 n$ and depth $L \asymp \log n$. Then with probability at least $1 -$*

$\exp(-n^{\frac{d}{\beta+d}}\log^8 n)$, *for n large enough,*

(a) $\|\widehat{f}_{\mathrm{MLP}} - f_*\|_{L_2(\boldsymbol{x})}^2 \le C \cdot \left\{ n^{-\frac{\beta}{\beta+d}} \log^8 n + \dfrac{\log\log n}{n} \right\}$ *and*

(b) $\mathbb{E}_n\left[(\widehat{f}_{\mathrm{MLP}} - f_*)^2 \right] \le C \cdot \left\{ n^{-\frac{\beta}{\beta+d}} \log^8 n + \dfrac{\log\log n}{n} \right\}$,

for a constant $C > 0$ independent of n, which may depend on d, M, and other fixed constants.

Several aspects of this result warrant discussion. We build on the recent results of Bartlett et al. (2017), who find nearly-tight bounds on the Vapnik-Chervonenkis (VC) and Pseudo-dimension of deep nets. One contribution of our proof is to use a *scale sensitive* localization theory with *scale insensitive* measures, such as VC- or Pseudo-dimension, for deep neural networks for general smooth loss functions. For the special case of least squares regression, Koltchinskii (2011) uses a similar approach, and a similar result to our Theorem 1(a) can be derived for this case using his Theorem 5.2 and Example 3 (p. 85f).

This approach has two tangible benefits. First, we do not restrict the class of network architectures to have bounded weights for each unit (scale insensitive), in accordance to standard practice (Zhang et al., 2016) and in contrast to the classic sieve analysis with scale sensitive measure such as metric entropy. Moreover, this allows for a richer set of approximating possibilities, in particular allowing more flexibility in seeking architectures with specific properties, as we explore in the next subsection. Second, from a technical point of view, we are able to attain a faster rate on the second term of the bound, order n^{-1} in the sample size, instead of the $n^{-1/2}$ that would result from a direct application of uniform deviation bounds. This upper bound informs the trade offs between width and depth, and the approximation power, and may point toward optimal architectures for statistical inference.

This result gives a nonasymptotic bound that holds with high probability. As mentioned above, we will generally refer to our results simply as "rates" when this causes no confusion. This result relies on choosing H appropriately given the smoothness β of Assumption 2. Of course, the true smoothness is unknown and thus in practice the "β" appearing in H, and consequently in the convergence rates, need not match that of Assumption 2. In general, the rate will depend on the smaller of the two. Most commonly it is assumed that the user-chosen β is fixed and that the truth is smoother; witness the ubiquity of cubic splines and local linear regression. Rather than spell

out these consequences directly, we will tacitly assume the true smoothness is not less than the β appearing in H (here and below). Adaptive approaches, as in classical nonparametrics, may also be possible with deep nets, but are beyond the scope of this study.

Even with these choices of H and L, the bound of Theorem 1 is not optimal (for fixed β, in the sense of Stone (1982)). We rely on the explicit approximating constructions of Yarotsky (2017), and it is possible that in the future improved approximation properties of MLPs will be found, allowing for a sharpening of the results of Theorem 1 immediately, i.e. without change to our theoretical argument. At present, it is not clear if this rate can be improved, but it is sufficiently fast for valid inference.

2.3   Other Network Architectures

Theorem 1 covers only one specific architecture, albeit the most important one at present. However, given that this field is rapidly evolving, it is important to consider other possible architectures which may be beneficial in some cases. To this end, we will state a more generic result and then two specific examples: one to obtain a faster rate of convergence and one for fixed-width networks. All of these results are, at present, more of theoretical interest than practical value, as they are either agnostic about the network (thus infeasible) or rely on more limiting assumptions.

In order to be agnostic about the specific architecture of the network we need to be flexible in the approximation power of the class. To this end, we will replace Assumption 2 with the following generic assumption, rather more of a definition, regarding the approximation power of the network.

Assumption 3. *Let f_* lie in a class $\mathcal{F}$. For the feedforward network class $\mathcal{F}_{\mathrm{DNN}}$, used in* (2.4), *let the approximation error ϵ_{DNN} be*

$$\epsilon_{\mathrm{DNN}} := \sup_{\substack{f_* \in \mathcal{F}}} \inf_{\substack{f \in \mathcal{F}_{\mathrm{DNN}} \\ \|f\|_\infty \le 2M}} \|f - f_*\|_\infty \ .$$

It may be possible to require only an approximation in the $L_2(\boldsymbol{X})$ norm, but this assumption matches the current approximation theory literature and is more comparable with other work in nonparametrics, and thus we maintain the uniform definition.

Under this condition we obtain the following generic result.

14

Theorem 2 (General Feedforward Architecture)**.** *Suppose Assumptions 1 and 3 hold. Let $\widehat{f}_{\mathrm{DNN}}$ be the deep ReLU network estimator defined by (2.4), for a loss function obeying (2.1). Then with probability at least $1 - e^{-\gamma}$, for n large enough,*

(a) $\|\widehat{f}_{\mathrm{DNN}} - f_*\|_{L_2(\boldsymbol{x})}^2 \leq C \left(\dfrac{WL \log W}{n} \log n + \dfrac{\log \log n + \gamma}{n} + \epsilon_{\mathrm{DNN}}^2 \right)$ *and*

(b) $\mathbb{E}_n \left[(\widehat{f}_{\mathrm{DNN}} - f_*)^2 \right] \leq C \left(\dfrac{WL \log W}{n} \log n + \dfrac{\log \log n + \gamma}{n} + \epsilon_{\mathrm{DNN}}^2 \right),$

for a constant $C > 0$ independent of n, which may depend on d, M, and other fixed constants.

This is a more general than Theorem 1, covering the general deep ReLU network problem defined in (2.4), general feedforward architectures, and the general class of losses defined by (2.1). The same comments as were made following Theorem 1 apply here as well: the same localization argument is used with the same benefits. We explicitly use this in the next two corollaries, where we exploit the allowed flexibility in controlling ϵ_{DNN} by stating results for particular architectures. The bound here is not directly applicable without specifying the network structure, which will determine both the variance portion (through W, L, and U) and the approximation error. With these set, the bound becomes operational upon choosing γ, which can be optimized as desired, and this will immediately then yield a convergence rate.

Turning to special cases, we first show that the optimal rate of Stone (1982) can be attained, up to log factors. However, this relies on a rather artificial network structure, designated to approximate functions in a Sobolev space well, but without concern for practical implementation. Thus, while the following rate improves upon Theorem 1, we view this result as mainly of theoretical interest: establishing that (certain) deep ReLU networks are able to attain the optimal rate.

Corollary 1 (Optimal Rate)**.** *Suppose Assumptions 1 and 2 hold. Let $\widehat{f}_{\mathrm{OPT}}$ solve (2.4) using the (deep and wide) network of Yarotsky (2017, Theorem 1), with $W \asymp U \asymp n^{\frac{d}{2\beta+d}} \log n$ and depth $L \asymp \log n$, the following hold with probability at least $1 - e^{-\gamma}$, for n large enough,*

(a) $\|\widehat{f}_{\mathrm{OPT}} - f_*\|_{L_2(\boldsymbol{x})}^2 \leq C \cdot \left\{ n^{-\frac{2\beta}{2\beta+d}} \log^4 n + \dfrac{\log \log n + \gamma}{n} \right\}$ *and*

(b) $\mathbb{E}_n \left[(\widehat{f}_{\mathrm{OPT}} - f_*)^2 \right] \leq C \cdot \left\{ n^{-\frac{2\beta}{2\beta+d}} \log^4 n + \dfrac{\log \log n + \gamma}{n} \right\},$

for a constant $C > 0$ independent of n, which may depend on d, M, and other fixed constants.

Next, we turn to *very* deep networks that are very narrow, which have attracted substantial recent interest. Theorem 1 and Corollary 1 dealt with networks where the depth and the width grow with sample size. This matches the most common empirical practice, and is what we use in Sections 5 and 6. However, it is possible to allow for networks of *fixed* width, provided the depth is sufficiently large. The next result is perhaps the largest departure from the classical study of neural networks: earlier work considered networks with diverging width but fixed depth (often a single layer), while the reverse is true here. The activation function is of course qualitatively different as well, being piecewise linear instead of smooth. Using recent results (Mhaskar and Poggio, 2016b; Hanin, 2017; Yarotsky, 2018) we can establish the following rate for very deep, fixed-width MLPs.

Corollary 2 (Fixed Width Networks). *Let the conditions of Theorem 1 hold, with $\beta \geq 1$ in Assumption 2. Let $\widehat{f}_{\mathrm{FW}}$ solve (2.4) for an MLP with fixed width $H = 2d+10$ and depth $L \asymp n^{\frac{d}{2(2+d)}}$. Then with probability at least $1 - e^{-\gamma}$, for n large enough,*

(a) $\|\widehat{f}_{\mathrm{FW}} - f_*\|^2_{L_2(\boldsymbol{x})} \leq C \cdot \left\{ n^{-\frac{2}{2+d}} \log^2 n + \dfrac{\log\log n + \gamma}{n} \right\}$ *and*

(b) $\mathbb{E}_n \left[(\widehat{f}_{\mathrm{FW}} - f_*)^2 \right] \leq C \cdot \left\{ n^{-\frac{2}{2+d}} \log^2 n + \dfrac{\log\log n + \gamma}{n} \right\},$

for a constant $C > 0$ independent of n, which may depend on d, M, and other fixed constants.

This result is again mainly of theoretical interest. The class is only able to approximate well functions with $\beta = 1$ (cf. the choice of L) which limits the potential applications of the result because, in practice, d will be large enough to render this rate, unlike those above, too slow for use in later inference procedures. In particular, if $d \geq 3$, the sufficient conditions of Theorem 3 fail.

Finally, as mentioned following Theorem 1, our theory here will immediately yield a faster rate upon discovery of improved approximation power of this class of networks. In other words, for example, if a proof became available that fixed-width, very deep networks can approximate β-smooth functions (as in Assumption 2), then Corollary 2 will trivially be improvable to match the rate of Theorem 1. Similarly, if the MLP architecture can be shown to share the approximation power with that of Corollary 1, then Theorem 1 will itself deliver the optimal rate. Our proofs will not require adjustment.

Remark 2. Although there has been a great deal of work in easing implementation (optimization and tuning) of deep nets, it still may be a challenge in some settings, particularly when using

non-standard architectures. See also Remark 1. Given the renewed interest in deep networks, this is an area of study already (Hartford et al., 2017; Polson and Rockova, 2018) and we expect this to continue and that implementations will rapidly evolve. This is perhaps another reason that Theorem 1 is, at the present time, the most practically useful, but that (as just discussed) Theorem 2 will be increasingly useful in the future. ⌟

Remark 3. Our results can be extended easily to include piecewise linear activation functions beyond ReLU. Intuitively, being itself piecewise linear, appropriate combinations of a fixed number of ReLU functions can equal a piecewise linear function (with a fixed number of knots) and therefore the complexity and approximation power can be easily adjusted to this case. See Bartlett et al. (2017).

In principle, similar rates of convergence could be attained for other activation functions, given results on their approximation error. However, it is not clear what practical value would be offered due to computational issues (in which the activation choice plays a crucial role). Indeed, the recent switch to ReLU stems not from their greater approximation power, but from the fact that optimizing a deep net with sigmoid-type activation is unstable or impossible in practice. Thus, while it is certainly possible that we could complement the single-layer results with rates for sigmoid-based deep networks, these results would have no consequences for real-world practice.

From a purely practical point of view, several variations of the ReLU activation function have been proposed recently (including the so-called Leaky ReLU, Randomized ReLU, (Scaled) Exponential Linear Units, and so forth) and have been found in some experiments to improve optimization properties. It is not clear what theoretical properties these activation functions have or if the computational benefits persist more generically, though this area is rapidly evolving. We conjecture that our results could be extended to include these activation functions. ⌟

3   Parameters of Interest

We will use the results above, in particular Theorem 1, coupled with results in the semiparametric literature, to deliver valid asymptotic inference for causal effects. The novelty of our results is not in this semiparametric stage per se, but rather in delivering valid inference after relying on deep

learning for the first step estimation. In this section we define the parameters of interest, while asymptotic inference is discussed next.

We will focus, for concreteness, on causal parameters that are of interest across different disciplines: average treatment effects, expected utility (or profits) under different targeting policies, average effects on (non-)treated subpopulations, and decomposition effects. Our focus on causal inference with observational data is due to the popularity of these estimands both in applications and in theoretical work, thus allowing our results to be put to immediate use and easily compared to prior literature. The average treatment effect in particular is often used as a benchmark parameter for studying inference following machine learning (see references in the Introduction). However, armed with our results for deep neural networks we can cover a great deal more (some discussion is in Section 3.4).

The estimation of average causal effects is a well-studied problem, and we will give only a brief overview here. Recent reviews and further references are given by Belloni et al. (2017); Athey et al. (2017); Abadie and Cattaneo (2018). We consider the standard setup for program evaluation with observational data: we observe a sample of n units, each exposed to a binary treatment, and for each unit we observe a vector of pre-treatment covariates, $\boldsymbol{X} \in \mathbb{R}^d$, treatment status $T \in \{0, 1\}$, and a scalar post-treatment outcome Y. The observed outcome obeys $Y = TY(1) + (1 - T)Y(0)$, where $Y(t)$ is the (potential) outcome under treatment status $t \in \{0, 1\}$. The "fundamental problem" is that only $Y(0)$ or $Y(1)$ is observed for each unit, never both.

The crucial identification assumptions, which pertain to all the parameters we consider, are selection on observables, also known as ignorability, unconfoundedness, missingness at random, or conditional independence, and overlap, or common support. Let $p(\boldsymbol{x}) = \mathbb{P}[T = 1 | \boldsymbol{X} = \boldsymbol{x}]$ denote the propensity score and $\mu_t(\boldsymbol{x}) = E[Y(t) | \boldsymbol{X} = \boldsymbol{x}]$, $t \in \{0, 1\}$ denote the two outcome regression functions. We then assume the following throughout, beyond which, we will mostly need only regularity conditions for inference.

Assumption 4. *For $t \in \{0, 1\}$ and almost surely $\boldsymbol{X}$, $\mathbb{E}[Y(t)|T, \boldsymbol{X} = \boldsymbol{x}] = \mathbb{E}[Y(t)|\boldsymbol{X} = \boldsymbol{x}]$ and $\bar{p} \leq p(\boldsymbol{x}) \leq 1 - \bar{p}$ for some $\bar{p} > 0$.*

It will be useful to divide our discussion between parameters that are fully marginal averages, such as the average treatment effect, and those which are for specific subpopulations. Here, "sub-

populations" refer to the treated or nontreated groups, with corresponding parameters such as the treatment effect for the treated. Any parameter, in either case, can be studied for a suitable subpopulation defined by the covariates $\boldsymbol{X}$, such as a specific demographic group. Though causal effects as a whole share some structure, there are slight conceptual and notational differences. In particular, the form of the efficient influence function and doubly robust estimator is different for the two sets, but common within.

3.1 Full-Population Average Effect Parameters

Here we are interested in averages over the entire population. The prototypical parameter of interest is the average treatment effect:

$$\tau = \mathbb{E}[Y(1) - Y(0)]. \tag{3.1}$$

In the context of our empirical example, the treatment is being mailed a catalog and the outcome is dollars spent (results for the binary purchase decision are available on request). The average treatment effect, also referred to as "lift" in digital contexts, corresponds to the expected gain in revenue from an average individual receiving the catalog compared to the same person not receiving the catalog.

A closely related parameter of interest is the average realized outcome, which in general may be interpreted as the expected utility or welfare from a *counterfactual* treatment policy. In the context of our empirical application this is expected profits; in a medical context it would be the total health outcome. The question of interest here is whether a change in the treatment policy would be beneficial in terms of increasing outcomes, and this is judged using observational data. Intuitively, the average treatment effect is the expected gain from treating the "next" person, relative to if they had not been exposed. That is, it is the expected change in the outcome. Expected utility/profit, on the other hand, is concerned with the total outcome, not the difference in outcomes. In the context of our empirical application, we are interested in total sales rather than the change in sales. Our discussion is grounded in this language for easy comparison.

The parameter depends on a counterfactual/hypothetical treatment targeting strategy, which is often itself the object of evaluation. This is simply a rule that assigns a given set of characteristics (e.g. a consumer profile), determined by the covariates X, to treatment status: that

is, a known function (which may include randomization but is not estimated from the sample) $s(\boldsymbol{x}) : \text{supp}\{\boldsymbol{X}\} \mapsto \{0,1\}$. Note well that this is *not* necessarily the observed treatment: $s(\boldsymbol{x}_i) \neq t_i$. The policy maker may wish to evaluate the gain from targeting only a certain subset of customers, a price discrimination strategy, or comparisons of different such policies. Our assumptions, while standard, deliver identification of such counterfactuals at no cost.

The parameter of interest is expected utility, or profit, from a fixed policy, given by

$$\pi(s) = \mathbb{E}\big[s(\boldsymbol{X})Y(1) + (1 - s(\boldsymbol{X}))\,Y(0)\big], \tag{3.2}$$

where we make explicit the dependence on the policy $s(\cdot)$. Compare to Equation (3.1) and recall that the *observed* outcome obeys $Y = TY(1) + (1 - T)Y(0)$. Whereas τ is the gain in assigning the next person to treatment and is given by the difference in potential outcomes, $\pi(s)$ is the expected outcome that would be observed for the next person if the treatment rule were $s(\boldsymbol{x})$.

A natural question is whether a candidate targeting strategy, say $s'(\boldsymbol{x})$, is superior to baseline or status quo policy, $s_0(\boldsymbol{x})$. This amounts to testing the hypothesis $H_0 : \pi(s') \geq \pi(s_0)$. To evaluate this, we can study the difference in expected profits, which amounts to

$$\pi(s', s_0) = \pi(s') - \pi(s_0) = \mathbb{E}\big[(s'(\boldsymbol{X}) - s_0(\boldsymbol{X}))Y(1) + \big(s_0(\boldsymbol{X}) - s'(\boldsymbol{X})\big)\,Y(0)\big]. \tag{3.3}$$

Assumption 4 provides identification for $\pi(s)$ and $\pi(s', s_0)$, arguing analogously as for τ. Moreover, notice that $\pi(s', s_0) = \mathbb{E}[(s'(\boldsymbol{X}) - s_0(\boldsymbol{X}))(Y(1) - Y(0))] = \mathbb{E}[(s'(\boldsymbol{X}) - s_0(\boldsymbol{X}))\tau(\boldsymbol{X})]$, where $\tau(\boldsymbol{x}) = \mathbb{E}[Y(1) - Y(0) \mid \boldsymbol{X} = \boldsymbol{x}]$ is the conditional average treatment effect. The latter form makes clear that only those differently treated, of course, impact the evaluation of s' compared to s_0. The strategy s' will be superior if, on average, it targets those with a higher individual treatment effect. Estimating the optimal treatment policy from the data is discussed briefly in Section 3.3.

The common structure of these parameters is that they all involve full-population averages of the potential outcomes, possibly scaled by a known function. For these parameters, the influence function is known from Hahn (1998), and estimators based on the influence function are doubly robust, as they remain consistent if either the regression functions or the propensity score are correctly specified (Robins et al., 1994, 1995). With a slight abuse of terminology (since we are

omitting the centering), the influence function for a single average potential outcome, $t \in \{0, 1\}$, is given by, for $\boldsymbol{z} = (y, t, \boldsymbol{x}')'$,

$$\psi_t(\boldsymbol{z}) = \frac{\mathbb{1}\{T = t\}(y - \mu_t(\boldsymbol{x}))}{\mathbb{P}[T = t \mid \boldsymbol{X} = \boldsymbol{x}]} + \mu_t(\boldsymbol{x}). \tag{3.4}$$

Our estimation of τ, $\pi(s)$, and $\pi(s', s_0)$ will utilize sample averages of this function, with unknown objects replaced by estimators. Our use of influence functions here follows the recent literature in econometrics showing that the double robustness implies valid inference under weaker conditions on the first step nonparametric estimates (Farrell, 2015; Chernozhukov et al., 2018a).

3.2   Subpopulation Effect Parameters

The second type of causal effects of interest are based on potential outcomes averaged over only a specific treatment group. A single such average, for $t, t' \in \{0, 1\}$, is denoted by

$$\rho_{t,t'} = \mathbb{E}[Y(t) \mid T = t']. \tag{3.5}$$

Many interesting parameters are linear combinations of these for different t and t'. We focus on two for concreteness. (We could also consider averages restricted by targeting-type functions, as in expected utility/profit, but for brevity we omit this.) The most well-studied of these parameters is the treatment effect on the treated, given by

$$\tau_{1,0} = \mathbb{E}[Y(1) - Y(0) \mid T = 1] = \rho_{1,1} - \rho_{0,1}. \tag{3.6}$$

To appreciate the breadth of this framework, and the applicability of our causal inference results, we also consider a decomposition parameter, a semiparametric analogue of Oaxaca-Blinder (Kitagawa, 1955; Oaxaca, 1973; Blinder, 1973). In this context, the "treatment" variable T is typically not a treatment assignment per se, but rather an exogenous covariate such as a demographic indicator, perhaps most commonly a male/female indicator. See Fortin et al. (2011) for a complete discussion and further references. The parameter of interest in this case is the decomposition of $\Delta = \mathbb{E}[Y(1) \mid T = 1] - \mathbb{E}[Y(0) \mid T = 0]$, into the difference in the covariate distributions and the difference in expected outcomes. These can be written as functions of different $\rho_{t,t'}$. For example,

$\Delta_X = \mathbb{E}[Y(1)|T{=}1] - \mathbb{E}[Y(1)|T{=}0] = \mathbb{E}[\mu_1(\boldsymbol{X})|T{=}1] - \mathbb{E}[\mu_1(\boldsymbol{X})|T{=}0] = \rho_{1,1} - \rho_{1,0}$. We are in general interested in

$$\Delta = \Delta_X + \Delta_\mu, \qquad \Delta_X = \rho_{1,1} - \rho_{1,0}, \qquad \text{and} \qquad \Delta_\mu = \rho_{1,0} - \rho_{0,0}. \tag{3.7}$$

Just as in the case of full-population averages, the influence function is known and leads to a doubly robust estimator. For a single $\rho_{t,t'}$, the (uncentered) influence function is (cf. (3.4)):

$$\psi_{t,t'}(\boldsymbol{z}) = \frac{\mathbb{P}[T = t' \mid \boldsymbol{X} = \boldsymbol{x}]}{\mathbb{P}[T = t']} \frac{\mathbb{1}\{T = t\}(y - \mu_t(\boldsymbol{x}))}{\mathbb{P}[T = t \mid \boldsymbol{X} = \boldsymbol{x}]} + \frac{\mathbb{1}\{T = t'\}\mu_t(\boldsymbol{x})}{\mathbb{P}[T = t']}. \tag{3.8}$$

Estimation and inference requires, as above, estimation of the propensity scores and regression functions, depending on the exact choices of t and t', and here we also require the marginal probability of treatments.

3.3   Optimal Policies

Moving beyond a fixed parameter, our results on deep neural networks can be used to address optimal targeting. In the notation of Section 3.1, this amounts to finding a policy, say $s_\star(\boldsymbol{x})$, that maximizes a given measure of utility stemming from treatment, generally the expected gain relative to a baseline policy. In Section 3.1 we considered the utility (or profit) difference between two given strategies, a candidate $s'(\boldsymbol{x})$ and a baseline $s_0(\boldsymbol{x})$. Instead of inference on $\pi(s', s_0)$, we can use the data to find the $s_\star(\boldsymbol{x})$ which maximizes the gain relative to the baseline. This problem has been widely studied in econometrics and statistics; for detailed discussion and numerous references see Manski (2004), Hirano and Porter (2009), Kitagawa and Tetenov (2018), and Athey and Wager (2018). In particular, the latter noticed using the locally robust framework allows policy optimization under nearly the same conditions as inference and proved fast convergence rates of the estimated policy in terms of regret.

More formally, we want to find the optimal choice $s_\star(\boldsymbol{x})$ in some policy/action space $\mathcal{S}$. The policy space, and thus its complexity, is user determined. Simple examples include simple decision trees or univariate-based strategies; more can be found in the references above. Recall that $\pi(s', s_0) = \mathbb{E}[Y(s')] - \mathbb{E}[Y(s_0)] = \mathbb{E}[(s'(\boldsymbol{X}) - s_0(\boldsymbol{X}))\tau(\boldsymbol{X})]$, where $\tau(\boldsymbol{x}) = \mathbb{E}[Y(1) - Y(0) \mid \boldsymbol{X} = \boldsymbol{x}]$

is the conditional average treatment effect. Given a space $\mathcal{S}$, we wish to find the policy $s_\star(\boldsymbol{x}) \in \mathcal{S}$ which solves $\max_{s' \in \mathcal{S}} \pi(s', s_0)$. The main result of Athey and Wager (2018) is that replacing π with the doubly-robust $\hat{\pi}$ of Equation (4.3), and minimizing the empirical analogue of regret, one obtains an estimator $\hat{s}(\boldsymbol{x})$ of the optimal policy that obeys the regret bound $\pi(s_\star, s_0) - \pi(\hat{s}, s_0) = O_P(\sqrt{\mathrm{VC}(\mathcal{S})/n})$ (a formal statement would be notationally burdensome). The complexity of the user-chosen policy space enters the bound through its VC dimension. Simple, interpretable policy classes often have bounded or slowly-growing dimension, implying rapid convergence.

3.4   Other Estimands

There are of course many other contexts where first-step deep learning is useful. Only trivial extensions to the above would be required for other causal effects, such as multi-valued treatments (reviewed by Cattaneo, 2010) and others with doubly-robust estimators (Sloczynski and Wooldridge, 2018). Further, under selection on observables, treatment effects, missing data, measurement error, and data combination are equivalent, and thus all our results apply immediately to those contexts. For reviews of these and Assumption 4 more broadly, see Chen et al. (2004); Tsiatis (2006); Heckman and Vytlacil (2007); Imbens and Wooldridge (2009).

Moving beyond causal effects, any estimand with a locally/doubly robust estimator depending only on target functions falling into our class of losses can be covered using the results of Section 2. For example, estimands requiring distribution estimation require further study; see Liang (2018) for recent results via Generative Adversarial Networks (GANs). More precisely, along with regularity conditions, our theory can be used to verify the conditions of Chernozhukov et al. (2018c), who treat more general semiparametric estimands using local robustness, sometimes relying on sample splitting or cross fitting. Further in this vein, our results on deep neural networks can be used to address optimal targeting, i.e., finding the policy, say $s_\star(\boldsymbol{x})$, that maximizes a given measure of utility, by applying the results of Athey and Wager (2018), who noticed that using the locally robust framework allows policy optimization.

More broadly, the learning of features using deep neural networks is becoming increasingly popular and our results speak to this context directly. To illustrate, consider the simple example of a linear model where some predictors are features learned from independent data. Here, the object of interest is the fixed-dimension coefficient vector $\boldsymbol{\lambda}$, which we assume can be partitioned

as $\boldsymbol{\lambda} = (\boldsymbol{\lambda}_1', \boldsymbol{\lambda}_2')'$ according to the model $Y = \boldsymbol{f}(\boldsymbol{X})'\boldsymbol{\lambda}_1 + W'\boldsymbol{\lambda}_2 + \varepsilon$. The features $\boldsymbol{f}(\boldsymbol{X})$, often a "score" of some type, are generally learned from auxiliary (and independent) data. For a recent example, see Liu et al. (2017). In such cases, inference on $\boldsymbol{\lambda}$ can proceed directly, as long as care is taken to interpret the results. See Section 4.2.

4 Asymptotic Inference

We now turn to asymptotic inference for the causal parameters discussed above. We first define the estimators, which are based on sample averages of the (uncentered) influence functions (3.4) and (3.8). We then give a generic result for single averages which can then be combined for inference on a given parameter of interest. Below we discuss inference under randomized treatment and using sample splitting.

Throughout, we assume we have a sample $\{\boldsymbol{z}_i = (y_i, t_i, \boldsymbol{x}_i')'\}_{i=1}^n$ from $\boldsymbol{Z} = (Y, T, \boldsymbol{X}')'$. We then form

$$\hat{\psi}_t(\boldsymbol{z}_i) = \frac{\mathbb{1}\{t_i = t\}(y_i - \hat{\mu}_t(\boldsymbol{x}_i))}{\hat{\mathbb{P}}[T = t \mid \boldsymbol{X} = \boldsymbol{x}_i]} + \hat{\mu}_t(\boldsymbol{x}_i), \tag{4.1}$$

where $\hat{\mathbb{P}}[T = t \mid \boldsymbol{X} = \boldsymbol{x}_i] = \hat{p}(\boldsymbol{x}_i)$ for $t = 1$ and $1 - \hat{p}(\boldsymbol{x}_i)$ for $t = 0$, and similarly

$$\hat{\psi}_{t,t'}(\boldsymbol{z}_i) = \frac{\hat{\mathbb{P}}[T = t' \mid \boldsymbol{X} = \boldsymbol{x}_i]}{\hat{\mathbb{P}}[T = t']} \frac{\mathbb{1}\{t_i = t\}(y_i - \hat{\mu}_t(\boldsymbol{x}_i))}{\hat{\mathbb{P}}[T = t \mid \boldsymbol{X} = \boldsymbol{x}_i]} + \frac{\mathbb{1}\{t_i = t'\}\hat{\mu}_t(\boldsymbol{x}_i)}{\hat{\mathbb{P}}[T = t']}, \tag{4.2}$$

where $\hat{\mathbb{P}}[T = t']$ is simply the sample frequency $\mathbb{E}_n[\mathbb{1}\{t_i = t'\}]$.

For the first stage estimates appearing in (4.1) and (4.2) we use our results on deep nets, and Theorem 1 in particular. Specifically, the estimated propensity score, $\hat{p}(\boldsymbol{x})$, is the estimate that results from solving (2.4), with the MLP architecture, for the logistic loss (2.3) with T as the outcome. Similarly, for each status $t \in \{0, 1\}$, we can let $\hat{\mu}_t(\boldsymbol{x})$ be the deep-MLP estimate of $f_*(\boldsymbol{x}) = \mathbb{E}[Y|T = t, \boldsymbol{X} = \boldsymbol{x}]$, solving (2.4) for least squares loss, (2.2), with outcome Y, using only observations with $t_i = t$. However, it is worth noting that the theoretically-equivalent joint estimation of Equation (5.1) performs much better, as the two groups may share features. To state the results, let β_p and β_μ be the smoothness parameters of Assumption 2 for the propensity score and outcome models, respectively.

We then obtain inference using the following results, essentially taken from Farrell (2015).

Similar results are given by Belloni et al. (2017) and Chernozhukov et al. (2018a). All of these provide high-level conditions for valid inference, and none verify these for deep nets as we do here.

Theorem 3. *Suppose that $\{z_i = (y_i, t_i, x_i')'\}_{i=1}^n$ are i.i.d. obeying Assumption 4 and the conditions Theorem 1 hold with $\beta_p \wedge \beta_\mu > d$. Further assume that, for $t \in \{0, 1\}$, $\mathbb{E}[(s(\boldsymbol{X})\psi_t(\boldsymbol{Z}))^2 | \boldsymbol{X}]$ is bounded away from zero and, for some $\delta > 0$, $\mathbb{E}[(s(\boldsymbol{X})\psi_t(\boldsymbol{Z}))^{4+\delta} | \boldsymbol{X}]$ is bounded. Then the deep MLP-ReLU network estimators defined above obey the following, for $t \in \{0, 1\}$,*

(a) $\mathbb{E}_n[(\hat{p}(\boldsymbol{x}_i) - p(\boldsymbol{x}_i))^2] = o_P(1)$ *and* $\mathbb{E}_n\left[(\hat{\mu}_t(\boldsymbol{x}_i) - \mu_t(\boldsymbol{x}_i))^2\right] = o_P(1)$,

(b) $\mathbb{E}_n[(\hat{\mu}_t(\boldsymbol{x}_i) - \mu_t(\boldsymbol{x}_i))^2]^{1/2}\mathbb{E}_n[(\hat{p}(\boldsymbol{x}_i) - p(\boldsymbol{x}_i))^2]^{1/2} = o_P(n^{-1/2})$, *and*

(c) $\mathbb{E}_n[(\hat{\mu}_t(\boldsymbol{x}_i) - \mu_t(\boldsymbol{x}_i))(1 - \mathbb{1}\{t_i = t\}/\mathbb{P}[T = t | \boldsymbol{X} = \boldsymbol{x}_i])] = o_P(n^{-1/2})$,

and therefore, if $\hat{p}(\boldsymbol{x}_i)$ is bounded inside $(0, 1)$, for a given $s(\boldsymbol{x})$ and $t \in \{0, 1\}$, we have

$$\sqrt{n}\mathbb{E}_n\left[s(\boldsymbol{x}_i)\hat{\psi}_t(\boldsymbol{z}_i) - s(\boldsymbol{x}_i)\psi_t(\boldsymbol{z}_i)\right] = o_P(1) \quad \text{and} \quad \frac{\mathbb{E}_n[(s(\boldsymbol{x}_i)\hat{\psi}_t(\boldsymbol{z}_i))^2]}{\mathbb{E}_n[(s(\boldsymbol{x}_i)\psi_t(\boldsymbol{z}_i))^2]} = o_P(1),$$

as well as,

$$\sqrt{n}\mathbb{E}_n\left[\hat{\psi}_{t,t'}(\boldsymbol{z}_i) - \psi_{t,t'}(\boldsymbol{z}_i)\right] = o_P(1) \quad \text{and} \quad \frac{\mathbb{E}_n[\hat{\psi}_{t,t'}(\boldsymbol{z}_i)^2]}{\mathbb{E}_n[\psi_{t,t'}(\boldsymbol{z}_i)^2]} = o_P(1).$$

This result, our main inference contribution, shows exactly how deep learning delivers valid asymptotic inference for our parameters of interest. Theorem 1 (a generic result using Theorem 2 could be stated) proves that the nonparametric estimates converge sufficiently fast, as formalized by conditions (a), (b), and (c), enabling feasible efficient semiparametric inference. In general, these are implied by, but may be weaker than, the requirement of that the first step estimates converge faster than $n^{-1/4}$, which our results yield for deep ReLU nets. The first is a mild consistency requirement. The second requires a rate, but on the product of the two estimates, which can be satisfied under weaker conditions. Finally, the third condition is the strongest. Intuitively, this condition arises from a "leave-in" type remainder, and as such, it can be weakened using sample splitting Chernozhukov et al. (2018a); Newey and Robins (2018). We opt to maintain (c) exactly because deep nets are not amenable to either simple leave-one-out forms (as are, e.g., classical kernel regression) or to sample splitting, being a data hungry method the gain in theoretically weaker rate requirements may not be worth the price paid in constants in finite samples. Instead,

we employ our localization analysis, as was used to obtain the results of Section 2, to verify (c) directly (see Lemma 10); this appears to be a novel application of localization, and this approach may be useful in future applications of second-step inference using machine learning methods.

From this result we immediately obtain inference for all the causal parameters discussed above. For the full-population averages, for example, we would form

$$
\begin{aligned}
\hat{\tau} &= \mathbb{E}_n \left[\hat{\psi}_1(\boldsymbol{z}_i) - \hat{\psi}_0(\boldsymbol{z}_i) \right], \\
\hat{\pi}(s) &= \mathbb{E}_n \left[s(\boldsymbol{x}_i)\hat{\psi}_1(\boldsymbol{z}_i) + (1 - s(\boldsymbol{x}_i))\hat{\psi}_0(\boldsymbol{z}_i) \right], \\
\hat{\pi}(s', s_0) &= \mathbb{E}_n \left[[s'(\boldsymbol{x}_i) - s_0(\boldsymbol{x}_i)]\hat{\psi}_1(\boldsymbol{z}_i) - [s'(\boldsymbol{x}_i) - s_0(\boldsymbol{x}_i)]\hat{\psi}_0(\boldsymbol{z}_i) \right].
\end{aligned}
\tag{4.3}
$$

The estimator $\hat{\tau}$ is exactly the doubly/locally robust estimator of the average treatment effect that is standard in the literature. The estimators for profits can be thought of as the doubly robust version of the constructs described in Hitsch and Misra (2018). Furthermore, to add a per-unit cost of treatment/targeting c and a margin m, simply replace ψ_1 with $m\psi_1 - c$ and ψ_0 with $m\psi_0$. Similarly, $\hat{\tau}_{1,0}$, $\hat{\Delta}_X$, and $\hat{\Delta}_\mu$ would be linear combinations of different $\hat{\rho}_{t,t'} = \mathbb{E}_n[\hat{\psi}_{t,t'}(\boldsymbol{z}_i)]$.

It is immediate from Theorem 3 that all such estimators are asymptotically Normal. The asymptotic variance can be estimated by simply replacing the sample first moments of (4.3) with second moments. That is, looking at $\hat{\pi}(s)$ to fix ideas,

$$
\sqrt{n}\hat{\Sigma}^{-1/2}\left(\hat{\pi}(s) - \pi(s)\right) \xrightarrow{d} \mathcal{N}(0,1), \quad \text{with} \quad \hat{\Sigma} = \mathbb{E}_n\left[\left(s(\boldsymbol{x}_i)\hat{\psi}_1(\boldsymbol{z}_i) + (1 - s(\boldsymbol{x}_i))\hat{\psi}_0(\boldsymbol{z}_i)\right)^2\right] - \hat{\pi}(s)^2.
$$

The others are similar. Further, Theorem 3 can be generalized straightforwardly to yield uniformly valid inference, following the approach of Romano (2004), exactly as in Belloni et al. (2014) or Farrell (2015).

Finally, we note that our focus with Theorem 3 is showcasing the practical utility of deep learning. Our use of local/double robustness here is toward the aim of attaining feasible inference without requiring more detailed assumptions on the machine learning step. This comes at the expense of, for example, stronger-than-minimal smoothness assumptions. That is, the requirement that $\beta_p \wedge \beta_\mu > d$ is not minimal, and moreover, neither is the weaker condition $\beta_p \wedge \beta_\mu > d/2$ that would be required after applying Corollary 1 instead of Theorem 1. Obtaining a Gaussian limit, and

26

possibly semiparametric efficiency, under minimal conditions has been studied by many, dating at least to Bickel and Ritov (1988); see Robins et al. (2009) for recent results and references on optimal estimation and minimal conditions. For causal inference, Chen et al. (2008) and Athey et al. (2018) obtain semiparametric efficiency under strictly weaker conditions than ours on $p(\boldsymbol{x})$ (the former under minimal smoothness on $\mu_t(\boldsymbol{x})$ and the latter under a sparsity in a high-dimensional linear model). Further, as above, cross-fitting (Newey and Robins, 2018) paired with local robustness may yield weaker smoothness conditions by providing underfitting" robustness (i.e. weakening bias-related tuning parameter assumptions). On the other hand, weaker variance-related assumptions, or "overfitting" robust inference procedures, (Cattaneo and Jansson, 2018; Cattaneo et al., 2018), may also be possible following deep learning, but are less automatic at present. Finally, other methods designed for causal inference under relaxed assumptions may be useful here, such as the recently developed extensions to doubly robust estimation (Tan, 2018) and inverse weighting (Ma and Wang, 2018): pursuing these in the context of deep learning is left to future work.

4.1   Inference Under Randomization

Our analysis thus far has focused on observational data, but it is worth spelling out results for randomized experiments. This is particularly important in the Internet age, where experimentation is common, vast amounts of data are available, and effects are often small in magnitude (Taddy et al., 2015). Indeed, our empirical illustration, detailed in the next section, stems from an experiment with 300,000 units and hundreds of covariates. When treatment is randomized, inference can be done directly using the mean outcomes in the treatment and control groups, such as the difference for the average treatment effect or the corresponding weighted sum for profit. However, pre-treatment covariates can be used to increase efficiency (Hahn, 2004).

   We will focus on the simple situation of a purely randomized binary treatment, but our results can be extended naturally to other randomization schemes. We formalize this with the following.

Assumption 5 (Randomized Treatment). *T is independent of $Y(0)$, $Y(1)$, and X, and is distributed Bernoulli with parameter p^*, such that $\bar{p} \leq p^* \leq 1 - \bar{p}$ for some $\bar{p} > 0$.*

   Under this assumption, the obvious simplification is that the propensity score need not be estimated using the covariates, but can be replaced with the (still nonparametric) sample frequency:

$\hat{p}(\boldsymbol{x}_i) \equiv \hat{p} = \mathbb{E}_n[t_i]$. This is plugged into Equation (4.3) and estimation and inference proceeds as above. Only rate conditions on the regression functions $\hat{\mu}_t(x)$ are needed. Further, conditions (a) and (b) of Theorem 3 collapse, as $\hat{p}$ is root-n consistent, leaving only condition (c) to be verified. Again, cross-fitting can be used in theory to remove this condition and thus weaken the requirement that $\beta_\mu > d$, but we maintain this for simplicity. We collect this into the following result, which is a trivial corollary of Theorem 3.

Corollary 3. *Let the conditions of Theorem 3 hold with Assumption 5 in place of Assumption 4 and only $\beta_\mu > d$. Then deep MLP-ReLU network estimators obey*

(a′) $\mathbb{E}_n\left[(\hat{\mu}_t(\boldsymbol{x}_i) - \mu_t(\boldsymbol{x}_i))^2\right] = o_P(1)$ *and*

(c′) $\mathbb{E}_n[(\hat{\mu}_t(\boldsymbol{x}_i) - \mu_t(\boldsymbol{x}_i))(1 - \mathbb{1}\{t_i = t\}/p^*)] = o_P(n^{-1/2})$

and the conclusions of Theorem 3 hold.

4.2   Sample Splitting

Sample splitting may be used to obtain valid inference in cases, unlike those above, where the parameter of interest itself is learned from the data. For the causal estimands above, the regression functions and propensity score must be estimated, but these are nuisance functions. This is not true in the inference after policy or feature learning (Sections 3.3 and 3.4). For policy learning, our results can be used to verify the high-level conditions of Athey and Wager (2018), though they require the additional condition of uniform consistency of the first stage estimators, and for machine learning estimators this is not clearly innocuous. However, this gives only point estimation.

Sample splitting is used in the obvious way: the first subsample, or more generally, independent auxiliary data, is used to learn the features or optimal policy, and then Theorem 3 is applied in the second subsample, conditional on the results of the first. For policy learning this delivers valid inference on $\pi(\hat{s})$ or $\pi(\hat{s}, s_0)$, while for the simple example of feature learning in a linear model we obtain inference on the parameters defined by the "model" $Y = \boldsymbol{f}(\boldsymbol{X})'\boldsymbol{\lambda}_1 + W'\boldsymbol{\lambda}_2 + \varepsilon$, where $\boldsymbol{f}(\boldsymbol{X})$ is estimated from auxiliary data. Care must be taken in interpreting the results. The results of the first-subsample estimation are effectively conditioned upon in the inference stage, redefining the target parameter to be in terms of the learned object. In many contexts this may be sufficient

([Chernozhukov et al., 2018b](#)), but further assumptions will generally be needed to assume that the first subsample has recovered the true population object. To fix ideas, consider policy learning: inference on $\pi(\hat{s}, s_0)$, conditional on the map $\hat{s}(\boldsymbol{x})$ learned in the first subsample, is immediate and requires no additional assumptions, but inference on $\pi(s_\star, s_0)$ is not obvious without further conditions.

5   Empirical Application

To illustrate our results, Theorems 1 and 3 in particular, we study, from a marketing point of view, a randomized experiment from a large US retailer of consumer products. The outcome of interest is consumer spending and the treatment is a catalog mailing. The firm sells directly to the customer (as opposed to via retailers) using a variety of channels such as the web and mail. The data consists of nearly three hundred thousand (292,657) consumers chosen at random from the retailer's database. Of these, 2/3 were randomly chosen to receive a catalog, and in addition to treatment status, we observe roughly one hundred fifty covariates, including demographics, past purchase behaviors, interactions with the firm, and other relevant information. For more on the data and a complete discussion of the decision making issues, we refer the reader to [Hitsch and Misra (2018)](#) (we use the 2015 sample). That paper studied various estimators, both traditional and modern, of average and heterogeneous causal effects. Importantly, they did not consider neural networks. Our results show that deep nets are at least as good as (and sometimes better than) the best methods in [Hitsch and Misra (2018)](#).

In terms of motivation, a key element of a firm's toolkit is the design and implementation of targeted marketing instruments. These instruments, aiming to induce demand, often contain advertising and informational content about the firms offerings. The targeting aspect thus boils down to the selection of which particular customers should be sent the material. This is a particularly important decision since the costs of creation and dissemination of the material can accumulate rapidly, particularly over a large customer base. For a typical retailer engaging in direct marketing the costs of sending out a catalog can be close to a dollar per targeted customer. With millions of catalogs being sent out, the cost of a typical campaign is quite high.

Given these expenses, an important problem for firms is ascertaining the causal effects of such

targeted mailing, and then using these effects to evaluate potential targeting strategies. At a high level, this approach is very similar to modern personalized medicine where treatments have to be targeted. In these contexts, both the treatment and the targeting can be costly, and thus careful assessment of $\pi(s)$ (interpreted as welfare) is crucial for decision making.

The outcome of interest for the firm is customer spending. This is the total amount of money that a given customer spends on purchases of the firm's products, within a specified time window. For the experiment in question the firm used a window of three months, and aggregated sales from all available purchase channels including phone, mail, and the web. In our data 6.2% of customers made a purchase. Overall mean spending is \$7.31; average spending conditional on buying is \$117.7, with a standard deviation of \$132.44. The idea then is to examine the incremental effect that the catalog had on this spending metric. Table 1 presents summary statistics for the outcome and treatment. Figure 3 displays the complete density of spending conditional on a purchase, which is quite skewed.
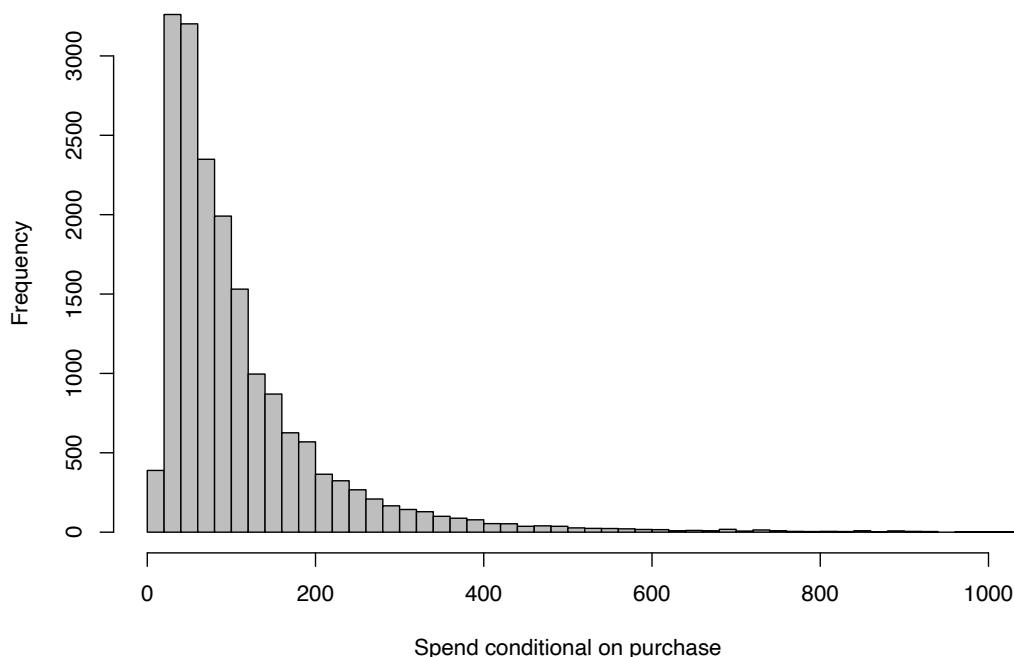

Figure 3: Spend Conditional on Purchase

Table 1: Summary Statistics

	Mean	SD	N
Purchase	0.062	0.24	292657
Spend	7.311	43.55	292657
Spend Conditional on Purchase	117.730	132.44	18174
Treatment	0.669	0.47	292657
Purchase | Treatment=1	0.069	0.25	195821
Purchase | Treatment=0	0.047	0.21	96836
Spend | Treatment=1	8.158	44.71	195821
Spend | Treatment=0	5.597	41.04	96836

5.1 Implementation Details

We estimated deep neural nets under a variety of architecture choices. In what follows we present eight examples and focus on one particular architecture to compute various statistics and tests to illustrate the use of the theory developed above. All computation was done using TensorFlow™.

For treatment effect and profit estimation we follow Equations (4.1) and (4.3). Because treatment is randomized, we apply Corollary 3, and thus, only require estimates of the regression functions $\mu_t(\boldsymbol{x}) = E[Y(t)|\boldsymbol{X} = \boldsymbol{x}]$, $t \in \{0, 1\}$. An important implementation detail, from a computation point of view (recall Remark 2) is that we will estimate $\mu_0(\boldsymbol{x})$ and $\tau(\boldsymbol{x})$ (and thereby $\mu_1(\boldsymbol{x})$) *jointly* (results from separate estimation are available). To be precise, recalling Equations (2.2) and (2.4), we solve

$$\begin{pmatrix} \hat{\mu}_0(\boldsymbol{x}) \\ \hat{\tau}(\boldsymbol{x}) = \hat{\mu}_1(\boldsymbol{x}) - \hat{\mu}_0(\boldsymbol{x}) \end{pmatrix} := \underset{\tilde{\mu}_0, \tilde{\tau}}{\arg\min} \sum_{i=1}^n \frac{1}{2} \Big(y_i - \tilde{\mu}_0(\boldsymbol{x}_i) - \tilde{\tau}(\boldsymbol{x}_i) t_i \Big)^2 \qquad (5.1)$$

where the minimization is over the relevant network architecture. Recall that, in the context of our empirical example y_i is the customer's spending, $\boldsymbol{x}_i$ are her characteristics, and t_i indicates receipt of a catalog. In this format, $\mu_0(\boldsymbol{x}_i)$ reflects base spending and $\tau(\boldsymbol{x}) = \mu_1(\boldsymbol{x}_i) - \mu_0(\boldsymbol{x}_i)$ is the conditional average treatment effect of the catalog mailing. In our application, this joint estimation outperforms separately estimating each $\mu_t(\boldsymbol{x})$ on the respective samples (though these two approaches are equivalent theoretically).

The details of the eight deep net architectures are presented in Table 2. See Section 2.1 for an

introduction to the terminology and network construction. Most yielded similar results, both in terms of fit and final estimates. A key measure of fit reported in the final column of the table is the portion of $\hat{\tau}(\boldsymbol{x}_i)$ that were negative. As argued by Hitsch and Misra (2018), it is implausible under standard marketing or economic theory that receipt of a catalog causes lower purchasing. On this metric of fit, deep nets perform as well as, and sometimes better than, the best methods found by Hitsch and Misra (2018): Causal KNN with Treatment Effect Projections (detailed therein) or Causal Forests (Wager and Athey, 2018). Figure 4 shows the distribution of $\hat{\tau}(\boldsymbol{x}_i)$ across customers for each of the eight architectures. While there are differences in the shapes of the densities, the mean and variance estimates are nonetheless quite similar.

Table 2: Deep Network Architectures

Architecture	Learning Rate	Widths $[H_1, H_2, ...]$	Dropout $[H_1, H_2, ...]$	Total Parameters	Validation Loss	Training Loss	$\mathbb{P}_n[\hat{\tau}(\boldsymbol{x}_i) < 0]$
1	0.0003	[60]	[0.5]	8702	1405.62	1748.91	0.0014
2	0.0003	[100]	[0.5]	14502	1406.48	1751.87	0.0251
3	0.0001	[30, 20]	[0.5, 0]	4952	1408.22	1751.20	0.0072
4	0.0009	[30, 10]	[0.3, 0.1]	4622	1408.56	1751.62	0.0138
5	0.0003	[30, 30]	[0, 0]	5282	1403.57	1738.59	0.0226
6	0.0003	[30, 30]	[0.5, 0]	5282	1408.57	1755.28	0.0066
7	0.0003	[100, 30, 20]	[0.5, 0.5, 0]	17992	1408.62	1751.52	0.0103
8	0.00005	[80, 30, 20]	[0.5, 0.5, 0]	14532	1413.70	1756.93	0.0002

Notes: All networks use the ReLU activation function. The width of each layer is shown, e.g. Architecture 3 consists of two layers, with 30 and 20 hidden units respectively. The final column shows the portion of estimated individual treatment effects below zero.

5.2   Results

We present now results for treatment effects, utility/profits, and targeting policy evaluations. Table 3 shows the estimates of the average treatment effect from the eight network architectures along with their respective 95% confidence intervals. These results are constructed following Section 4, using Equations (4.1) and (4.3) in particular, and valid by Corollary 3. Because this is an experiment, we can compare to the standard unadjusted difference in means, which yields an average treatment effect of 2.561.

Turning to expected profits, we estimate $\pi(s) = \mathbb{E}\big[s(\boldsymbol{X})(mY(1) - c) + (1 - s(\boldsymbol{X}))\, mY(0)\big]$,
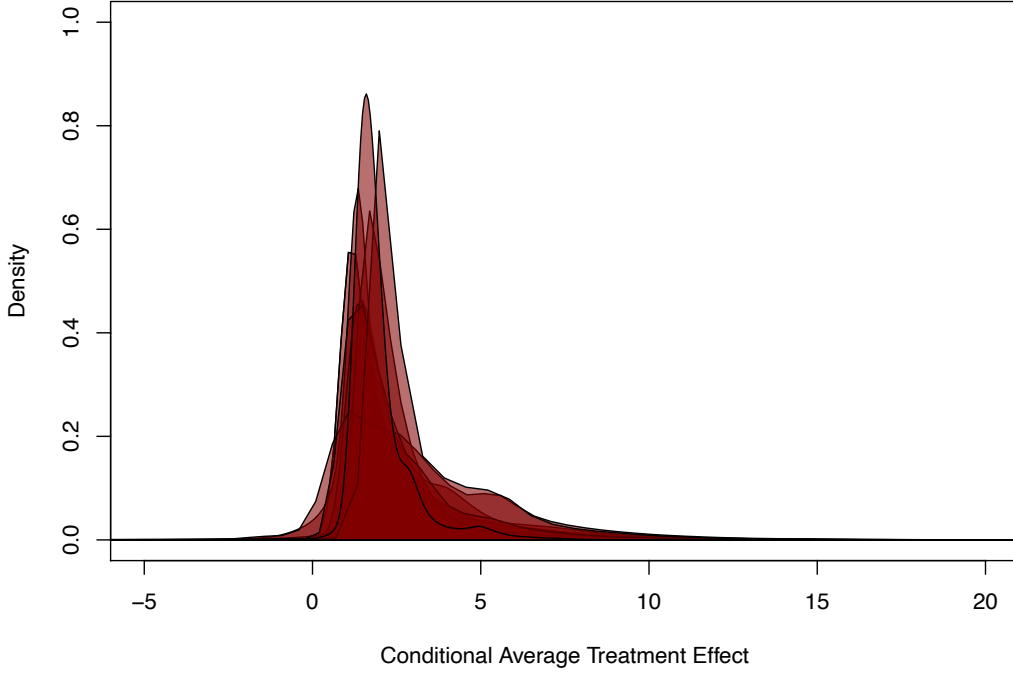
Figure 4: Conditional Average Treatment Effects Across Architectures

Table 3: Average Treatment Effect Estimates and 95% Confidence Intervals

Architecture	Average Treatment Effect ($\hat{\tau}$)	95% Confidence Interval
1	2.606	[2.273 , 2.932]
2	2.577	[2.252 , 2.901]
3	2.547	[2.223 , 2.872]
4	2.488	[2.160 , 2.817]
5	2.459	[2.127 , 2.791]
6	2.430	[2.093 , 2.767]
7	2.400	[2.057 , 2.744]
8	2.371	[2.021 , 2.721]

adding a profit margin m and a mailing cost c to (3.2) (our NDA with the firm forbids revealing m and c). We consider three different counterfactual policies $s(\boldsymbol{x})$: (i) *never* treat, $s(\boldsymbol{x}) \equiv 0$; (ii) a *blanket* treatment, $s(\boldsymbol{x}) \equiv 1$; (iii) a *loyalty* policy, $s(\boldsymbol{x}_i) = 1$ only for those who had purchased in the prior calendar year. Results are shown in Table 4. It is clear that profits from the three policies are ordered as $\pi(\text{never}) < \pi(\text{blanket}) < \pi(\text{loyalty})$.

For both the average effects of Table 3 and the counterfactuals of Table 4 there is broad agree-

Table 4: Counterfactual Profits from Three Targeting Strategies

	Never Treat		Blanket Treatment		Loyalty Policy	
Architecture	$\hat{\pi}(s)$	95% CI	$\hat{\pi}(s)$	95% CI	$\hat{\pi}(s)$	95% CI
1	2.016	[1.923 , 2.110]	2.234	[2.162 , 2.306]	2.367	[2.292 , 2.443]
2	2.022	[1.929 , 2.114]	2.229	[2.157 , 2.301]	2.363	[2.288 , 2.438]
3	2.027	[1.934 , 2.120]	2.224	[2.152 , 2.296]	2.358	[2.283 , 2.434]
4	2.037	[1.944 , 2.130]	2.213	[2.140 , 2.286]	2.350	[2.274 , 2.425]
5	2.043	[1.950 , 2.136]	2.208	[2.135 , 2.281]	2.345	[2.269 , 2.422]
6	2.048	[1.954 , 2.142]	2.202	[2.128 , 2.277]	2.341	[2.263 , 2.418]
7	2.053	[1.959 , 2.148]	2.197	[2.122 , 2.272]	2.336	[2.258 , 2.414]
8	2.059	[1.963 , 2.154]	2.192	[2.116 , 2.268]	2.332	[2.253 , 2.411]

ment among the eight architectures both numerically and substantially. This may be due to the fact that the data is experimental, so that the propensity score is constant. In true observational data this may not be the case. We explore this issue in our Monte Carlo analysis below.

5.2.1 Placebo Experiment

We conducted a set of placebo tests to examine whether the deep neural networks we use can truly recover causal effects. In particular, we take only the untreated customers in the data and randomly assign half to treated status.[2] We then ran the eight architectures of Table 2, as in the true data. The conditional average treatment effects across the architectures are plotted in Figure 5. We see that the "true" zero average effect is recovered precisely and with the expected distribution. The average treatment effect across all models is estimated to be around -0.024, compared to 2.56 in the original data. Exercises with different proportions of (placebo) treated customers revealed similar results.

5.2.2 Optimal Targeting

To explore further, we focus on architecture #3 and study subpopulation treatment targeting strategies following the ideas of Section 3.3. (The other architectures yield similar results, so we omit them.) Architecture #3 has depth $L = 2$ with widths $H_1 = 30$ and $H_2 = 20$. The learning rate was set at 0.0001 and the specification had a total of 4,952 parameters. For this architecture, recalling Remark 1, we added dropout for the second layer with a fixed probability
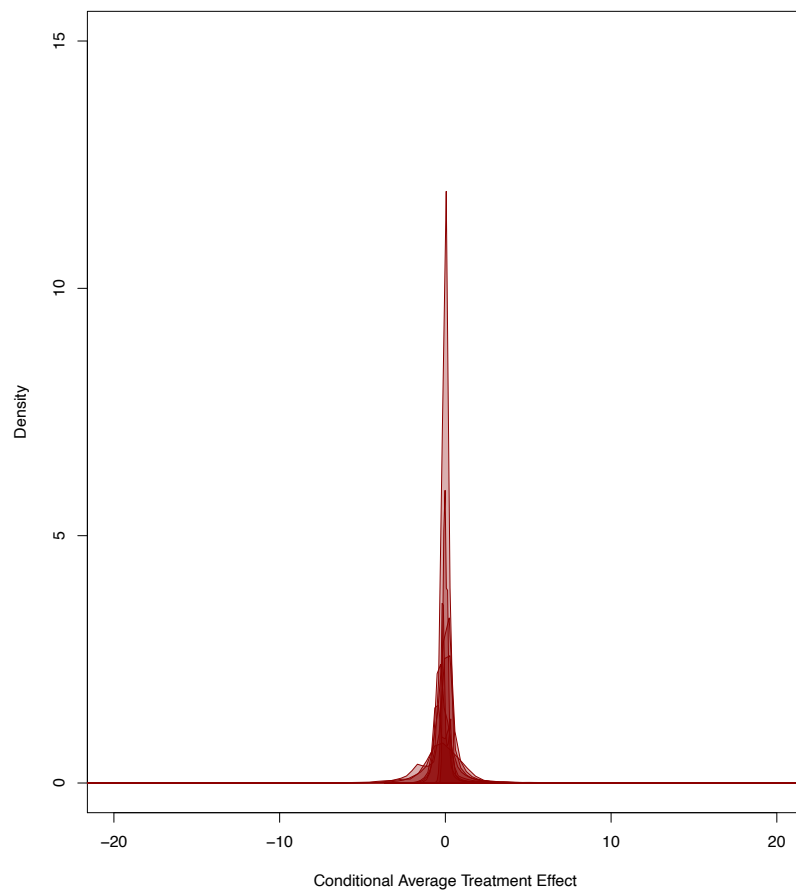

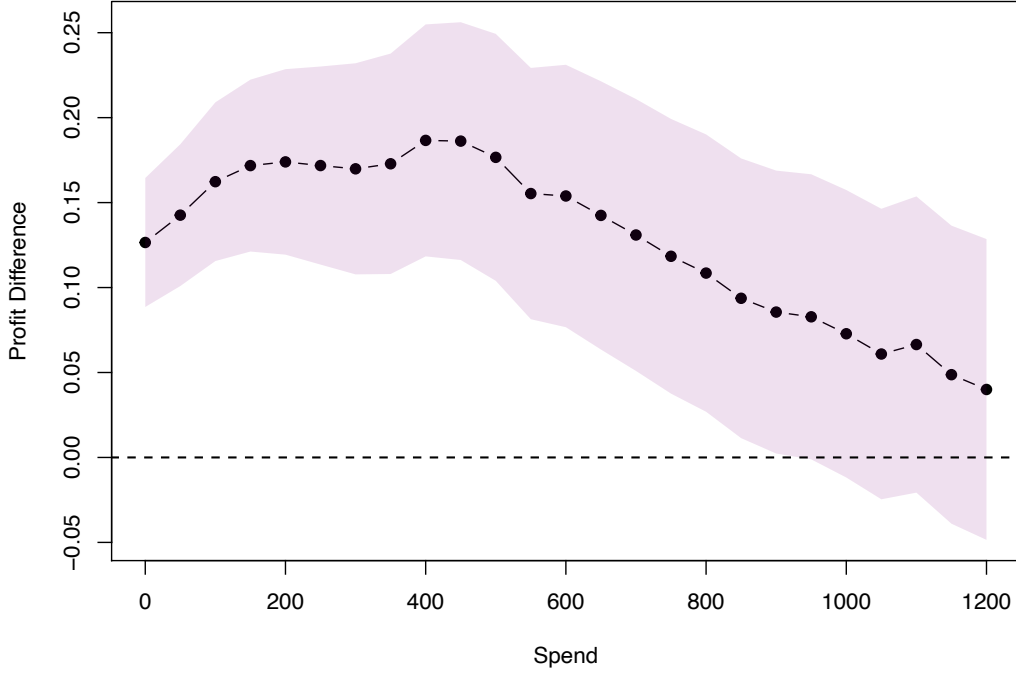[2]We thank Guido Imbens suggesting this analysis.

Figure 5: Placebo Test

Figure 6: Expected Profits from Threshold Targeting Based on Prior Year Spend

of $1/2$. Using this architecture, we compare the blanket strategy (so $s_0(\boldsymbol{x}) = 1$) to targeting customers with spend of at least $\bar{y}$ dollars in the prior calendar year (prior spending is one of the covariates), in \$50 increments to \$1200. The policy class is therefore $\mathcal{S} = \{s(\boldsymbol{x}) = \mathbb{1}(\texttt{prior spend} > \bar{y}), \bar{y} = 0, 50, 100, \ldots, 1150, 1200\}$. Figure 6 presents the results. The black dots show the difference $\{\hat{\pi}(\text{spend} > \bar{y}) - \hat{\pi}(\text{blanket})\}$ and the shaded region gives a *pointwise* 95% confidence band (to ease presentation, sample splitting is not used). We see that there is a significant difference between various choices of $\bar{y}$. Initially, targeting customers with higher spend yields higher profits, as would be expected, but this effect diminishes beyond a certain $\bar{y}$, roughly \$500, as fewer and fewer are targeted. The optimal policy estimate is $\hat{s}(\boldsymbol{x}) = \mathbb{1}(\texttt{prior spend} > 400)$. In general, simpler policy classes may yield better decisions, but it is certainly possible to expand our search to different $\mathcal{S}$ by considering further covariates and/or transformations.

6   Monte Carlo Analysis

We conducted a set of Monte Carlo experiments to evaluate our theoretical results. We study inference on the average treatment effect, τ of (3.1), under different data generating processes (DGPs). In each DGP we take $n = 10,000$ i.i.d. samples and use 1,000 replications. For either $d = 20$ or 100, $\boldsymbol{X}$ includes a constant term and d independent uniform random variables, $\mathcal{U}(0,1)$. Treatment assignment is Bernoulli with probability $p(\boldsymbol{x})$, where $p(\boldsymbol{x})$ is the propensity score. We consider both (i) randomized treatments with $p(\boldsymbol{x}) = 0.5$ and (ii) observational data with $p(\boldsymbol{x}) = (1 + \exp(-\boldsymbol{\alpha}_p'\boldsymbol{x}))^{-1}$, where $\alpha_{p,1} = 0.09$ and the remainder are drawn once as $\mathcal{U}(-0.55, 0.55)$, and then fixed for the replications. For $d = 100$, we maintain $\|\boldsymbol{\alpha}_p\|_0 = 20$ for the simplicity. These generate propensities with an approximate range of approximately $(0.30, 0.75)$ and mean roughly 0.5.

Given covariates and treatment assignment, the outcomes are generated according to

$$y_i = \mu_0(\boldsymbol{x}_i) + \tau(\boldsymbol{x}_i)t_i + \varepsilon_i, \qquad \mu_0(\boldsymbol{x}) = \boldsymbol{\alpha}_\mu'\boldsymbol{x} + \boldsymbol{\beta}_\mu'\varphi(\boldsymbol{x}), \qquad \tau(x_i) = \boldsymbol{\alpha}_\tau'\boldsymbol{x} + \boldsymbol{\beta}_\tau'\varphi(\boldsymbol{x}),$$

where $\varepsilon_i \sim \mathcal{N}(0,1)$ and $\varphi(\boldsymbol{x})$ are second-degree polynomials including pairwise interactions. For $\mu_0(\boldsymbol{x})$ and $\tau(\boldsymbol{x})$ we consider two cases, linear and nonlinear models. In both cases the intercepts are $\alpha_{\mu,1} = 0.09$ and $\alpha_{\tau,1} = -0.05$ and slopes are drawn (once) as $\alpha_{\mu,k} \sim \mathcal{N}(0.3, 0.7)$ and $\alpha_{\tau,k} \sim \mathcal{U}(0.1, 0.22)$, $k = 2, \ldots, d+1$. The linear models set $\boldsymbol{\beta}_\mu = \boldsymbol{\beta}_\tau = \boldsymbol{0}$ while the nonlinear models take $\beta_{\mu,k} \sim \mathcal{N}(0.01, 0.3)$ and $\beta_{\tau,k} \sim \mathcal{U}(-0.05, 0.06)$. Altogether, this yields eight designs: $d = 20$ or 100, $p(\boldsymbol{x})$ constant or not, and outcome models linear or nonlinear.

For each design, we consider a variety of network architectures, all ReLU-based MLPs. These architectures are variants of the ones used in the empirical application (which were customized for the application). All networks vary in their depth and width, as spelled out in Table 5.

Tables 6 and 7 show the results for all eight DGPs. Table 6 shows randomized treatment while Table 7 shows results mimicking observational data. Overall, the results reported show excellent performance of deep learning based semiparametric inference. The bias is minimal and the coverage is quite accurate, while the interval length is under control. Notice that the most architectures yield similar results with no architecture dominating the others. Further, the coverage and interval

Table 5: Monte Carlo Architectures Explored

Architecture	Structure
1	{20, 15, 5}
2	{60, 30, 20}
3	{80, 80, 80}
4	{20, 15, 10, 5}
5	{60, 30, 20, 10}
6	{80, 80, 80, 80}
7	{20, 15, 15, 10, 10, 5}
8	{60, 30, 20, 20, 10, 5}
9	{80, 80, 80, 80, 80, 80}

length are fairly similar with the more complex architecture not exhibiting any systematic patterns of length inflation.

None of the architectures we presented earlier used regularization. In typical empirical applications, including our own, researchers adopt architectures that employ dropout, a common method of regularization; see Remark 1. Our own preliminary exploration of dropout and other forms of regularization found expected departures form nonregularized models. In most, but not all, cases the coverage remained accurate, but with increased bias and interval length compared to Table 6 and 7. The results preach caution when applying regularization in applications.

7 Conclusion

The utility of deep learning in social science applications is still a subject of interest and debate. While there is an acknowledgment of its predictive power, there has been limited adoption of deep learning in social sciences such as economics. Some part of the reluctance to adopting these methods stems from the lack of theory facilitating use and interpretation. We have shown, both theoretically as well as empirically, that these methods can offer excellent performance.

In this paper, we have given a formal proof that inference can be valid after using deep learning methods for first-step estimation. To the best of our knowledge, ours is the first inference result using deep nets. Our results thus contribute directly to the recent explosion in both theoretical and applied research using machine learning methods in economics, and to the recent adoption of deep learning in empirical settings. We obtained novel bounds for deep neural networks, speaking directly to the modern (and empirically successful) practice of using fully-connected feedfoward

Table 6: Simulations Results - Constant Propensity Score

Model	Architecture	20 Covariates			100 Covariates		
		Bias	IL	Coverage	Bias	IL	Coverage
	1	0.00027	0.079	0.947	0.00067	0.080	0.946
	2	-0.00032	0.079	0.951	0.00012	0.080	0.958
	3	-0.00025	0.079	0.955	-0.00167	0.080	0.939
	4	-0.00068	0.079	0.949	0.00038	0.080	0.949
Linear	5	0.00008	0.079	0.945	-0.00219	0.080	0.929
	6	0.00007	0.079	0.955	-0.00010	0.080	0.946
	7	0.00128	0.079	0.952	-0.00041	0.080	0.944
	8	0.00108	0.079	0.949	-0.00088	0.080	0.941
	9	0.00021	0.078	0.948	-0.00080	0.081	0.953
	1	0.00087	0.081	0.946	-0.00067	0.163	0.940
	2	0.00015	0.079	0.954	0.00093	0.153	0.927
	3	-0.00072	0.079	0.940	0.00245	0.148	0.926
	4	0.00101	0.080	0.945	-0.00087	0.165	0.956
Nonlinear	5	0.00027	0.079	0.935	-0.00190	0.154	0.923
	6	-0.00025	0.079	0.929	-0.00117	0.146	0.902
	7	-0.00052	0.080	0.947	0.00091	0.165	0.941
	8	0.00077	0.079	0.938	0.00201	0.153	0.927
	9	-0.00013	0.079	0.940	0.00049	0.154	0.936

networks. Our results allow for different network architectures, including fixed width, very deep networks. Our results cover general nonparametric regression-type loss functions, covering most nonparametric practice. We used our bounds to deliver fast convergence rates allowing for second-stage inference on a finite-dimensional parameter of interest.

There are practical implications of the theory presented in this paper. We focused on semiparametric causal effects as a concrete illustration, but deep learning is a potentially valuable tool in many diverse economic settings. Our results allow researchers to embed deep learning into standard econometric models such as linear regressions, generalized linear models, and other forms of limited dependent variables models (e.g. censored regression). Our theory can also be used as a starting point for constructing deep learning implementations of two-step estimators in the context of selection models, dynamic discrete choice, and the estimation of games.

Table 7: Simulations Results - Non-constant Propensity Score

Model	Architecture	20 Covariates			100 Covariates		
		Bias	IL	Coverage	Bias	IL	Coverage
	1	-0.00202	0.080	0.948	0.0009	0.081	0.955
	2	0.00011	0.079	0.946	0.0007	0.081	0.945
	3	-0.00130	0.079	0.964	-0.0001	0.081	0.937
	4	-0.00106	0.079	0.945	0.0002	0.081	0.933
Linear	5	-0.00083	0.079	0.951	-0.0004	0.081	0.944
	6	-0.00068	0.079	0.955	0.0001	0.081	0.924
	7	-0.00119	0.079	0.953	-0.0001	0.081	0.942
	8	-0.00056	0.079	0.952	-0.0008	0.081	0.939
	9	-0.00096	0.079	0.948	-0.0007	0.081	0.952
	1	-0.00076	0.081	0.946	-0.00279	0.164	0.937
	2	-0.00122	0.080	0.939	0.00020	0.155	0.941
	3	-0.00074	0.080	0.926	-0.00080	0.148	0.914
	4	-0.00171	0.081	0.940	-0.00184	0.166	0.938
Linear	5	-0.00135	0.080	0.952	-0.00103	0.154	0.912
	6	-0.00075	0.080	0.950	-0.00174	0.147	0.905
	7	-0.00153	0.081	0.928	-0.00377	0.165	0.929
	8	0.00082	0.080	0.953	0.00031	0.154	0.919
	9	-0.00127	0.080	0.931	-0.00094	0.156	0.917

To be clear, we see our paper as a first step in the exploration of deep learning as a tool for economic applications. There are a number of opportunities, questions, and challenges that remain. For example, factor models in finance might benefit from the use of auto-encoders and recurrent neural nets may have applications in time series. For some estimands, it may be crucial to estimate the density as well, and this problem can be challenging in high dimensions. Deep nets, in the form of GANs are a promising tool for distribution estimation. There are also interesting questions remaining as to an optimal network architecture, and if this can be itself learned from the data, as well as computational and optimization guidance. Research into these further applications and structures is underway.

8 References

ABADIE, A. AND M. D. CATTANEO (2018): "Econometric Methods for Program Evaluation," *Annual Review of Economics*, 10, 465–503.

ANTHONY, M. AND P. L. BARTLETT (1999): *Neural Network Learning: Theoretical Foundations*, Campbridge University Press.

ATHEY, S., G. IMBENS, T. PHAM, AND S. WAGER (2017): "Estimating average treatment effects: Supplementary analyses and remaining challenges," *American Economic Review: Papers & Proceeding*, 107, 278–81.

ATHEY, S., G. W. IMBENS, AND S. WAGER (2018): "Approximate residual balancing: debiased inference of average treatment effects in high dimensions," *Journal of the Royal Statistical Society, Series B*, 80, 597–623.

ATHEY, S. AND S. WAGER (2018): "Efficient Policy Learning," *arXiv preprint arXiv:1702.02896*.

BARRON, A. R. (1993): "Universal approximation bounds for superpositions of a sigmoidal function," *IEEE Transactions on Information theory*, 39, 930–945.

BARTLETT, P. L., O. BOUSQUET, AND S. MENDELSON (2005): "Local rademacher complexities," *The Annals of Statistics*, 33, 1497–1537.

BARTLETT, P. L., N. HARVEY, C. LIAW, AND A. MEHRABIAN (2017): "Nearly-tight VC-dimension bounds for piecewise linear neural networks," in *Proceedings of the 22nd Annual Conference on Learning Theory (COLT 2017)*.

BAUER, B. AND M. KOHLER (2017): "On Deep Learning as a remedy for the curse of dimensionality in nonparametric regression," Tech. rep., Technical report.

BELLONI, A., D. CHEN, V. CHERNOZHUKOV, AND C. HANSEN (2012): "Sparse models and methods for optimal instruments with an application to eminent domain," *Econometrica*, 80, 2369–2429.

BELLONI, A., V. CHERNOZHUKOV, D. CHETVERIKOV, C. HANSEN, AND K. KATO (2018): "High-Dimensional Econometrics and Generalized GMM," *arXiv preprint arXiv:1806.01888*.

BELLONI, A., V. CHERNOZHUKOV, I. FERNÁNDEZ-VAL, AND C. HANSEN (2017): "Program Evaluation and Causal Inference With High-Dimensional Data," *Econometrica*, 85, 233–298.

BELLONI, A., V. CHERNOZHUKOV, AND C. HANSEN (2014): "Inference on Treatment Effects after Selection Amongst High-Dimensional Controls," *Review of Economic Studies*, 81, 608–650.

BELLONI, A., V. CHERNOZHUKOV, AND L. WANG (2011): "Square-root lasso: pivotal recovery of sparse signals via conic programming," *Biometrika*, 98, 791–806.

BICKEL, P. J. AND Y. RITOV (1988): "Estimating Integrated Squared Density Derivatives: Sharp Best Order of Convergence Estimates," *Sankhyā*, 50, 381–393.

BICKEL, P. J., Y. RITOV, AND A. B. TSYBAKOV (2009): "Simultaneous Analysis of LASSO and Dantzig Selector," *The Annals of Statistics*, 37, 1705–1732.

BLINDER, A. (1973): "Wage Discrimination: Reduced Form and Structural Estimates," *Journal of Human Resources*, 8, 436–455.

CATTANEO, M. D. (2010): "Efficient Semiparametric Estimation of Multi-valued Treatment Effects under Ignorability," *Journal of Econometrics*, 155, 138–154.

CATTANEO, M. D. AND M. JANSSON (2018): "Kernel-Based Semiparametric Estimators: Small Bandwidth Asymptotics and Bootstrap Consistency," *Econometrica*, 86, 955–995.

CATTANEO, M. D., M. JANSSON, AND X. MA (2018): "Two-step Estimation and Inference with Possibly Many Included Covariates," *arXiv:1807.10100, Review of Economic Studies*, forthcoming.

CHEN, X. (2007): "Large Sample Sieve Estimation of Semi-Nonparametric Models," in *Handbook of Econometrics*, ed. by J. Heckman and E. Leamer, Elsevier, vol. 6B of *Handbook of Econometrics*, chap. 76.

CHEN, X., H. HONG, AND A. TAROZZI (2004): "Semiparametric Efficiency in GMM Models of Nonclassical Measurament Errors, Missing Data and Treatment Effects," *Cowles Foundation Discussion Paper No. 1644*.

——— (2008): "Semiparametric Efficiency in GMM Models With Auxiliary Data," *The Annals of Statistics*, 36, 808–843.

CHEN, X. AND X. SHEN (1998): "Sieve extremum estimates for weakly dependent data," *Econometrica*, 66, 289–314.

CHEN, X. AND H. WHITE (1999): "Improved rates and asymptotic normality for nonparametric neural network estimators," *IEEE Transactions on Information Theory*, 45, 682–691.

CHERNOZHUKOV, V., D. CHETVERIKOV, M. DEMIRER, E. DUFLO, C. HANSEN, W. NEWEY, AND J. ROBINS (2018a): "Double/debiased machine learning for treatment and structural parameters," *The Econometrics Journal*, 21, C1–C68.

CHERNOZHUKOV, V., M. DEMIRER, E. DUFLO, AND I. FERNANDEZ-VAL (2018b): "Generic Machine Learning Inference on Heterogenous Treatment Effects in Randomized Experiments," *arXiv preprint arXiv:1712.04802*.

CHERNOZHUKOV, V., J. C. ESCANCIANO, H. ICHIMURA, W. K. NEWEY, AND J. M. ROBINS (2018c): "Locally Robust Semiparametric Estimation," *arXiv:1608.00033*.

DANIELY, A. (2017): "Depth separation for neural networks," *arXiv preprint arXiv:1702.08489*.

FARRELL, M. H. (2015): "Robust Inference on Average Treatment Effects with Possibly More Covariates than Observations," *arXiv:1309.4686, Journal of Econometrics*, 189, 1–23.

FORTIN, N., T. LEMIEUX, AND S. FIRPO (2011): "Decomposition Methods in Economics," vol. 4 of *Handbook of Labor Economics*, 1–102.

GINE, E. AND R. NICKL (2016): *Mathematical Foundations of Infinite-Dimensional Models*, Cambridge.

GOODFELLOW, I., Y. BENGIO, AND A. COURVILLE (2016): *Deep learning*, Cambridge: MIT Press.

HAHN, J. (1998): "On the Role of the Propensity Score in Efficient Semiparametric Estimation of Average Treatment Effects," *Econometrica*, 66, 315–331.

——— (2004): "Functional restriction and efficiency in causal inference," *Review of Economics and Statistics*, 84, 73–76.

HANIN, B. (2017): "Universal function approximation by deep neural nets with bounded width and relu activations," *arXiv preprint arXiv:1708.02691*.

HANSEN, C., D. KOZBUR, AND S. MISRA (2017): "Targeted Undersmoothing," *arXiv:1706.07328*.

HARTFORD, J., G. LEWIS, K. LEYTON-BROWN, AND M. TADDY (2017): "Deep iv: A flexible approach for counterfactual prediction," in *International Conference on Machine Learning*, 1414–1423.

HASTIE, T., R. TIBSHIRANI, AND J. FRIEDMAN (2009): *The elements of statistical learning*, Springer Series in Statistics, New York: Springer-Verlag.

HE, K., X. ZHANG, S. REN, AND J. SUN (2016): "Identity mappings in deep residual networks," in *European conference on computer vision*, Springer, 630–645.

HECKMAN, J. AND E. J. VYTLACIL (2007): "Econometric Evaluation of Social Programs, Part I," in *Handbook of Econometrics, vol. VIB*, ed. by J. Heckman and E. Leamer, Elsevier Science B.V., 4780–4874.

HIRANO, K. AND J. PORTER (2009): "Asymptotics for statistical treatment rules," *Econometrica*, 77, 1683–1701.

HITSCH, G. J. AND S. MISRA (2018): "Heterogeneous Treatment Effects and Optimal Targeting Policy Evaluation," *SSRN preprint 3111957*.

HORNIK, K., M. STINCHCOMBE, AND H. WHITE (1989): "Multilayer feedforward networks are universal approximators," *Neural networks*, 2, 359–366.

IMBENS, G. W. AND D. B. RUBIN (2015): *Causal Inference in Statistics, Social, and Biomedical Sciences*, Cambridge University Press.

IMBENS, G. W. AND J. M. WOOLDRIDGE (2009): "Recent Developments in the Econometrics of Program Evaluation," *Journal of Economic Literature*, 47, 5–86.

JAVANMARD, A. AND A. MONTANARI (2014): "Confidence intervals and hypothesis testing for high-dimensional regression," *The Journal of Machine Learning Research*, 15, 2869–2909.

JOHANSSON, F., U. SHALIT, AND D. SONTAG (2016): "Learning representations for counterfactual inference," in *International Conference on Machine Learning*, 3020–3029.

KINGMA, D. P. AND J. BA (2014): "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*.

KITAGAWA, E. M. (1955): "Components of a Difference Between Two Rates," *Journal of the American Statistical Association*, 50, 1168–1194.

KITAGAWA, T. AND A. TETENOV (2018): "Who should be treated? empirical welfare maximization methods for treatment choice," *Econometrica*, 86, 591–616.

KOLTCHINSKII, V. (2006): "Local Rademacher complexities and oracle inequalities in risk mini- mization," *The Annals of Statistics*, 34, 2593–2656.

——— (2011): *Oracle Inequalities in Empirical Risk Minimization and Sparse Recovery Problems*, Springer-Verlag.

KOLTCHINSKII, V. AND D. PANCHENKO (2000): "Rademacher processes and bounding the risk of function learning," in *High dimensional probability II*, Springer, 443–457.

KRIZHEVSKY, A., I. SUTSKEVER, AND G. E. HINTON (2012): "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 1097– 1105.

LECUN, Y., L. BOTTOU, Y. BENGIO, AND P. HAFFNER (1998): "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, 86, 2278–2324.

LIANG, T. (2018): "On How Well Generative Adversarial Networks Learn Densities: Nonparamet- ric and Parametric Results," *arXiv:1811.03179*.

LIANG, T., A. RAKHLIN, AND K. SRIDHARAN (2015): "Learning with square loss: Localization through offset Rademacher complexity," in *Conference on Learning Theory*, 1260–1285.

LIU, X., D. LEE, AND K. SRINIVASAN (2017): "Large scale cross category analysis of consumer review content on sales conversion leveraging deep learning," *working paper, NYU Stern*.

MA, X. AND J. WANG (2018): "Robust Inference Using Inverse Probability Weighting," *arXiv preprint arXiv:1810.11397*.

MAKOVOZ, Y. (1996): "Random approximants and neural networks," *Journal of Approximation Theory*, 85, 98–109.

MANSKI, C. F. (2004): "Statistical treatment rules for heterogeneous populations," *Econometrica*, 72, 1221–1246.

MENDELSON, S. (2003): "A few notes on statistical learning theory," in *Advanced lectures on machine learning*, Springer, 1–40.

——— (2014): "Learning without concentration," in *Conference on Learning Theory*, 25–39.

MHASKAR, H. AND T. POGGIO (2016a): "Deep vs. shallow networks: An approximation theory perspective," *arXiv preprint arXiv:1608.03287*.

MHASKAR, H. N. AND T. POGGIO (2016b): "Deep vs. shallow networks: An approximation theory perspective," *Analysis and Applications*, 14, 829–848.

NAIR, V. AND G. E. HINTON (2010): "Rectified linear units improve restricted boltzmann ma- chines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 807–814.

NEWEY, W. K. AND J. M. ROBINS (2018): "Cross-fitting and fast remainder rates for semipara- metric estimation," *arXiv preprint arXiv:1801.09138*.

OAXACA, R. (1973): "Male-Female Wage Differentials in Urban Labor Markets," *International Economic Review*, 14, 693–709.

PISIER, G. (1981): "Remarques sur un résultat non publié de B. Maurey," in *Séminaire Analyse fonctionnelle (dit" Maurey-Schwartz")*, 1–12.

POLSON, N. AND V. ROCKOVA (2018): "Posterior Concentration for Sparse Deep Learning," *arXiv preprint arXiv:1803.09138*.

RAGHU, M., B. POOLE, J. KLEINBERG, S. GANGULI, AND J. SOHL-DICKSTEIN (2017): "On the Expressive Power of Deep Neural Networks," in *Proceedings of the 34th International Conference on Machine Learning*, ed. by D. Precup and Y. W. Teh, International Convention Centre, Sydney, Australia: PMLR, vol. 70 of *Proceedings of Machine Learning Research*, 2847–2854.

ROBINS, J., L. LI, R. MUKHERJEE, E. TCHETGEN, AND A. VAN DER VAART (2017): "Minimax Estimation of a Functional on a Structured High-Dimensional Model," *The Annals of Statistics*, 45, 1951–1987.

ROBINS, J., L. LI, E. TCHETGEN, AND A. VAN DER VAART (2008): "Higher order influence functions and minimax estimation of nonlinear functionals," in *Probability and Statistics: Essays in Honor of David A. Freedman*, ed. by D. Nolan and T. Speed, Beachwood, Ohio, USA: Institute of Mathematical Statistics, vol. 2.

ROBINS, J., E. T. TCHETGEN, L. LI, AND A. VAN DER VAART (2009): "Semiparametric Minimax Rates," *Electronic Journal of Statistics*, 3, 1305–1321.

ROBINS, J. M., A. ROTNITZKY, AND L. ZHAO (1994): "Estimation of Regression Coefficients When Some Regressors Are Not Always Observed," *Journal of the American Statistical Association*, 89, 846–866.

——— (1995): "Analysis of Semiparametric Regression Models for Repeated Outcomes in the Presence of Missing Data," *Journal of the American Statistical Association*, 90, 846–866.

ROMANO, J. P. (2004): "On non-parametric testing, the uniform behaviour of the *t*-test, and related problems," *Scandinavian Journal of Statistics*, 31, 567–584.

SAFRAN, I. AND O. SHAMIR (2016): "Depth separation in relu networks for approximating smooth non-linear functions," *arXiv preprint arXiv:1610.09887*.

SCHMIDT-HIEBER, J. (2017): "Nonparametric regression using deep neural networks with ReLU activation function," *arXiv preprint arXiv:1708.06633*.

SHALIT, U., F. D. JOHANSSON, AND D. SONTAG (2017): "Estimating individual treatment effect: generalization bounds and algorithms," *arXiv preprint arXiv:1606.03976*.

SLOCZYNSKI, T. AND J. M. WOOLDRIDGE (2018): "A General Double Robustness Result for Estimating Average Treatment Effects," *Econometric Theory*, 34, 112–133.

STONE, C. J. (1982): "Optimal global rates of convergence for nonparametric regression," *The annals of statistics*, 1040–1053.

TADDY, M., M. GARDNER, L. CHEN, AND D. DRAPER (2015): "A nonparametric Bayesian analysis of heterogeneous treatment effects in digital experimentation," *Arxiv preprint arXiv:1412.8563*.

TAN, Z. (2018): "Model-assisted inference for treatment effects using regularized calibrated estimation with high-dimensional data," *arXiv preprint arXiv:1801.09817*.

TELGARSKY, M. (2016): "Benefits of depth in neural networks," *arXiv preprint arXiv:1602.04485*.

TSIATIS, A. A. (2006): *Semiparametric Theory and Missing Data*, New York: Springer.

VAN DE GEER, S., P. BUHLMANN, Y. RITOV, AND R. DEZEURE (2014): "On Asymptotically Optimal Confidence Regions and Tests for High-Dimensional Models," *The Annals of Statistics*, 42, 1166–1202.

VAN DER LAAN, M. AND S. ROSE (2001): *Targeted Learning: Causal Inference for Observational and Experimental Data*, Springer-Verlag.

WAGER, S. AND S. ATHEY (2018): "Estimation and Inference of Heterogeneous Treatment Effects using Random Forests," *Journal of the American Statistical Association,* forthcoming.

WESTREICH, D., J. LESSLER, AND M. J. FUNK (2010): "Propensity score estimation: neural networks, support vector machines, decision trees (CART), and meta-classifiers as alternatives to logistic regression," *Journal of clinical epidemiology*, 63, 826–833.

WHITE, H. (1989): "Learning in artificial neural networks: A statistical perspective," *Neural computation*, 1, 425–464.

———— (1992): *Artificial neural networks: approximation and learning theory*, Blackwell Publishers, Inc.

YAROTSKY, D. (2017): "Error bounds for approximations with deep ReLU networks," *Neural Networks*, 94, 103–114.

———— (2018): "Optimal approximation of continuous functions by very deep ReLU networks," *arXiv preprint arXiv:1802.03620*.

ZHANG, C., S. BENGIO, M. HARDT, B. RECHT, AND O. VINYALS (2016): "Understanding deep learning requires rethinking generalization," *arXiv preprint arXiv:1611.03530*.

A   Proofs

In this section we provide a proof of Theorems 1 and 2, our main theoretical results for deep ReLU networks, and their corollaries. The proof proceeds in several steps. We first give the main breakdown and bound the bias (approximation error) term. We then turn our attention to the empirical process term, to which we apply our localization. Much of the proof uses a generic architecture, and thus pertains to both results. We will specialize the architecture to the multi-layer perceptron only when needed later on. Other special cases and related results are covered in Section A.4. Supporting Lemmas are stated in Section B.

The statements of Theorems 1 and 2 assume that n is large enough. Precisely, we require $n > (2eM)^2 \vee \text{Pdim}(\mathcal{F}_{\text{DNN}})$. For notational simplicity we will denote $\widehat{f}_{\text{DNN}} := \hat{f}$, see (2.4), and $\epsilon_{\text{DNN}} := \epsilon_n$, see Assumption 3. As we are simultaneously consider Theorems 1 and 2, the generic notation DNN will be used throughout.

A.1 Main Decomposition and Bias Term

Referring to Assumption 3, define the best approximation realized by the deep ReLU network class $\mathcal{F}_{\text{DNN}}$ as

$$f_n := \underset{\substack{f \in \mathcal{F}_{\text{DNN}} \\ \|f\|_\infty \le 2M}}{\arg\min} \|f - f_*\|_\infty.$$

By definition, $\epsilon_n := \epsilon_{\text{DNN}} := \|f_n - f_*\|_\infty$.

Recalling the optimality of the estimator in (2.4), we know, as both f_n and $\hat{f}$ are in $\mathcal{F}_{\text{DNN}}$, that

$$-\mathbb{E}_n[\ell(\hat{f}, \boldsymbol{z})] + \mathbb{E}_n[\ell(f_n, \boldsymbol{z})] \ge 0.$$

This result does not hold for f_* in place of f_n, because $f_* \notin \mathcal{F}_{\text{DNN}}$. Using the above display and the curvature of Equation (2.1) (which does not hold with f_n in place of f_* therein), we obtain

$$
\begin{aligned}
c_1 \|\hat{f} - f_*\|_{L_2(\boldsymbol{x})}^2 &\le \mathbb{E}[\ell(\hat{f}, \boldsymbol{z})] - \mathbb{E}[\ell(f_*, \boldsymbol{z})] \\
&\le \mathbb{E}[\ell(\hat{f}, \boldsymbol{z})] - \mathbb{E}[\ell(f_*, \boldsymbol{z})] - \mathbb{E}_n[\ell(\hat{f}, \boldsymbol{z})] + \mathbb{E}_n[\ell(f_n, \boldsymbol{z})] \\
&= \mathbb{E}\left[\ell(\hat{f}, \boldsymbol{z}) - \ell(f_*, \boldsymbol{z})\right] - \mathbb{E}_n\left[\ell(\hat{f}, \boldsymbol{z}) - \ell(f_*, \boldsymbol{z})\right] + \mathbb{E}_n\left[\ell(f_n, \boldsymbol{z}) - \ell(f_*, \boldsymbol{z})\right] \\
&= (\mathbb{E} - \mathbb{E}_n)\left[\ell(\hat{f}, \boldsymbol{z}) - \ell(f_*, \boldsymbol{z})\right] + \mathbb{E}_n\left[\ell(f_n, \boldsymbol{z}) - \ell(f_*, \boldsymbol{z})\right]. \quad (\text{A.1})
\end{aligned}
$$

Equation (A.1) is the main decomposition that begins the proof. The decomposition must be done this way because of the above notes regarding f_* and f_n. The first term is the empirical process term that will be treated in the subsequent subsection. For the second term in (A.1), the bias term or approximation error, we apply Bernstein's inequality to find that, with probability at least $1 - e^{-\tilde{\gamma}}$,

$$
\begin{aligned}
\mathbb{E}_n\left[\ell(f_n, \boldsymbol{z}) - \ell(f_*, \boldsymbol{z})\right] &\le \mathbb{E}\left[\ell(f_n, \boldsymbol{z}) - \ell(f_*, \boldsymbol{z})\right] + \sqrt{\frac{2C_\ell^2 \|f_n - f_*\|_\infty^2 \tilde{\gamma}}{n}} + \frac{21 C_\ell M \tilde{\gamma}}{3n} \\
&\le c_2 \mathbb{E}\left[\|f_n - f_*\|^2\right] + \sqrt{\frac{2C_\ell^2 \|f_n - f_*\|_\infty^2 \tilde{\gamma}}{n}} + \frac{7 C_\ell M \tilde{\gamma}}{n} \\
&\le c_2 \epsilon_n^2 + \epsilon_n \sqrt{\frac{2C_\ell^2 \tilde{\gamma}}{n}} + \frac{7 C_\ell M \tilde{\gamma}}{n}, \quad (\text{A.2})
\end{aligned}
$$

using the Lipschitz and curvature of the loss function defined in Equation (2.1) and $\mathbb{E}\left[\|f_n - f_*\|^2\right] \le \|f_n - f_*\|_\infty^2$, along with the definition of ϵ_n^2.

Once the empirical process term is controlled (in Section A.2), the two bounds will be brought back together to compute the final result, see Section A.3.

A.2 Localization Analysis

We now turn to bounding the first term in (A.1) (the empirical processes term) using a localized analysis that derives bounds based on scale insensitive complexity measure. The ideas of our localization are rooted in Koltchinskii and Panchenko (2000) and Bartlett et al. (2005), and related to Koltchinskii (2011). Localization analysis extending to the unbounded f case has been developed in Mendelson (2014); Liang et al. (2015). This proof section proceeds in several steps.

A key quantity is the Rademacher complexity of the function class at hand. Given i.i.d. Rademacher draws, $\eta_i = \pm 1$ with equal probability independent of the data, the random variable $R_n \mathcal{F}$, for a function class $\mathcal{F}$, is defined as

$$R_n \mathcal{F} := \sup_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} \eta_i f(\boldsymbol{x}_i).$$

Intuitively, $R_n \mathcal{F}$ measures how flexible the function class is for predicting random signs. Taking the expectation of $R_n \mathcal{F}$ conditioned on the data we obtain the *empirical Rademacher complexity*, denoted $\mathbb{E}_\eta[R_n \mathcal{F}]$. When the expectation is taken over both the data and the draws η_i, $\mathbb{E}R_n \mathcal{F}$, we get the *Rademacher complexity*.

A.2.1 Step I: Quadratic Process

The first step is to show that, with high probability, the empirical L_2 norm of the error $(f - f_*)$ is at most twice the population L_2 norm bound for the same error, for certain functions f outside a certain critical radius. This will be an ingredient to be used later on. Denote $\|f\|_n := \left(\frac{1}{n} \sum_{i=1}^{n} f(\boldsymbol{x}_i)^2\right)^{1/2}$ to be the empirical L_2 norm. To do so, we study the quadratic process

$$\|f - f_*\|_n^2 - \|f - f_*\|_{L_2(\boldsymbol{x})}^2 = \mathbb{E}_n(f - f_*)^2 - \mathbb{E}(f - f_*)^2.$$

We will apply the symmetrization of Lemma 5 to $g = (f - f_*)^2$ restricted to a radius $\|f - f_*\|_{L_2(\boldsymbol{x})} \leq r$. This function g has variance bounded as

$$\mathbb{V}[g] \leq \mathbb{E}[g^2] \leq \mathbb{E}((f - f_*)^4) \leq 9M^2 r^2.$$

Writing $g = (f + f_*)(f - f_*)$, we see that by Assumption 1, $|g| \leq 3M|f - f_*| \leq 9M^2$, where the first inequality verifies that g has a Lipschitz constant of $3M$ (when viewed as a function of its argument f), and second that g itself is bounded. We therefore apply Lemma 5, to obtain, with probability at least $1 - \exp(-\tilde{\gamma})$, that for any $f \in \mathcal{F}$ with $\|f - f_*\|_{L_2(\boldsymbol{x})} \leq r$,

$$\mathbb{E}_n(f - f_*)^2 - \mathbb{E}(f - f_*)^2$$

$$\leq 3\mathbb{E}R_n\{g = (f - f_*)^2 : f \in \mathcal{F}, \|f - f_*\|_{L_2(\boldsymbol{x})} \leq r\} + 3Mr\sqrt{\frac{2\tilde{\gamma}}{n}} + \frac{36M^2}{3}\frac{\tilde{\gamma}}{n}$$

$$\leq 18M\mathbb{E}R_n\{f - f_* : f \in \mathcal{F}, \|f - f_*\|_{L_2(\boldsymbol{x})} \leq r\} + 3Mr\sqrt{\frac{2\tilde{\gamma}}{n}} + \frac{12M^2\tilde{\gamma}}{n}, \qquad (A.3)$$

where the second inequality applies Lemma 2 to the Lipschitz functions $\{g\}$ (as a function of the real values $f(\boldsymbol{x})$) and iterated expectations.

Suppose the radius r satisfies

$$r^2 \geq 18M\mathbb{E}R_n\{f - f_* : f \in \mathcal{F}, \|f - f_*\|_{L_2(\boldsymbol{x})} \leq r\} \tag{A.4}$$

and

$$r^2 \geq \frac{6\sqrt{6}M^2\tilde{\gamma}}{n}. \tag{A.5}$$

Then we conclude from from (A.3) that

$$\mathbb{E}_n(f - f_*)^2 \leq r^2 + r^2 + 3Mr\sqrt{\frac{2\tilde{\gamma}}{n}} + \frac{12M^2\tilde{\gamma}}{n} \leq (2r)^2 \tag{A.6}$$

where the first inequality uses (A.4) and the second line uses (A.5). This means that for r above the "critical radius" (see **Step III**), the empirical L_2-norm is at most twice the population one with probability at least $1 - \exp(-\tilde{\gamma})$.

A.2.2 Step II: One Step Improvement

In this step we will show that given a bound on $\|\hat{f} - f_*\|_{L_2(\boldsymbol{x})}$ we can use this bound as information to obtain a tighter bound, if the initial bound is loose as made precise at the end of this step. This tightening will then be pursued to its limit in **Step III**, which leads to the final rate obtained in **Step IV**. **Step I** will be used herein.

Suppose we know that for some r_0, $\|\hat{f} - f_*\|_{L_2(\boldsymbol{x})} \leq r_0$. We may always start with $r_0 = 3M$ given Assumption 1 and (2.4). Apply Lemma 5 with $\mathcal{G} := \{g = \ell(f, \boldsymbol{z}) - \ell(f_*, \boldsymbol{z}) : f \in \mathcal{F}_{\text{DNN}}, \|f - f_*\|_{L_2(\boldsymbol{x})} \leq r_0\}$, we find that, with probability at least $1 - 2e^{-\tilde{\gamma}}$, the empirical process term of (A.1) is bounded as

$$(\mathbb{E} - \mathbb{E}_n)\left[\ell(\hat{f}, \boldsymbol{z}) - \ell(f_*, \boldsymbol{z})\right] \leq 6\mathbb{E}_\eta R_n\mathcal{G} + \sqrt{\frac{2C_\ell^2 r_0^2\tilde{\gamma}}{n}} + \frac{23 \cdot 3MC_\ell}{3}\frac{\tilde{\gamma}}{n}, \tag{A.7}$$

where the middle term is due to the following variance calculation (recall Equation (2.1))

$$\mathbb{V}[g] \leq \mathbb{E}[g^2] = \mathbb{E}[|\ell(f, \boldsymbol{z}) - \ell(f_*, \boldsymbol{z})|^2] \leq C_\ell^2\mathbb{E}(f - f_*)^2 \leq C_\ell^2 r_0^2$$

Here the fact that Lemma 5 is variance dependent, and that the variance depends on the radius r_0, is important. It is this property which enables a sharpening of the rate with step-by-step reductions in the variance bound, as in Section A.2.4.

For the empirical Rademacher complexity term, the first term of (A.7), Lemma 2, **Step I**, and

Lemma 3, yield

$$\begin{aligned}
\mathbb{E}_\eta R_n \mathcal{G} = \mathbb{E}_\eta R_n \{g : g = \ell(f, \boldsymbol{z}) - \ell(f_*, \boldsymbol{z}), f \in \mathcal{F}_{\mathrm{DNN}}, \|f - f_*\| \le r_0\} \\
\le 2C_\ell \mathbb{E}_\eta R_n \{f - f_* : f \in \mathcal{F}_{\mathrm{DNN}}, \|f - f_*\| \le r_0\} \\
\le 2C_\ell \mathbb{E}_\eta R_n \{f - f_* : f \in \mathcal{F}_{\mathrm{DNN}}, \|f - f_*\|_n \le 2r_0\} \\
\le 2C_\ell \inf_{0 < \alpha < 2r_0} \left\{ 4\alpha + \frac{12}{\sqrt{n}} \int_\alpha^{2r_0} \sqrt{\log \mathcal{N}(\delta, \mathcal{F}_{\mathrm{DNN}}, \|\cdot\|_n)} d\delta \right\} \\
\le 2C_\ell \inf_{0 < \alpha < 2r_0} \left\{ 4\alpha + \frac{12}{\sqrt{n}} \int_\alpha^{2r_0} \sqrt{\log \mathcal{N}(\delta, \mathcal{F}_{\mathrm{DNN}}|_{x_1,\ldots,x_n}, \infty)} d\delta \right\} ,
\end{aligned}$$

with probability $1 - \exp(-\tilde{\gamma})$ (when applying **Step I**). Recall Lemma 4, one can further upper bound the entropy integral when $n > \mathrm{Pdim}(\mathcal{F}_{\mathrm{DNN}})$,

$$\begin{aligned}
\inf_{0 < \alpha < 2r_0} \left\{ 4\alpha + \frac{12}{\sqrt{n}} \int_\alpha^{2r_0} \sqrt{\log \mathcal{N}(\delta, \mathcal{F}_{\mathrm{DNN}}|_{x_1,\ldots,x_n}, \infty)} d\delta \right\} \\
\le \inf_{0 < \alpha < 2r_0} \left\{ 4\alpha + \frac{12}{\sqrt{n}} \int_\alpha^{2r_0} \sqrt{\mathrm{Pdim}(\mathcal{F}_{\mathrm{DNN}}) \log \frac{2eMn}{\delta \cdot \mathrm{Pdim}(\mathcal{F}_{\mathrm{DNN}})}} d\delta \right\} \\
\le 32r_0 \sqrt{\frac{\mathrm{Pdim}(\mathcal{F}_{\mathrm{DNN}})}{n} \left(\log \frac{2eM}{r_0} + \frac{3}{2} \log n \right)}
\end{aligned}$$

with a particular choice of $\alpha = 2r_0 \sqrt{\mathrm{Pdim}(\mathcal{F}_{\mathrm{DNN}})/n} < 2r_0$. Therefore, whenever $r_0 \ge 1/n$ and $n \ge (2eM)^2$,

$$\mathbb{E}_\eta R_n \mathcal{G} \le 128 C_\ell r_0 \sqrt{\frac{\mathrm{Pdim}(\mathcal{F}_{\mathrm{DNN}})}{n} \log n}.$$

Applying this bound to (A.7), we have

$$(\mathbb{E} - \mathbb{E}_n) \left[\ell(\hat{f}, \boldsymbol{z}) - \ell(f_*, \boldsymbol{z}) \right] \le K r_0 \sqrt{\frac{\mathrm{Pdim}(\mathcal{F}_{\mathrm{DNN}})}{n} \log n} + r_0 \sqrt{\frac{2C_\ell^2 \tilde{\gamma}}{n}} + \frac{23 M C_\ell \tilde{\gamma}}{n} \qquad \text{(A.8)}$$

where $K = 6 \times 128 C_\ell$.

Going back now to the main decomposition, plug (A.8) and (A.2) into (A.1), and we overall have found that, with probability at least $1 - 4\exp(-\tilde{\gamma})$, the following holds:

$$\begin{aligned}
c_1 \|\hat{f} - f_*\|_{L_2(\boldsymbol{x})}^2 \\
\le K r_0 \sqrt{\frac{\mathrm{Pdim}(\mathcal{F}_{\mathrm{DNN}})}{n} \log n} + r_0 \sqrt{\frac{2C_\ell^2 \tilde{\gamma}}{n}} + \frac{23 M C_\ell \tilde{\gamma}}{n} + \left(c_2 \epsilon_n^2 + \epsilon_n \sqrt{\frac{2C_\ell^2 \tilde{\gamma}}{n}} + \frac{7 C_\ell M \tilde{\gamma}}{n} \right) \\
\le r_0 \cdot \left(K \sqrt{\frac{\mathrm{Pdim}(\mathcal{F}_{\mathrm{DNN}})}{n} \log n} + \sqrt{\frac{2C_\ell^2 \tilde{\gamma}}{n}} \right) + c_2 \epsilon_n^2 + \epsilon_n \sqrt{\frac{2C_\ell^2 \tilde{\gamma}}{n}} + 30 M C_\ell \frac{\tilde{\gamma}}{n}
\end{aligned}$$

$$\leq r_0 \cdot \left(K\sqrt{C}\sqrt{\frac{WL\log W}{n}}\log n + \sqrt{\frac{2C_\ell^2 \tilde{\gamma}}{n}} \right) + c_2 \epsilon_n^2 + \epsilon_n \sqrt{\frac{2C_\ell^2 \tilde{\gamma}}{n}} + 30MC_\ell \frac{\tilde{\gamma}}{n}. \tag{A.9}$$

The last line applies Lemma 6. Therefore, whenever $\epsilon_n \ll r_0$ and $\sqrt{\frac{WL\log W}{n}}\log n \ll r_0$, the knowledge that $\|\hat{f} - f_*\|_{L_2(\boldsymbol{x})} \leq r_0$ implies that (with high probability) $\|\hat{f} - f_*\|_{L_2(\boldsymbol{x})} \leq r_1$, for $r_1 \ll r_0$. One can recursively improve the bound r to a fixed point/radius r_*, which describes the fundamental difficulty of the problem. This is done in the course of the next two steps.

A.2.3 Step III: Critical Radius

We now use the tightening of **Step II** to obtain the critical radius for this problem that is then used as an input in the final rate derivation of **Step IV**. Formally, define the critical radius r_* to be the largest fixed point

$$r_* = \inf\left\{ r > 0 : 18M\mathbb{E}R_n\{f - f_* : f \in \mathcal{F}, \|f - f_*\|_{L_2(\boldsymbol{x})} \leq s\} < s^2, \forall s \geq r \right\}.$$

By construction this obeys (A.4), and thus so does $2r_*$. Denote the event E (depending on the data) to be

$$E = \left\{ \|f - f_*\|_n \leq 4r_*, \quad \text{for all } f \in \mathcal{F} \text{ and } \|f - f_*\|_{L_2(\boldsymbol{x})} \leq 2r_* \right\}$$

and $\mathbb{1}_E$ to be the indicator that event E holds. We know from (A.6) that $\mathbb{P}(\mathbb{1}_E = 1) \geq 1 - n^{-1}$, provided $r_* \geq \sqrt{18}M\sqrt{\log n/n}$ to satisfy (A.5).

We can now give an upper bound for the the critical radius r_*. Using the logic of **Step II** to bound the empirical Rademacher complexity, and then applying Lemma 6, we find that

$$\begin{aligned}
r_*^2 &\leq 18M\mathbb{E}R_n\{f - f_* : f \in \mathcal{F}, \|f - f_*\|_{L_2(\boldsymbol{x})} \leq r_*\} \\
&\leq 18M\mathbb{E}R_n\{f - f_* : f \in \mathcal{F}, \|f - f_*\|_{L_2(\boldsymbol{x})} \leq 2r_*\} \\
&\leq 18M\mathbb{E}\{\mathbb{E}_\eta R_n\{f - f_* : f \in \mathcal{F}, \|f - f_*\|_n \leq 4r_*\}\mathbb{1}_E + 3M(1 - \mathbb{1}_E)\} \\
&\leq 36MK\sqrt{C} \cdot r_*\sqrt{\frac{WL\log W}{n}}\log n + 36M^2\frac{1}{n} \\
&\leq 72MK\sqrt{C} \cdot r_*\sqrt{\frac{WL\log W}{n}}\log n,
\end{aligned}$$

with the last line relying on the above restriction that $r_* \geq \sqrt{18}M\sqrt{\log n/n}$. Dividing through by r_* yields the final bound:

$$r_* \leq 72MK\sqrt{C}\sqrt{\frac{WL\log W}{n}}\log n. \tag{A.10}$$

A.2.4 Step IV: Localization

We are now able to derive the final rate using a localization argument. This applies the results of **Step I** and **Step II** repeatedly. Divide the space $\mathcal{F}_{\mathrm{DNN}}$ into shells of increasing radius by intersecting it with the L_2 balls

$$B(f_*, \bar{r}), B(f_*, 2\bar{r}) \backslash B(f_*, \bar{r}), \dots B(f_*, 2^l \bar{r}) \backslash B(f_*, 2^{l-1}\bar{r}) \tag{A.11}$$

where $l \geq 1$ is chosen to be the largest integer no greater than $\log_2 \frac{2M}{\sqrt{(\log n)/n}}$. We will proceed to find a bound on $\bar{r}$ which determines the final rate results.

Suppose $\bar{r} > r_*$. Then for each shell, **Step I** and the union bound imply that with probability at least $1 - 2l\exp(-\tilde{\gamma})$,

$$\|f - f_*\|_{L_2(\boldsymbol{x})} \leq 2^j \bar{r} \implies \|f - f_*\|_n \leq 2^{j+1}\bar{r}. \tag{A.12}$$

Further, suppose that for some $j \leq l$

$$\hat{f} \in B(f_*, 2^j \bar{r}) \backslash B(f_*, 2^{j-1}\bar{r}). \tag{A.13}$$

Then applying the one step improvement argument in **Step II** (again the variance dependence captured in Lemma 5 is crucial, here reflected in the variance within each shell), Equation (A.9) yields that with probability at least $1 - 4\exp(-\tilde{\gamma})$,

$$\|\hat{f} - f_*\|_{L_2(\boldsymbol{x})}^2 \leq \frac{1}{c_1} \left\{ 2^j \bar{r} \cdot \left(K\sqrt{C}\sqrt{\frac{WL \log W}{n}} \log n + \sqrt{\frac{2C_\ell^2 t}{n}} \right) + c_2 \epsilon_n^2 + \epsilon_n \sqrt{\frac{2C_\ell^2 \tilde{\gamma}}{n}} + 30 MC_\ell \frac{\tilde{\gamma}}{n} \right\}$$

$$\leq 2^{2j-2}\bar{r}^2,$$

if the following two conditions hold:

$$\frac{1}{c_1} \left(K\sqrt{C}\sqrt{\frac{WL \log W}{n}} \log n + \sqrt{\frac{2C_\ell^2 \tilde{\gamma}}{n}} \right) \leq \frac{1}{2} 2^j \bar{r}$$

$$\frac{1}{c_1} \left(c_2 \epsilon_n^2 + \epsilon_n \sqrt{\frac{2C_\ell^2 \tilde{\gamma}}{n}} + 26 MC_\ell \frac{\tilde{\gamma}}{n} \right) \leq \frac{1}{4} 2^{2j}\bar{r}^2.$$

It is easy to see that these two hold for all j if we choose

$$\bar{r} = \frac{8}{c_1} \left(K\sqrt{C}\sqrt{\frac{WL \log W}{n}} \log n + \sqrt{\frac{2C_\ell^2 \tilde{\gamma}}{n}} \right) + \left(\sqrt{\frac{2(c_2 \vee 1)}{c_1}} \epsilon_n + \sqrt{\frac{120 MC_\ell}{c_1} \frac{\tilde{\gamma}}{n}} \right) + r_*. \tag{A.14}$$

Therefore with probability at least $1 - 6l\exp(-\tilde{\gamma})$, we can perform shell-by-shell argument

combining the results in **Step I** and **Step II**:

$$\|\hat{f} - f_*\|_{L_2(\boldsymbol{x})} \leq 2^l \bar{r} \quad \text{and} \quad \|\hat{f} - f_*\|_n \leq 2^{l+1} \bar{r}$$
$$\text{implies} \quad \|\hat{f} - f_*\|_{L_2(\boldsymbol{x})} \leq 2^{l-1} \bar{r} \quad \text{and} \quad \|\hat{f} - f_*\|_n \leq 2^l \bar{r}$$
$$\ldots\ldots$$
$$\text{implies} \quad \|\hat{f} - f_*\|_{L_2(\boldsymbol{x})} \leq 2^0 \bar{r} \quad \text{and} \quad \|\hat{f} - f_*\|_n \leq 2^1 \bar{r}.$$

The "and" part of each line follows from **Step I** and the implication uses the above argument following **Step II**. Therefore in the end, we conclude with probability at least $1 - 6l \exp(-\tilde{\gamma})$,

$$\|\hat{f} - f_*\|_{L_2(\boldsymbol{x})} \leq \bar{r} \ , \tag{A.15}$$

$$\|\hat{f} - f_*\|_n \leq 2\bar{r} \ . \tag{A.16}$$

Therefore choose $\gamma = -\log(6l) + \tilde{\gamma}$, we know from (A.14), and the upper bound on r_* in (A.10)

$$
\bar{r} \leq \frac{8}{c_1} \left(K\sqrt{C} \sqrt{\frac{WL \log W}{n}} \log n + \sqrt{\frac{2C_\ell^2 (\log\log n + \gamma)}{n}} \right)
$$
$$
+ \left(\sqrt{\frac{2(c_2 \vee 1)}{c_2}} \epsilon_n + \sqrt{\frac{120 M C_\ell}{c_1} \frac{\log\log n + \gamma}{n}} \right) + r_*
$$
$$
\leq C' \left(\sqrt{\frac{WL \log W}{n}} \log n + \sqrt{\frac{\log\log n + \gamma}{n}} + \epsilon_n \right), \tag{A.17}
$$

with some constant $C' > 0$ that does not depend on n. This completes the proof of Theorem 2.

A.3 Final Steps for the MLP case

For the multi-layer perceptron, $W \leq C \cdot H^2 L$, and plugging this into the bound (A.17), we obtain

$$
C' \left(\sqrt{\frac{H^2 L^2 \log(H^2 L)}{n}} \log n + \sqrt{\frac{\log\log n + \gamma}{n}} + \epsilon_n \right)
$$

To optimize this upper bound on $\bar{r}$, we need to specify the trade-offs in ϵ_n and H and L. To do so, we utilize the MLP-specific approximation rate of Lemma 7 and the embedding of Lemma 1. Lemma 1 implies that, for any ϵ_n, one can embed the approximation class $\mathcal{F}_{\text{DNN}}$ given by Lemma 7 into a standard MLP architecture $\mathcal{F}_{\text{MLP}}$, where specifically

$$
H = H(\epsilon_n) \leq W(\epsilon_n) L(\epsilon_n) \leq C^2 \epsilon_n^{-\frac{d}{\beta}} (\log(1/\epsilon_n) + 1)^2,
$$
$$
L = L(\epsilon_n) \leq C \cdot (\log(1/\epsilon_n) + 1).
$$

For standard MLP architecture $\mathcal{F}_{\mathrm{MLP}}$,

$$H^2 L^2 \log(H^2 L) \leq \tilde{C} \cdot \epsilon_n^{-\frac{2d}{\beta}} (\log(1/\epsilon_n) + 1)^7.$$

Thus we can optimize the upper bound

$$\bar{r} \leq C' \left(\sqrt{\frac{\epsilon_n^{-\frac{2d}{\beta}} (\log(1/\epsilon_n) + 1)^7}{n}} \log n + \sqrt{\frac{\log \log n + \gamma}{n}} + \epsilon_n \right)$$

by choosing $\epsilon_n = n^{-\frac{\beta}{2(\beta+d)}}$, $H \asymp \cdot n^{\frac{d}{2(\beta+d)}} \log^2 n$, $L \asymp \cdot \log n$. This gives

$$\bar{r} \leq C \left(n^{-\frac{\beta}{2(\beta+d)}} \log^4 n + \sqrt{\frac{\log \log n + t'}{n}} \right).$$

Hence putting everything together, with probability at least $1 - \exp(-\gamma)$,

$$\mathbb{E}(\hat{f} - f_*)^2 \leq \bar{r}^2 \leq C \left(n^{-\frac{\beta}{\beta+d}} \log^8 n + \frac{\log \log n + \gamma}{n} \right) \quad,$$

$$\mathbb{E}_n(\hat{f} - f_*)^2 \leq (2\bar{r})^2 \leq 4C \left(n^{-\frac{\beta}{\beta+d}} \log^8 n + \frac{\log \log n + \gamma}{n} \right) \quad.$$

This completes the proof of Theorem 1.

A.4 Proof of Corollaries 1 and 2

For Corollary 1, we want to optimize

$$\frac{WL \log U}{n} \log n + \frac{\log \log n + \gamma}{n} + \epsilon_{\mathrm{DNN}}^2.$$

Yarotsky (2017, Theorem 1) shows that for the approximation error ϵ_{DNN} to obey $\epsilon_{\mathrm{DNN}} \leq \epsilon$, it suffices to choose $W, U \propto \epsilon^{-\frac{d}{\beta}} (\log(1/\epsilon) + 1)$ and $L \propto (\log(1/\epsilon) + 1)$, given the specific architecture described therein. Therefore, we attain $\epsilon \asymp n^{-\beta/(2\beta+d)}$ by setting $W, U \asymp n^{d/(2\beta+d)}$ and $L \asymp \log n$, yielding the desired result.

For Corollary 2, we need to optimize

$$\frac{H^2 L_2 \log(HL)}{n} \log n + \frac{\log \log n + \gamma}{n} + \epsilon_{\mathrm{MLP}}^2.$$

Yarotsky (2018, Theorem 1) shows that for the approximation error ϵ_{MLP} to obey $\epsilon_{\mathrm{MLP}} \leq \epsilon$, it suffices to choose $H \propto 2d + 10$ and $L \propto \epsilon^{-\frac{d}{2}}$, given the specific architecture described therein. Thus, for $\epsilon \asymp n^{-1/(2+d)}$ we take $L \asymp n^{-d/(4+2d)}$, and the result follows.

B Supporting Lemmas

First, we show that one can embed a feedforward network into the multi-layer perceptron architecture by adding auxiliary hidden nodes. This idea is due to Yarotsky (2018).

Lemma 1 (Embedding). *For any function $f \in \mathcal{F}_{\mathrm{DNN}}$, there is a $g \in \mathcal{F}_{\mathrm{MLP}}$, with $H \leq WL + U$, such that $g = f$.*
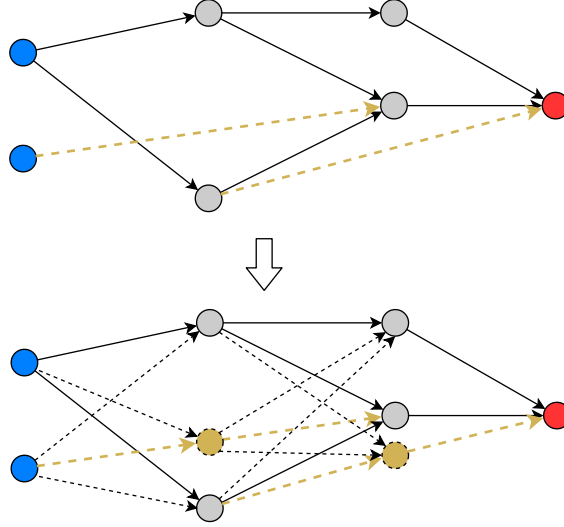

Figure 7: Illustration of how to embed a feedforward network into a multi-layer perceptron, with auxiliary hidden nodes (shown in yellow).

Proof. The idea is illustrated in Figure 7. For the edges in the directed graph of $f \in \mathcal{F}_{\mathrm{DNN}}$ that connect nodes not in adjacent layers (shown in yellow in Figure 7), one can insert auxiliary hidden units in order to simply "pass forward" the information. The number of such auxiliary "passforward units" is at most the number of offending edges times the depth L (i.e. for each edge, at most L auxiliary nodes are required), and this is bounded by WL. Therefore the width of the MLP network that subsumes the original is upper bounded by $WL + U$ while still maintaining the required embedding that for any $f_\theta \in \mathcal{F}_{\mathrm{DNN}}$, there is a $g_{\theta'} \in \mathcal{F}_{\mathrm{MLP}}$ such that $g_{\theta'} = f_\theta$. In order to match modern practice we only need to show that auxiliary units can be implemented with ReLU activation. This can be done by setting the constant ("bias") term b of each auxiliary unit large enough to ensure $\sigma(\tilde{\boldsymbol{x}}'\boldsymbol{w} + b) = \tilde{\boldsymbol{x}}'\boldsymbol{w} + b$, and then subtracting the same b in the last receiving unit along the path. $\qquad\square$

Next, we give two properties of the Rademacher complexity that we require (see Mendelson, 2003).

Lemma 2 (Contraction). *Let $\phi : \mathbb{R} \to \mathbb{R}$ be a Lipschitz function $|\phi(f_1) - \phi(f_2)| \leq L|f_1 - f_2|$, then*

$$\mathbb{E}_\eta R_n\{\phi \circ f : f \in \mathcal{F}\} \leq 2L\mathbb{E}_\eta R_n \mathcal{F}.$$

Lemma 3 (Dudley's Chaining). *Let $\mathcal{N}(\delta, \mathcal{F}, \|\cdot\|_n)$ denote the metric entropy for class $\mathcal{F}$ (with covering radius δ and metric $\|\cdot\|_n$), then*

$$\mathbb{E}_\eta R_n\{f : f \in \mathcal{F}, \|f\|_n \le r\} \le \inf_{0<\alpha<r} \left\{ 4\alpha + \frac{12}{\sqrt{n}} \int_\alpha^r \sqrt{\log \mathcal{N}(\delta, \mathcal{F}, \|\cdot\|_n)} d\delta \right\} \ .$$

Furthermore, because $\|f\|_n \le \max_i |f(\boldsymbol{x}_i)|$, and therefore $\mathcal{N}(\delta, \mathcal{F}, \|\cdot\|_n) \le \mathcal{N}(\delta, \mathcal{F}|_{\boldsymbol{x}_1,...,\boldsymbol{x}_n}, \infty)$ and so the upper bound in the conclusions also holds with $\mathcal{N}(\delta, \mathcal{F}|_{\boldsymbol{x}_1,...,\boldsymbol{x}_n}, \infty)$.

The next two results, Theorems 12.2 and 14.1 in Anthony and Bartlett (1999), show that the metric entropy may be bounded in terms of the pseudo-dimension and that the latter is bounded by the Vapnik-Chervonenkis (VC) dimension.

Lemma 4. *Assume for all $f \in \mathcal{F}$, $\|f\|_\infty \le M$. Denote the pseudo-dimension of $\mathcal{F}$ as $\mathrm{Pdim}(\mathcal{F})$, then for $n \ge \mathrm{Pdim}(\mathcal{F})$, we have for any δ,*

$$\mathcal{N}(\delta, \mathcal{F}|_{x_1,...,x_n}, \infty) \le \left(\frac{2eM \cdot n}{\delta \cdot \mathrm{Pdim}(\mathcal{F})} \right)^{\mathrm{Pdim}(\mathcal{F})} \ .$$

The following symmetrization lemma bounds the empirical processes term using Rademacher complexity, and is thus a crucial piece of our localization. This is a standard result based on Talagrand's concentration, but here special care is taken with the dependence on the variance.

Lemma 5 (Symmetrization, Theorem 2.1 in Bartlett et al. (2005)). *For any $g \in \mathcal{G}$, assume that $|g| \le G$ and $\mathbb{V}[g] \le V$. Then for every $\gamma > 0$, with probability at least $1 - e^{-\gamma}$*

$$\sup_{g \in \mathcal{G}} \{\mathbb{E}g - \mathbb{E}_n g\} \le 3\mathbb{E}R_n\mathcal{G} + \sqrt{\frac{2V\gamma}{n}} + \frac{4G}{3}\frac{\gamma}{n} \ ,$$

and with probability at least $1 - 2e^{-t}$

$$\sup_{g \in \mathcal{G}} \{\mathbb{E}g - \mathbb{E}_n g\} \le 6\mathbb{E}_\eta R_n\mathcal{G} + \sqrt{\frac{2V\gamma}{n}} + \frac{23G}{3}\frac{\gamma}{n} \ .$$

The same result holds for $\sup_{g \in \mathcal{G}} \{\mathbb{E}_n g - \mathbb{E}g\}$.

When bounding the complexity of $\mathcal{F}_{\mathrm{DNN}}$, we use the following result. Bartlett et al. (2017) also verify these bounds for the VC-dimension.

Lemma 6 (Theorem 6 in Bartlett et al. (2017), ReLU case). *Consider a ReLU network architecture $\mathcal{F} = \mathcal{F}_{\mathrm{DNN}}(W, L, U)$, then the pseudo-dimension is sandwiched as*

$$c \cdot WL \log(W/L) \le \mathrm{Pdim}(\mathcal{F}) \le C \cdot WL \log W,$$

with some universal constants $c, C > 0$.

For multi-layer perceptrons we use the following approximation result, Theorem 1 of Yarotsky (2017).

Lemma 7. *There exists a network class $\mathcal{F}_{\text{DNN}}$, with ReLU activation, such that for any $\epsilon > 0$:*

(a) $\mathcal{F}_{\text{DNN}}$ *approximates the $W^{\beta,\infty}([-1,1]^d)$ in the sense for any $f_* \in W^{\beta,\infty}([-1,1]^d)$, there exists a $f_n(\epsilon) := f_n \in \mathcal{F}_{\text{DNN}}$ such that*

$$\|f_n - f_*\|_\infty \le \epsilon,$$

(b) *and $\mathcal{F}_{\text{DNN}}$ has $L(\epsilon) \le C \cdot (\log(1/\epsilon) + 1)$ and $W(\epsilon), U(\epsilon) \le C \cdot \epsilon^{-\frac{d}{\beta}}(\log(1/\epsilon) + 1)$.*

Here C only depends on d and β.

For completeness, we verify the requirements on the loss functions, Equation (2.1), for several examples. We first treat least squares and logistic losses, in slightly more detail, as these are used in our subsequent inference results and empirical application.

Lemma 8. *Both the least squares (2.2) and logistic (2.3) loss functions obey the requirements of Equation (2.1). For least squares, $c_1 = c_2 = 1/2$ and $C_\ell = M$. For logistic regression, $c_1 = (2(\exp(M) + \exp(-M) + 2))^{-1}$, $c_2 = 1/8$ and $C_\ell = 1$.*

Proof. The Lipschitz conditions are trivial. For least squares, using iterated expectations

$$
\begin{aligned}
2\mathbb{E}\ell(f, \boldsymbol{Z}) - 2\mathbb{E}\ell(f_*, \boldsymbol{Z}) &= \mathbb{E}\left[-2Yf + f^2 + 2Yf_* - f_*^2\right] \\
&= \mathbb{E}\left[-2f_*f(\boldsymbol{x}) + f^2 + 2(f_*)^2 - f_*^2\right] \\
&= \mathbb{E}\left[(f - f_*)^2\right].
\end{aligned}
$$

For logistic regression,

$$\mathbb{E}[\ell(f, \boldsymbol{Z})] - \mathbb{E}[\ell(f_*, \boldsymbol{Z})] = \mathbb{E}\left[-\frac{\exp(f_*)}{1 + \exp(f_*)}(f - f_*) + \log\left(\frac{1 + \exp(f)}{1 + \exp(f_*)}\right)\right].$$

Define $h_a(b) = -\frac{\exp(a)}{1+\exp(a)}(b - a) + \log\left(\frac{1+\exp(b)}{1+\exp(a)}\right)$, then

$$h_a(b) = h_a(a) + h_a'(a)(b - a) + \frac{1}{2}h_a''\left(\xi a + (1 - \xi)b\right)(b - a)^2$$

and $h_a''(b) = \frac{1}{\exp(b)+\exp(-b)+2} \le \frac{1}{4}$. The lower bound holds as $|\xi f_* + (1 - \xi)f| \le M$. $\square$

Beyond least squares and logistic regression, we give three further examples, discussed in the general language of generalized linear models. Note that in the final example we move beyond a simple scalar outcome.

Lemma 9. *For a convex function $g(\cdot) : \mathbb{R} \to \mathbb{R}$, consider the generalized linear loss function $\ell(f, \boldsymbol{z}) = -\langle y, f(\boldsymbol{x})\rangle + g(f(\boldsymbol{x}))$. The curvature and the Lipschitz conditions in (2.1) will hold given specific $g(\cdot)$. In each case, the loss function corresponds to the negative log likelihood function.*

(a) *Poisson:* $g(t) = \exp(t)$, *with* $f_*(\boldsymbol{x}) = \log \mathbb{E}[y|\boldsymbol{X} = \boldsymbol{x}]$.

(b) *Gamma:* $g(t) = -\log t$, *with* $f_*(\boldsymbol{x}) = -(\mathbb{E}[y|\boldsymbol{X} = \boldsymbol{x}])^{-1}$.

(c) *Multinomial Logistic,* $K+1$ *classes:* $g(t) = \log(1 + \sum_{k \in K} \exp(t^{[k]}))$, *with*

$$\exp(f_*^{[k]}(\boldsymbol{x}))/(1 + \sum_{k' \in K} \exp(f_*^{[k']}(\boldsymbol{x}))) = \mathbb{E}[y^{[k]}|\boldsymbol{X} = \boldsymbol{x}].$$

Here $v^{[k]}$ *denotes the k-th coordinate of a vector* $\boldsymbol{v}$.

Proof. Denote ∇g, $\text{Hessian}[g]$ to be the gradient and Hessian of the convex function g. By the convexity of g, the optimal f_* satisfies $\mathbb{E}[\partial \ell(f_*, \boldsymbol{Z})/\partial f|\boldsymbol{X} = \boldsymbol{x}] = 0$, which implies

$$\nabla g(f_*) = \mathbb{E}[Y|\boldsymbol{X} = \boldsymbol{x}].$$

If $2c_0 \preceq \text{Hessian}[g(f)] \preceq 2c_1$ for all f of interest, then the curvature condition in (2.1) holds, because

$$\begin{aligned}
\mathbb{E}[\ell(f, \boldsymbol{Z})] - \mathbb{E}[\ell(f_*, \boldsymbol{Z})] &= \mathbb{E}[-\langle \nabla g(f_*), f - f_* \rangle + g(f) - g(f_*)] \\
&= \frac{1}{2}\mathbb{E}\langle f - f_*, \text{Hessian}[g(\tilde{f})]f - f_* \rangle \\
&\geq c_0 \mathbb{E}\|f - f_*\|^2,
\end{aligned}$$

and the parallel argument for $\leq c_1\mathbb{E}\|f - f_*\|^2$. The Lipschitz condition is equivalent to $\|\nabla g(f)\| \leq C'_\ell$ for all f of interest, with bounded Y.

For our three examples in particular, we have the following.

(a) For Poisson regression:

$$\|\nabla c(f)\| = |\exp(f)| \leq \exp(M), \qquad \text{Hessian}[c(f)] = \exp(f) \in [\exp(-M), \exp(M)].$$

(b) For Gamma regression, bounding $-Y$ above and below is equivalent to $1/M \leq \|f\| \leq M$ and therefore:

$$\|\nabla c(f)\| = |1/f| \leq M, \quad \text{Hessian}[c(f)] = 1/f^2 \in [1/M^2, M^2].$$

(c) For multinomial logistic regression, with general K, we know

$$\|\nabla c(f)\| \leq 1$$

$$\text{Hessian}[c(f)] = \text{diag}\{z\} - zz^\top \quad \text{where } z^{[k]} := \frac{\exp(f^{[k]})}{1 + \sum_{k'} f^{[k']}}.$$

One can easily verify that the eigenvalues are bounded in the following sense, for bounded f,

$$\frac{1}{(1 + K \exp(M))^2} \leq \lambda(\text{Hessian}[c(f)]) \leq \frac{\exp(M)}{1 + (K-1)\exp(-M) + \exp(M)}.$$

This completes the proof. $\qquad\square$

Our last result is to verify condition (c) of Theorem 3. We do so using our localization, which may be of future interest in second-step inference with machine learning methods.

Lemma 10. *Let the conditions of Theorem 3 hold. Then*

$$\mathbb{E}_n\left[(\hat{\mu}_t(\boldsymbol{x}_i) - \mu_t(\boldsymbol{x}_i))\left(1 - \frac{\mathbb{1}\{t_i = t\}}{\mathbb{P}[T = t | \boldsymbol{X} = \boldsymbol{x}_i]}\right)\right] = o_P\left(n^{-\frac{\beta}{\beta+d}}\log^8 n + \frac{\log\log n}{n}\right) = o_P\left(n^{-1/2}\right).$$

Proof. Without loss of generality we can take $\bar{p} < 1/2$. The only estimated function here is $\mu_t(\boldsymbol{x})$, which plays the role of f_* here. For function(als) $L(\cdot)$ of the form

$$L(f) := (f(\boldsymbol{x}_i) - f_*(\boldsymbol{x}_i))\left(1 - \frac{\mathbb{1}\{t_i = t\}}{\mathbb{P}[T = t | \boldsymbol{X} = \boldsymbol{x}_i]}\right),$$

it is true that

$$\mathbb{E}[L(f)] = \mathbb{E}\left[(f(\boldsymbol{X}) - f_*(\boldsymbol{X}))\left(1 - \frac{\mathbb{E}[\mathbb{1}\{t_i = t\}|\boldsymbol{x}_i]}{\mathbb{P}[T = t | \boldsymbol{X} = \boldsymbol{x}_i]}\right)\right] = 0$$

and

$$\mathbb{V}[L(f)] \leq (1/\bar{p} - 1)^2 \mathbb{E}\left[(f(\boldsymbol{X}) - f_*(\boldsymbol{X}))^2\right] \leq (1/\bar{p} - 1)^2 \bar{r}^2$$

$$|L(f)| \leq (1/\bar{p} - 1) 2M.$$

For $\bar{r}$ defined in (A.14),

$$18M\mathbb{E}R_n\{f - f_* : f \in \mathcal{F}, \|f - f_*\|_{L_2(\boldsymbol{x})} \leq \bar{r}\} \leq \bar{r}^2$$

$$\mathbb{E}R_n\{L(f) : f \in \mathcal{F}, \|f - f_*\|_{L_2(\boldsymbol{x})} \leq \bar{r}\} \leq 2(1/\bar{p} - 1)\mathbb{E}R_n\{f - f_* : f \in \mathcal{F}, \|f - f_*\|_{L_2(\boldsymbol{x})} \leq \bar{r}\}$$

where the first line is due to $\bar{r} > r_*$, and second line uses Lemma 2.

Then by the localization analysis and Lemma 5, for all $f \in \mathcal{F}, \|f - f_*\|_{L_2(\boldsymbol{x})} \leq \bar{r}$, $L(f)$ obeys

$$\mathbb{E}_n[L(f)] = \mathbb{E}_n[L(f)] - \mathbb{E}[L(f)] \leq 6C\bar{r}^2 + \bar{r}\sqrt{\frac{4(1/\bar{p}-1)^2 t}{n}} + \frac{8(1/\bar{p}-1)3M}{3}\frac{t}{n} \leq 4C\bar{r}^2$$

$$\leq C \cdot \left\{n^{-\frac{\beta}{\beta+d}}\log^8 n + \frac{\log\log n}{n}\right\},$$

$$\sup_{f \in \mathcal{F}, \|f - f_*\|_{L_2(\boldsymbol{x})} \leq \bar{r}} \mathbb{E}_n[L(f)] \leq C \cdot \left\{n^{-\frac{\beta}{\beta+d}}\log^8 n + \frac{\log\log n}{n}\right\}.$$

With probability at least $1 - \exp(-n^{\frac{d}{\beta+d}} \log^8 n)$, $\hat{f}_{\mathrm{MLP}}$ lies in this set of functions, and therefore

$$\mathbb{E}_n[L(\hat{f}_{\mathrm{MLP}})] = \mathbb{E}_n\left[(\widehat{f}_{n,H,L}(x) - f_*(x))\left(1 - \frac{\mathbb{1}(T=t)}{P(T=t|\boldsymbol{x}=x)}\right)\right] \le C \cdot \left\{n^{-\frac{\beta}{\beta+d}} \log^8 n + \frac{\log\log n}{n}\right\},$$

as claimed. $\qquad\square$