Efficient Estimation in NPIV Models: A Comparison of Various Neural Networks-Based Estimators*

Jiafeng Chen† Xiaohong Chen‡ Elie Tamer§

First draft: September 2019 - Revised draft: September 23, 2021

Abstract

We investigate the computational performance of Artificial Neural Networks (ANNs) in semi-nonparametric instrumental variables (NPIV) models of high dimensional covariates that are relevant to empirical work in economics. We focus on efficient estimation of and inference on expectation functionals (such as weighted average derivatives) and use optimal criterion-based procedures (sieve minimum distance or SMD) and novel efficient score-based procedures (ES). Both these procedures use ANN to approximate the unknown function. Then, we provide a detailed practitioner's recipe for implementing these two classes of estimators. This involves the choice of tuning parameters both for the unknown functions (that include conditional expectations) but also for the choice of estimation of the optimal weights in SMD and the riesz representers used with the ES estimators. Finally, we conduct a large set of Monte Carlo experiments that compares the finite-sample performance in complicated designs that involve a large set of regressors (up to 13 continuous), and various underlying nonlinearities and covariate correlations. Some of the takeaways from our results include: 1) tuning and optimization are delicate especially as the problem is nonconvex; 2) various architectures of the ANNs do not seem to matter for the designs we consider and given proper tuning, ANN methods perform well; 3) stable inferences are more difficult to achieve with ANN estimators; 4) optimal SMD based estimators perform adequately; 5) there seems to be a gap between implementation and approximation theory. Finally, we apply ANN NPIV to estimate average price elasticity and average derivatives in two demand examples.

JEL Classification: C14; C22

Keywords: Artificial neural networks; Relu; Sigmoid; High dimensional regressors; Nonparametric instrumental variables; Expectation functionals; Semiparametric efficiency; Price elasticity.

*We thank Vanya Klenovskiy for excellent research assistance in producing the empirical results for the demand example. We thank Josh Purtell for help in checking our code. We also thank Andrii Babii, Denis Chetverikov, Oliver Linton, Markus Pelger, Pedro Sant'Anna, Allan Timmermann, Ying Zhu, and participants at various seminars and conferences for helpful comments. Any errors are the responsibility of the authors.

†Department of Economics, Harvard University. *Email*: jiafengchen@g.harvard.edu.

‡Cowles Foundation for Research in Economics, Yale University. *Email*: xiaohong.chen@yale.edu.

§Department of Economics, Harvard University. *Email*: elietamer@fas.harvard.edu.

1 Introduction

Deep layer Artificial Neural Networks (ANNs) are increasingly popular in machine learning (ML), statistics, business, finance, and other fields. The universal approximation property of multi-layer (including single-hidden layer) ANNs (with various nonlinear activation functions) has been established by Hornik, Stinchcombe and White (1989) and others. Early on, computational difficulties have hindered the wide applicability of ANNs. Recently, fast algorithms have led to successful applications of deep layer ANNs in image detection, speech recognition, natural language processing and other areas with complex nonlinearity and large data sets of high quality.[1] Many problems where neural networks are extremely effective involve prediction problems (i.e. estimating conditional means)—or problems in which nuisance parameters are themselves predictions—on large datasets. It remains to be seen whether deep ANNs are similarly effective for structural estimation problems with nonparametric endogeneity, where relations to prediction are more tenuous or more complex.

To that end, we consider semiparametric efficient estimation of the average partial derivative of a nonparametric instrumental variables regression (NPIV) via ANN sieves. Average partial derivatives of structural relationships are linked to (cross) elasticities of endogenous demand systems in economics, finance, and business. We make three contributions. First, we consider efficient estimation for average derivatives in NPIV models with ANN sieves and derive the theoretical properties of two classes of estimation procedures—optimal criterion-based procedures (sieve minimum distance) and efficient score-based procedures.[2] Second, we detail a practitioner's recipe for implementing these two classes of estimators. Third, and perhaps most importantly, we show a large amount of Monte Carlo evidence that compares the finite-sample performance of the estimation paradigms that we consider. These are implemented using large scale designs, some with up to 13 continuous regressors, various nonlinearities and correlations among the covariates.

We now briefly introduce the two classes of estimation procedures that we consider. The first set of estimators belong to the class of *sieve minimum distance* (SMD), both under identity and optimal weighting (which we refer to as P-ISMD , for plug-in SMD and OP-OSMD, for orthogonal plug-in optimal SMD, respectively). The criterion-based SMD paradigm is numerically equivalent to a semiparametric two-step procedure, where the unknown NPIV function $h(\cdot)$ (in the model $E[Y_1 - h(Y_2) \mid X] = 0$) is estimated via an ANN SMD in the first step, and the average partial derivative of $h_0(\cdot)$ is estimated using an unconditional moment in the second step. The results of

[1] By high quality we mean a data set with signal-to-noise ratio that is very high. Unfortunately, many economic and social science data sets have low signal-to-noise ratios.

[2] We also exposit the theoretical properties of the associated inefficient estimators.

Ai and Chen (2003, 2007, 2012) (henceforth, AC03, AC07, and AC12) show that the optimally-weighted SMD procedure (OP-OSMD) automatically yields a semiparametrically efficient estimator of average derivatives of $h(\cdot)$. The second type of estimators are based on influence functions (equivalently, on semiparametric scores).[3] By considering the influence functions of the efficient SMD estimator (OP-OSMD, derived in AC12), we obtain the efficient score estimation procedure. Similarly, we may derive the identity score procedure (IS) from (P-ISMD). In particular, we establish the asymptotic properties of the efficient score estimator when combined with a sieve first-step, which is novel to our knowledge.

These two classes of estimators–criterion-based (SMD) and score-based estimation procedures—represent two different perspectives for semiparametric estimation. In sieve minimum distance, semiparametric efficiency is achieved through clever choices of weighting of the minimum distance criterion, similar to optimally weighted GMM. Influence function estimators, on the other hand, treat the estimation as a two-step GMM problem. Compared to simpler settings, e.g. estimating average treatment effect under unconfoundedness, the influence functions here are not in closed form and involve a Riesz representer of a Hilbert space constructed by a norm connected to the SMD objective. Components of the influence functions may nonetheless be consistently estimated via sieve approximations.[4]

We compare the finite sample performance of these estimation procedures (OP-OSMD and ES) in three Monte Carlo designs with moderate sample sizes ($n = 1000$ or $n = 5000$).[5] In our first Monte Carlo design we estimate the average partial derivative of a NPIV function with respect to an exogenous variable using various ANN sieves. In the second and third Monte Carlo designs, we estimate the average partial derivative of a NPIV function with respect to an endogenous variables using various ANN sieves and spline sieves. Our Monte Carlo experiments allow for comparisons along several dimensions:

- Within a type of estimation procedure, do ANN estimators exhibit superior finite-sample performance compared to linear sieve estimators (e.g. splines), when dimension of exogenous variables is moderately high?[6]

- Across types of estimators, how do ANN SMD estimators compare to ANN score estima-

[3] These estimators have a long history in semiparametrics. See Bickel *et al.* (1993) and Section 25.8 of Van der Vaart (2000), as well as references therein, for an introduction.

[4] In the case of linear sieves, these components may be estimated in closed form without use of nonlinear optimization.

[5] For reference and comparison, we also compare with the inefficient counterparts, P-ISMD and IS; moreover, we also compare with versions of the score-based estimators that utilize cross-fitting, popular in the double machine learning literature (Chernozhukov, Chetverikov, Demirer, Duflo, Hansen, Newey and Robins, 2018; Chernozhukov, Escanciano, Ichimura, Newey and Robins, 2021).

[6] To be clear, we are not speaking of "high dimension" in the $d/n \not\to 0$ sense.

tors, along with alternative procedures like adversarial GMM (Dikkala, Lewis, Mackey and Syrgkanis, 2020)?

- For ANN estimators, how much does ANN architecture matter? How much do other tuning parameters matter?

The results from the sets of numerical experiments we conduct provide us with some takeaways.

- Hyperparameter tuning—choice of instrument basis, learning rate, stopping criterion—is delicate and can affect performance of ANN-based estimators. In particular, generally nonconvex optimization could lead to unstable performances. However, certain values of the hyperparameters do result in good performance.
- We do not empirically observe systematic differences in performance as a function of neural architecture, within the feedforward neural network family. In our experience, the importance of neural architecture in our setting is not as high as tuning the optimization procedure.
- Stable inferences are currently more difficult to achieve for ANN based estimators for models with nonparametric endogeneity.
- OP-OSMD and IS have smaller bias than P-ISMD estimators, when combined with ANN for the average derivative parameter.
- OP-OSMD and P-ISMD with splines for the average derivative parameter are less biased, stable and accurate, and can outperform their ANN counterparts, even when the dimension is high (as high as thirteen, which exceeds theory predictions).
- Generally, there seems to be gaps between intuitions suggested by approximation theory and current implementation.

Lastly, as an application to real data, we apply the single hidden layer ANN sieve NPIV to estimate average price elasticity of gasoline demand using the data set of Blundell, Horowitz and Parey (2012) and to estimate average derivatives of the price-quantity relationship in strawberry, both applications containing multi-dimensional covariates.

Literature on ANNs in econometrics. ANNs can be viewed as an example of *nonlinear* sieves, which, compared to linear sieves (or series), can have faster approximation error rates for large classes of nonlinear functions of high dimensional regressors. Once after the approximation error rate of a specific ANN sieve is established for a class of unknown functions, the asymptotic

properties of estimation and inference based on the ANN sieve could be established by applying the general theory of sieve-based methods. Prior to the current wave of theoretical papers on multi-layer ANN in econometrics and statistics, including Yarotsky (2017),Farrell *et al.* (2018), Schmidt-Hieber (2019), Athey *et al.* (2019) and the references there in, there are already many theoretical results on nonparametric M-estimation and inference based on single hidden layer (now called shallow) ANNs. For example, Wooldridge and White (1988) obtained consistency of ANN least squares estimation of conditional mean function for time series heterogenous near epoch dependent data. Chen and White (1999) established faster than $n^{-1/4}$ (in root mean-squared error metric) ANN sieve M-estimation of various nonparametric functions, such as conditional means, conditional quantiles, conditional densities, of time series models, and also obtained the root-n consistent, asymptotic normality of plug-in ANN estimator of regular functionals. Stinchcombe and White (1998) provided consistent specification test via ANN sieves. Chen, Hong and Shum (2007) studied nonparametric likelihood ratio Vuong-style model selection test via ANN sieves, which relies on the root-n consistent asymptotic normality of plug-in ANN estimation of nonparametric entropy. Perhaps due to computational difficulties, ANNs nonlinear sieves have not been popular in economics until the recent computational advances from the machine learning community.

To the best of our knowleadge, Hartford, Lewis, Leyton-Brown and Taddy (2017) is the first paper to apply multi-layer ANNs to estimate NPIV function. The nonparametric convergence rates in AC03 and AC07 explicitly allow for nonlinear sieves such as ANNs to approximate and estimate the unknown structure functions of endogenous variables. They establish the root-n asymptotic normality of regular functionals of nonparametric conditional moment restrictions with smooth residual functions. However, there is no published work on efficient estimation of expectation functionals of unknown functions of endogenous variables via ANNs. We provide these results in this paper.

Literature on estimation of linear functionals of NPIVs. The identification and nonparametric consistent estimation of a pure NPIV function $h_0(Y_2)$ in a model $E[Y_1 - h_0(Y_2)|X] = 0$ have been first considered in Newey and Powell (2003). The semiparametric efficiency bound for EFs, including WADs of NPIV or nonparametric quantile instrumental variables (NPQIV) functions as examples, has been previously characterized by AC12.[7] Although AC12 suggested an estimator that could achieve the semiparametric efficiency bound for regular EFs of nonparametric conditional moment restrictions, they did not provide sufficient conditions to formally establish that their estimator is indeed semiparametric efficient. AC07 established the $\sqrt{n}$ asymptotic normality

[7] Ai and Chen (2012) derived the efficiency bound via the "orthogonalized residual" approach, which extends the earlier work of Chamberlain (1992a) and Brown and Newey (2002) to allow for unknown functions entering a system of sequential moment restrictions.

of plug-in estimator of WAD of a possibly misspecified NPIV model. Severini and Tripathi (2013) presented efficiency bound calculation for average weighted derivatives of a NPIV model without assuming point identification of the NPIV function, but pointed out that the $\sqrt{n}$ asymptotic normality estimator of linear functionals of NPIV in Santos (2012) fails to achieve the efficiency bound. Chen *et al.* (2019) proposed efficient estimation of weighted average derivatives of nonparametric quantile IV regression via penalized sieve GEL procedure, extending that of Donald *et al.* (2003) to allow for unknown quantile function of endogenous variables.

The rest of the paper is organized as follows. Section 2 introduces the model as a special case of the sequential moment restrictions containing unknown functions. It also presents the semiparametric efficiency score (or efficient influence function). Section 3 provides implementation details for all the estimators considered in the Monte Carlo studies. Section 4 contains three simulation studies and detailed Monte Carlo comparisons of various ANN and spline based estimators. Section 5 presents two empirical illustrations and Section 6 concludes the paper.

2 Two Efficient Estimation Procedures for Average Derivatives in NPIV Models

Using the framework in AC12, we first present the NPIV model and then present two procedures that will lead to semiparametric efficient estimation for average derivatives in NPIV models. The first one is based on efficient score (or efficient influence) equation, and the second one is based on an optimally weighted criterion.

We are interested in semiparametric efficient estimation of the average partial derivative:

$$\theta_0 \equiv \mathbb{E}[a(Y_2)\nabla_1 h_0(Y_2))],$$

where $a(\cdot)$ is a known positive weight function, ∇_1 takes the partial derivative w.r.t. the first argument and the unknown function $h_0 \in \mathcal{H}$ is identified via a real-valued conditional moment restriction

$$\mathbb{E}[Y_1 - h_0(Y_2)|X] = 0, \quad X \text{ almost sure} \tag{1}$$

Previously, while allowing for $\inf_h \mathbb{E}[(\mathbb{E}[Y_1 - h(Y_2)|X])^2] > 0$ (global misspecification), Ai and Chen (2007) (AC07) presented a root-n consistent asymptotically normally distributed identity-weighted SMD estimator of θ, allowing for nonlinear sieves such as single hidden layer ANN sieve is allowed for in their sufficient conditions. Ai and Chen (2012) (AC12) presented the semiparametric efficiency bound of θ (see their example 3.3) and an efficient estimator based on orthogonalized optimally weighted SMD (see their section 4.2).

In this paper we present several efficient estimators of θ via more general ANN nonlinear sieve approximation to $h_0(Y_2)$ when Y_2 is a vector of continuous endogenous and exogenous covariates of moderate high dimension (say up to 13 dimension). For example:

- In Monte Carlo 1, the DGP corresponds to

  $$h_0(Y_2) = X_1\theta_0 + h_{01}(R) + h_{02}(X_2) + h_{03}(\tilde{X}), \quad Y_2 = (X_1, R, X_2, \tilde{X}), \quad X = (X_1, X_3, X_2, \tilde{X})$$

  where R is endogenous and $\tilde{X}$ can be of high dimension. Note here that X_1 is exogenous. The parameter of interest is $\theta_0 = \mathbb{E}\nabla_1 h_0(.) = \mathbb{E}[X_1]$.

- In Monte Carlo 2, the DGP corresponds to

  $$h_0(Y_2) = R_1\theta_0 + h_{01}(R_2) + h_{02}(X_2) + h_{03}(\tilde{X}), \quad Y_2 = (R_1, R_2, X_2, \tilde{X}), \quad X = (X_1, X_3, X_2, \tilde{X})$$

  where (R_1, R_2) is endogenous but R_1 enters linearly. The parameter of interest is $\theta_0 = \mathbb{E}\nabla_1 h_0(.) = \mathbb{E}[R_1]$.

- In Monte Carlo 3, the DGP corresponds to

  $$h_0(Y_2) = f_0(R_1, X_2) + h_{01}(R_2) + h_{02}(X_2) + h_{03}(\tilde{X}), \quad Y_2 = (R_1, R_2, X_2, \tilde{X}), \quad X = (X_1, X_3, X_2, \tilde{X})$$

  where (R_1, R_2) is endogenous but R_1 now enters nonlinearly. Monte Carlo 3(a) specifies $f_0(R_1, X_2) = R_1^2$ and hence the parameter of interest $\theta_0 = \mathbb{E}\nabla_1 h_0(.) = 2\mathbb{E}[R_1]$. Monte Carlo 3(b) lets $f_0(R_1, X_2) = R_1^2/2 + R_1 f(X_2)$ where $f(.)$ is a nonlinear function, and then the parameter of interest $\theta_0 = \mathbb{E}\nabla_1 h_0(.) = \mathbb{E}[R_1 + f(X_2)]$.

2.1 Efficient score and efficient variance for θ

In this section, we specialize the general efficiency bound result of AC12 to our setting. We rewrite our model using their notation. Denote $\alpha_0 \equiv (\theta_0, h_0) \in \Theta \times \mathcal{H} \equiv \mathcal{A}$. The model can be written as

$$\begin{aligned} \mathbb{E}[\rho_2(Z, h_0(\cdot))|X] &= \mathbb{E}[Y_1 - h_0(Y_2) \mid X] = 0, \quad X \text{ almost sure} \\ \mathbb{E}[\rho_1(Z, \alpha_0)] &\equiv \mathbb{E}[a(Y_2)\nabla_1 h_0(Y_2) - \theta_0] = 0 \end{aligned} \tag{2}$$

We define the orthogonalized residual as

$$\varepsilon_1(Z, \alpha) \equiv \rho_1(Z, \alpha) - \Gamma(X)\rho_2(Z, h) = a(Y_2)\nabla_1 h(Y_2) - \theta - \Gamma(X) \cdot (Y_1 - h(Y_2)),$$

which is the residual from a projection of ρ_1 on ρ_2 conditional on X, where $\Gamma(X)$ is the orthogonal projection coefficient:

$$\Gamma(X) \equiv \frac{\text{Cov}(\rho_1(Z, \alpha_0)\rho_2(Z, h_0) \mid X)}{\text{Var}(\rho_2(Z, h_0) \mid X)}$$

Orthogonalizing the two moment conditions makes an efficiency analysis tractable—the same technique is used in, e.g., Chamberlain (1992b).

We know apply and specialize the results in AC12 to the plug-in model (3)

$$\mathbb{E}[\rho_2(Z, h_0(\cdot))|X] = 0 \text{ and } \mathbb{E}[\varepsilon_1(Z, \alpha_0)] = 0, \tag{3}$$

where θ is a scalar and h is a real-valued function of Y_2, and $\alpha = (\theta, h)$.

Let $m_1(\alpha) = \mathbb{E}[\varepsilon_1(Z, \alpha)]$, $m_2(X, h) = \mathbb{E}[\rho_2(Z, h)|X]$ and

$$\sigma_0^2 \equiv \mathbb{E}[\{\varepsilon_1(Z, \alpha_0)\}^2] = \text{Var}\left[a(Y_2)\nabla_1 h_0(Y_2) - \theta_0 - \Gamma(X)(Y_1 - h_0(Y_2))\right]$$

$$\Sigma(X) \equiv \text{Var}(\rho_2(Z, h_0) \mid X) = \text{Var}(Y_1 - h_0(Y_2)) \mid X).$$

We note that this is also a special case of Example 3.3 of AC12, which already characterized the efficiency bound for θ. We recall their result for the sake of easy reference, and compute

$$J_0 \equiv \inf_{r \in \overline{\mathcal{W}}} E\left\{\{\sigma_0\}^{-2}\left(1 + \mathbb{E}[a(Y_2)\nabla_1 r(Y_2) + \Gamma(X)r(Y_2)]\right)^2 + \Sigma(X)^{-1}\left(\mathbb{E}[r(Y_2) \mid X]\right)^2\right\} \tag{4}$$

where $\overline{\mathcal{W}} = \{r : \mathbb{E}[\Sigma(X)^{-1}(\mathbb{E}\{r(Y_2)|X\})^2] + (E\{a(Y_2)\nabla_1 r(Y_2) + \Gamma(X)r(Y_2)\})^2 < \infty\}$. Let $r_0 \in \overline{\mathcal{W}}$ be one solution (not necessarily unique) to the optimization problem (4). We note that such one solution always exists since the problem is convex, and we have:

$$J_0 = \frac{1 + \mathbb{E}[a(Y_2)\nabla_1 r_0(Y_2) + \Gamma(X)r_0(Y_2)]}{\sigma_0^2} \tag{5}$$

Remark 2.1. Characterization of Efficient Score. Applying Theorem 2.3 of AC12, we have: *the semiparametric efficient score* S^* for θ_0 in Model *(3)* is given by

$$S^*(Z) = \frac{1 + \mathbb{E}[a(Y_2)\nabla_1 r_0(Y_2) + \Gamma(X)r_0(Y_2)]}{\sigma_0^2}\varepsilon_1(Z, \alpha_0) + \frac{\mathbb{E}[r_0(Y_2)|X]}{\Sigma(X)}(Y_1 - h_0(Y_2))$$

where $r_0 \in \overline{\mathcal{W}}$ is one solution to *(4)*. And the semiparametric information bound for θ_0 is $J_0 \equiv \text{Var}(S^*)$.

(1) If $J_0 = 0$, then θ_0 can not be estimated at the $\sqrt{n}$-rate.

(2) If $J_0 > 0$, then the semiparametric efficient variance for θ_0 is: $\Omega_0 \equiv (J_0)^{-1}$.

In the rest of the paper we shall assume that $J_0 > 0$ and hence θ_0 is a $\sqrt{n}$ estimable regular parameter. We note that by definition, the efficient score (indeed any moment condition proportion to an influence function) automatically satisfies the orthogonal moment condition.

2.2 Efficient influence function equation based procedure

From the remark above, the semiparametric efficient influence function for θ_0 takes the form

$$\psi^*(Z,\theta_0) \equiv (J_0)^{-1} S^*(Z) = \varepsilon_1(Z,\theta_0,h_0) + (J_0)^{-1} \frac{\mathbb{E}[r_0(Y_2)|X]}{\Sigma(X)}(Y_1 - h_0(Y_2)), \tag{6}$$

Denote

$$\alpha_e(X) \equiv (J_0)^{-1} \frac{\mathbb{E}[r_0(Y_2)|X]}{\Sigma(X)}.$$

It is clear that θ_0 is the unique solution to the efficient IF equation $\mathbb{E}[\psi^*(Z,\theta_0)] = 0$, that is

$$\mathbb{E}\left[a(Y_2)\nabla_1 h_0(Y_2) - \theta - [\Gamma(X) - \alpha_e(X)](Y_1 - h_0(Y_2))\right] = 0 \quad iff \quad \theta = \theta_0.$$

One efficient estimator, $\hat{\theta}_{ES}$, for θ_0 is simply based on the sample version of the efficient IF equation with plug-in consistent estimates of all the nuisance functions:

$$\hat{\theta}_{ES} = n^{-1} \sum_{i=1}^{n} \left(a(Y_{2i})\nabla_1 \hat{h}(Y_{2i}) - [\hat{\Gamma}(X_i) - \hat{\alpha}_e(X_i)](Y_{1i} - \hat{h}(Y_{2i})) \right).$$

In this paper $\hat{h}(Y_2)$ can be various ANN sieve minimum distance estimators (see below), but, for simplicity, the nuisance functions $\hat{\Gamma}(X)$ and $\hat{\alpha}_e(X)$ are estimated by plug-in linear sieves estimators.

2.3 Optimally weighted SMD procedure

Another efficient estimator for θ_0 can be found by optimally-weighted sieve minimum distance, where the population criterion is (see AC12):

$$Q^0(\alpha) = \mathbb{E}[m'(Z,\alpha)W_0(X)m(Z,\alpha)] = \mathbb{E}\left[\frac{1}{\sigma_0^2}[\mathbb{E}(\varepsilon_1(Z,\alpha))]^2 + \frac{1}{\Sigma(X)}(\mathbb{E}[Y_1 - h(Y_2) \mid X])^2\right] \tag{7}$$

The discrepancy measure is the expectation of the two moment conditions, conditional on their respective σ-fields:

$$m(X,\alpha) = \begin{bmatrix} \mathbb{E}[\varepsilon_1(Z,\alpha)] \\ \mathbb{E}[Y_1 - h(Y_2) \mid X] \end{bmatrix}$$

and the optimal weight matrix $W_0(.)$ is diagonal and proportional to the inverse variance of each moment condition:

$$W_0(X) = \begin{bmatrix} 1/\sigma_0^2 & 0 \\ 0 & 1/\Sigma(X) \end{bmatrix}$$

A sieve minimum distance estimator for (h,θ) may be constructed by (i) replacing expectations with sample means, (ii) replacing conditional expectations with projection onto linear sieve bases, (iii) replacing the optimal weight matrix with a consistent estimator, and (iv) replacing the infinite

dimensional optimization with finite dimensional optimization over a sieve space for h. This paper focuses on approximating h by ANN sieves. In particular, a sample analogue of the above objective function is

$$\widehat{Q}_n^0(\alpha) \equiv \frac{1}{n}\sum_{i=1}^{n} \widehat{m}(X_i,\alpha)'[\widehat{W}_0(X_i)]\widehat{m}(X_i,\alpha)$$

where $\widehat{m}(.;.)$ and $\widehat{W}_0(.)$ are estimators of $m(.,.)$ and $W(.)$ respectively; see Sections 3 and 4 below for examples of different estimators. Let $\mathcal{H}_n$ be a sieve for h (and in this paper we focus on various ANN sieves). We define the optimally weighted SMD estimator $\hat{\alpha} = (\hat{\theta}, \hat{h})$ as an approximate solution to

$$\min_{\theta\in\Theta, h\in\mathcal{H}_n} \widehat{Q}_n^0(\theta, h).$$

This is an estimator proposed in AC12.

Two remarks are in order. First, note that the optimal weight matrix $W_0(X)$ is diagonal because ε_1 and ρ_2 are uncorrelated by design. Second, since the optimal weight matrix is diagonal and θ is a free parameter, we can view the minimization as sequential:

$$h_0 = \arg\min_h \mathbb{E}\left[\frac{1}{\Sigma(X)}(\mathbb{E}[Y_1 - h(Y_2) \mid X])^2\right], \quad \theta_0 = \mathbb{E}[a(Y_2)\nabla_1 h_0(Y_2) - \Gamma(X)(Y_1 - h_0(Y_2))].$$

This is important because solving the model sequentially while maintaining efficiency plays a role in computing the estimators.

We may analyze the asymptotic properties of this estimator. Since we may view the optimally weighted SMD problem as either a minimum distance program or a sequential GMM estimator, we may carry out two separate analyses of the asymptotic properties. The analysis of the estimator as a minimum distance problem is a specialization of Ai and Chen (2007, 2012, 2003); Chen and Pouzo (2015), while the analysis as a sequential moment restriction specializes Chen and Liao (2015) in Appendix **??**.

2.4 Analysis as optimally weighted SMD

We are interested in a functional of the parameter $\phi(\alpha) = \phi(\theta, h) = \theta$. A simple linearization shows that[8]

$$\sqrt{n}(\phi(\hat{\alpha}) - \phi(\alpha)) = \sqrt{n}(\hat{\theta} - \theta) \approx \sqrt{n}\frac{d\phi}{d\alpha}[\hat{\alpha} - \alpha].$$

A key insight is that we may define an inner product over the space of $\alpha = (\theta, h)$, $\mathcal{A}$, as

$$\begin{aligned}\langle u, v\rangle_{\text{MD}} &= \mathbb{E}\left[\frac{dm}{d\alpha}[u]' W_0(X)\frac{dm}{d\alpha}[v]\right] \\ &= \mathbb{E}\left[\frac{\mathbb{E}[-v_\theta + a(Y_2)\nabla_1 v_h + \Gamma(X)v_h]\mathbb{E}[-u_\theta + a(Y_2)\nabla_1 u_h + \Gamma(X)u_h]}{\sigma_0^2} + \frac{\mathbb{E}[v_h \mid X]\cdot\mathbb{E}[u_h \mid X]}{\Sigma(X)}\right],\end{aligned}$$

where $u = (u_\theta, u_h)$ and $v = (v_\theta, v_h)$.
The linear operator, $v \mapsto \frac{d\phi}{d\alpha}[v]$, then admits a Riesz representation $\frac{d\phi}{d\alpha}[v] = \langle v, v^\star\rangle_{\text{MD}}$, where $v^\star$ is the Riesz representer.

Next, we analyze the Riesz representer and the inner product. First, observe that by picking $v = (\tilde{\theta}, 0)$, we can simplify some terms in the Riesz representer:

$$\frac{d\phi}{d\alpha}[(\tilde{\theta}, 0)] = \tilde{\theta} = \langle(\tilde{\theta}, 0), v^\star\rangle_{\text{MD}} = -\frac{1}{\sigma_0^2}\mathbb{E}[-v_\theta^\star + a(Y_2)\nabla_1 v_h^\star + \Gamma(X)v_h^\star]\tilde{\theta}$$

which implies that $\frac{1}{\sigma_0^2}\mathbb{E}[-v_\theta^\star + a(Y_2)\nabla_1 v_h^\star + \Gamma(X)v_h^\star] = -1$.

We can now consider the inner product, which turns out has a representation as a sample mean, from a local expansion of the criterion function (for the precise argument, see Ai and Chen, 2003, 2007):

$$\begin{aligned}\langle v^\star, \hat{\alpha} - \alpha\rangle_{\text{MD}} &\approx -\frac{1}{n}\sum_{i=1}^n \frac{dm}{d\alpha}[v^\star]' W_0(X_i)\rho(Z_i, \alpha) \\ &= -\frac{1}{n}\sum_{i=1}^n -1\cdot(a(Y_2)\nabla_1 h(Y_{2i}) - \theta - \Gamma(X)(Y_{1i} - h(Y_{2i}))) - \frac{\mathbb{E}[v_h^\star \mid X_i]}{\Sigma(X_i)}(Y_{1i} - h(Y_{2i})) \\ &= \frac{1}{n}\sum_{i=1}^n a(Y_2)\nabla_1 h(Y_{2i}) - \theta - \Gamma(X)(Y_{1i} - h(Y_{2i})) + \frac{\mathbb{E}[v_h^\star \mid X_i]}{\Sigma(X_i)}(Y_{1i} - h(Y_{2i})). \qquad (8)\end{aligned}$$

[8] Let $\psi : V \to \mathbb{R}$ be a real-valued function whose domain is some topological vector space V. The notation $\frac{d\psi}{dv}[w]$ denotes the directional derivative of ψ with respect to v in the direction of w, which is

$$\frac{d\psi}{dv}[w] = \lim_{t\to 0}\frac{\phi(v + tw) - \phi(v)}{t}.$$

Note that when $\psi : \mathbb{R}^k \to \mathbb{R}$, the above definition is a usual directional derivative

$$\frac{d\psi}{dv}[w] = (\nabla_v \psi)' w.$$

We then have a heuristic derivation of the influence function

$$\sqrt{n}(\hat{\theta}-\theta_0)=\frac{1}{\sqrt{n}}\sum_{i=1}^{n}\left[a(Y_2)\nabla_1 h(Y_{2i})-\theta_0-\left(\Gamma(X_i)-\frac{\mathbb{E}[v_h^\star\mid X]}{\Sigma(X)}\right)(Y_{1i}-h_0(Y_{2i}))\right]+o_p(1). \quad (9)$$

Riesz Representer. Lastly, we need to characterize the Riesz representer $v^\star$. The argument in AC03 parametrizes $v^\star=v_\theta^\star(1,-w^\star)$ as a "scale times direction" coordinate. For a fixed scale $v_\theta^\star$, the minimum norm property of Riesz representers implicitly defines

$$w^\star=\arg\min_w\|(1,-w)\|_{\text{MD}}^2=\arg\min_w\mathbb{E}\left[\frac{1}{\sigma_0^2}(\mathbb{E}[1+a(Y_2)\nabla_1 w+\Gamma(X)w])^2+\frac{1}{\Sigma(X)}(\mathbb{E}[w\mid X])^2\right]. \quad (10)$$

Solving the condition

$$\frac{1}{\sigma_0^2}\mathbb{E}[-v_\theta^\star+a(Y_2)\nabla_1 v_h^\star+\Gamma(X)v_h^\star]=-1$$

by plugging in $v_h=-w^\star v_\theta^\star$ then yields

$$v_\theta^\star=\frac{\sigma_0^2}{\mathbb{E}[1+a(Y_2)\nabla_1 w^\star+\Gamma(X)w^\star]}\quad v_h^\star=\frac{-w^\star\sigma_0^2}{\mathbb{E}[1+a(Y_2)\nabla_1 w^\star+\Gamma(X)w^\star]}$$

as the solutions for the representers where $w^\star$ is defined in (10) above. If we assume completeness condition then$w^\star=r_0$ as the unique solution to (4) or (5) and $v_\theta^\star=(J_0)^{-1}$

The consistency, root-n asymptotic normality, consistent variance estimation can all be obtained by directly applying AC (2003, 2007) for single hidden layer ANN sieves. Chen, Liao and Wang (2021) results can be applied for multi-layer ANN sieves.

2.4.1 Identity weighted SMD

This is a special case of AC (2007, section 4.2). We include the asymptotic linear expansion for the sake of comparison. In particular, the influence function, associated with the identity-weighted SMD estimator, is of the form

$$\sqrt{n}(\hat{\theta}-\theta)=\frac{1}{\sqrt{n}}\sum_{i=1}^{n}\left[\nabla_1 h(Y_{2i})-\theta+\mathbb{E}[v_{h,\text{id}}^\star\mid X](Y_{1i}-h(Y_{2i}))\right]+o_p(1).$$

where

$$v_{h,\text{id}}^\star=\frac{-w_{\text{id}}^\star}{1+\mathbb{E}[\nabla_1 w_{\text{id}}^\star]}\quad w_{\text{id}}^\star=\arg\min_w\left\{\mathbb{E}[\mathbb{E}[w(Y_2)\mid X]^2]+(1+\mathbb{E}[\nabla_1 w(Y_2)])^2\right\}. \quad (11)$$

The consistency, root-n asymptotic normality, consistent variance estimation can all be obtained by directly applying AC (2007) for single hidden layer ANN sieves.

3 Implementation of the estimators

We now explain the implementation of various estimators for the average derivative as these tend to be complex, especially when we would like to estimate functionals of NPIV models efficiently. In this section, we describe in broad strokes the construction of the eventual estimators for the average derivative, which often involve estimation of nuisance parameters and functions. These nuisance parameters—which often take the form of known transformations of conditional means and variances—require further choice of estimation routines and tuning parameters, details of which are relegated to Section 4.2.

Quick map of estimation procedures We provide a simple map that connects the above models and approaches to estimators we use. For implementation details see Section 3 below.

1. For SMD estimators [P-ISMD, OP-OSMD]: Solve sample and sieve version of (7) (Section 3.1)
2. Standard error for SMD estimators: Estimate the components of the influence functions in (9), and take the sample variance. (Section 3.3)
3. Score estimators [IS, ES]: Estimate the components of the influence functions as in (**??**). Set the influence functions to zero and solve for θ. (Section 3.2)

Additionally, we describe the estimator when the analyst is willing to assume more semiparametric structure (e.g. partial linearity) on the structural function. We also conclude the section with a brief discussion of software implementation issues.

A note on notation Recall that we use Y_1 to denote the outcome, Y_2 to denote variables (endogenous or exogenous) that are included in the structural function, and X to denote exogenous variables that are excluded from the structural function. Certain entries of X and Y_2 may be shared. Again, the NPIV moment condition

$$\mathbb{E}\left[Y_1 - h_0(Y_2) \mid X\right] = 0 \tag{12}$$

and we are interested in $\theta_0 = \mathbb{E}\left[\nabla_1 h_0(Y_2)\right]$, where $\nabla_1 h_0(Y_2)$ is the partial derivative of h_0 with respect to its first argument, evaluated at Y_2. Let $Z = [Y_1, Y_2, X]$ collect the data, viewed as random variables in the population.

We also set up notation for objects related to the sample. Let there be a sample of n observations. We denote $y_1 \in \mathbb{R}^n, y_2 \in \mathbb{R}^{n\times p}, x \in \mathbb{R}^{n\times q}$ as vectors and matrices respectively of realized values of the random vector (Y_1, Y_2, X). We will slightly abuse notation and write $f(y_2)$, for a function $f : \mathbb{R}^p \to \mathbb{R}^d$, to be the $(n \times d)$-matrix of outputs obtained by applying f row-wise, and similarly

for expressions of the type $f(x)$.[9] For a vector valued function f, we let $P_f = f(x)(f(x)'f(x))^+f(x)'$ be the projection matrix onto the column space of $f(x)$.

3.1 Sieve minimum distance (SMD) estimators

Consider a linear sieve basis $\phi(\cdot)$ for X, where $\phi(X) \in \mathbb{R}^k$. For a sample of realizations $v \in \mathbb{R}^n$ of V, $P_\phi v$ is the sample best mean square linear predictor (that approximates the conditional mean) of v, since it returns the fitted values of a regression of v on flexible functions of x:

$$P_\phi v \approx [\mathbb{E}[V_1 \mid X_1], \ldots, \mathbb{E}[V_n \mid X_n]]'.$$

Under the NPIV restriction (12), taking $V = Y_1 - h_0(Y_2)$ and $v = y_1 - h_0(y_2)$, we should expect

$$P_\phi(y_1 - h_0(y_2)) \approx 0.$$

This motivates the analogue of the SMD criterion (7) in the sample, where we choose h so as to minimize the size of the projected residual $P_\phi(y_1 - h(y_2))$:

$$\hat{h} = \arg\min_{h \in \mathcal{H}} \frac{1}{n} \big\| P_\phi[y_1 - h(y_2)] \big\|^2. \tag{13}$$

When the norm chosen is the usual Euclidean norm $\|\cdot\| = \|\cdot\|_2$, we obtain the **identity-weighted SMD estimator** for h_0, $\hat{h}_{\text{ISMD}}$.

Given a preliminary estimator $\tilde{h}$ for h_0, we may form an estimator of the residual conditional variance $\Sigma_0(X) \equiv \mathbb{E}[(Y_1 - h_0(Y_2))^2 \mid X]$ by forming the estimated residuals $y_1 - \tilde{h}(y_2)$ and then projecting $(y_1 - \tilde{h}(y_2))^2$ onto x, e.g. via the linear sieve basis $\phi(x)$ or via other nonparametric regression techniques such as nearest neighbors. With such an estimator of the heteroskedasticity, we can form a weight matrix $\hat{W} = \text{diag}(\hat{\Sigma}(x))^{-1}$. Using the norm $\|z\|_W^2 \equiv z'Wz$ in (13) yields the **optimally-weighted SMD estimator** for h_0, $\hat{h}_{\text{OSMD}}$.

With an estimated $\hat{h}$ of the structural function h_0, we can form two plug-in estimators of θ. The first is the **simple plug-in estimator**:

$$\hat{\theta}_{\text{SP}}(\hat{h}) = \frac{1}{n} \sum_{i=1}^{n} \nabla_1 \hat{h}(y_{2i}).$$

The simple plug-in estimator does not take into account the covariance between the two moment conditions, $Y_1 - h(Y_2)$ and $\nabla_1(Y_2) - \theta$. The second estimator, the **orthogonalized plug-in** esti-

[9] This notation conforms with how vector operations are broadcast in popular numerical software packages, such as Matlab and the Python scientific computing ecosystem (NumPy, SciPy, PyTorch, etc.).

mator, orthogonalizes the second moment against the first:

$$\hat{\theta}_{\text{OP}}(\hat{h}, \hat{\Gamma}) = \frac{1}{n}\sum_{i=1}^{n}[\nabla_1 \hat{h}(y_{2i}) - \hat{\Gamma}(x_i)(y_{1i} - \hat{h}(y_{2i}))],$$

where $\hat{\Gamma}$ is an estimator of the population projection coefficient of the second moment $\nabla_1 h_0(Y_2) - \theta_0$ onto the first moment condition $Y_1 - h_0(Y_2)$:

$$\Gamma(X) \equiv \mathbb{E}[(\nabla_1 h_0(Y_2) - \theta_0)(Y_1 - h_0(Y_2)) \mid X]\Sigma_0^{-1}(X). \tag{14}$$

One choice of $\hat{\Gamma}$ is to plug in sample counterparts—plugging in $\hat{h}$ for h_0, plugging in a preliminary $\hat{\theta}$ (which could be the $\hat{\theta}_{\text{SP}}(\hat{h})$) for θ_0, and plugging in an estimator $\hat{\Sigma}$ for Σ_0—and finally approximate $\mathbb{E}[\cdot \mid X]$ via a linear sieve regression, say with the basis $\phi(\cdot)$.

To summarize, the SMD estimator can be implemented as follows.

Identity Weighted SMD Estimator of $h(.)$

1. *Sieve for Conditional Expectation*: Choose a Sieve basis $\phi(.)$ for X: $\phi(.) \in \mathbb{R}^k$ (more details on this later)
2. *Construct Objective function*
   (a) Obtain $P_\phi(y_1 - h(y_2))$ the sample least squares projection of $(y_1 - h(y_2))$ onto ϕ.
   (b) *Optimizing* $h(.)$*:* define $\hat{h} = \arg\min_{h\in\mathcal{H}} \frac{1}{n}\left\|P_\phi[y_1 - h(y_2)]\right\|_2^2$.

Optimal SMD Estimator of $h(.)$

1. Same as Step (1) above
2. *Estimate Weight Function* Σ: with a preliminary estimator $\tilde{h}$ of h (use id weighted one for instance), form an estimator $\hat{\Sigma}(x)$ by projecting $(y_1 - \tilde{h}(y_2))^2$ on $\phi(.)$, the sieve basis for X to obtain $P_\phi((y_1 - \tilde{h}(y_2)^2)$. Form $\hat{W} = \text{diag}(\hat{\Sigma}(x))^{-1}$.
3. *Optimizing* $h(.)$*:* define $\hat{h} = \arg\min_{h\in\mathcal{H}} \frac{1}{n}\left\|P_\phi[y_1 - h(y_2)]\right\|_{\hat{W}}^2$.

Estimators for θ_0

1. *Simple Plug In Estimator.* Given an estimator $\hat{h}$ of h, use

$$\hat{\theta}_{\text{SP}}(\hat{h}) = \frac{1}{n}\sum_{i=1}^{n}\nabla_1 \hat{h}(y_{2i})$$

2. *Orthogonalized Plug In Estimator*

   (a) Obtain an estimator of Γ. One can use $\hat{\Gamma}(\hat{\theta}, \hat{h}) = P_\phi[(\nabla_1 \hat{h}(Y_2) - \hat{\theta})(Y_1 - \hat{h}(Y_2))]\hat{\Sigma}^{-1}(X)$ with $\hat{\theta}$ being for example the simple plug in estimator and $\hat{\Sigma}(x)$ the above estimator of the variance of the first moment.

   (b) *Orthogonal Plug In Estimator.* Obtain

$$\hat{\theta}_{\text{OP}}(\hat{h}, \hat{\Gamma}) = \frac{1}{n}\sum_{i=1}^{n}[\nabla_1 \hat{h}(y_{2i}) - \hat{\Gamma}(x_i)(y_{1i} - \hat{h}(y_{2i}))]$$

Combining simple plug-in with identity-weighted SMD yields the estimation procedure that we term P-ISMD, and combining orthogonal plug-in with optimally weighted SMD yields the estimation procedure that we call OP-OSMD.

3.2 Influence function-based estimators

We also implement influence function based estimators. As we highlighted in the previous section, one influence function estimator for θ_0 takes the following form

$$\psi(Z, \theta, h, \kappa) = \nabla_1 h(Y_2) - \kappa(X)(Y_1 - h(Y_2)) - \theta. \tag{15}$$

with $\kappa(.)$ defined below. Moreover, given an estimator $\hat{h}$ for h and $\hat{\kappa}$ for κ, we can form the influence function estimator:

$$\hat{\theta}(\hat{h}, \hat{\kappa}) = \frac{1}{n}\sum_{i=1}^{n}[\nabla_1 \hat{h}(y_{2i}) - \hat{\kappa}(x_i)\left(y_{1i} - \hat{h}(y_{2i})\right)].$$

Identity score estimator (IS) One influence function, which corresponds to the influence function of the P-ISMD estimator has κ taking the following form. We refer to the resulting influence function estimator as IS, for *identity score*.

$$\kappa_{\text{ID}}(X) = \mathbb{E}[-v^\star(Y_2) \mid X] \tag{16}$$

$$v^\star(Y_2) = \frac{-w^\star(Y_2)}{1 + \mathbb{E}[\nabla_1 w^\star(Y_2)]} \tag{17}$$

$$w^\star(Y_2) = \arg\min_w \left\{\mathbb{E}\left[(\mathbb{E}[w(Y_2) \mid X])^2\right] + (1 + \mathbb{E}[\nabla_1 w(Y_2)])^2\right\}. \tag{18}$$

Efficient score estimator (ES) On the other hand, the efficient influence function (ES) uses a different $\kappa(\cdot)$:

$$\kappa_{\text{EIF}}(X) = \Gamma(X) - \mathbb{E}[v^\star(Y_2) \mid X]\Sigma(X)^{-1},$$

where $\Gamma(\cdot)$ is as in (14), and

$$v^\star(Y_2) = \frac{-w^\star}{\mathbb{E}[1 + \nabla_1 w^\star + \Gamma(X) w^\star(Y_2)]} \operatorname{Var}\left[\nabla_1 h_0 - \theta_0 - \Gamma(X)(Y_1 - h_0(Y_2))\right] \tag{19}$$

$$w^\star(Y_2) = \arg\min_w \left\{ \mathbb{E}\left[\Sigma(X)^{-1} \left(\mathbb{E}[w(Y_2) \mid X]\right)^2\right] + \left(\frac{1 + \mathbb{E}[\nabla_1 w(Y_2) + \Gamma(X) w(Y_2)]}{\sqrt{\operatorname{Var}\left[\nabla_1 h_0 - \Gamma(X)(Y_1 - h_0(Y_2)) - \theta_0\right]}} \right)^2 \right\} \tag{20}$$

are the same as (10), which are also weighted analogues of the identity weighted $w^\star$, (18).

The above formulation writes $v^\star$ as a function of $w^\star$; alternatively, we may follow the strategy in **??** and estimate $v^\star$ directly. One way to estimate the above representer and hence get a feasible score is as follows. Recall that the definition of $v^\star$ is

$$\mathbb{E}[\mathbb{E}[v^\star \mid X] \Sigma(X)^{-1} \mathbb{E}[v \mid X]] = \mathbb{E}[\nabla_1 v + \Gamma(X) v] \quad \|v^\star\|_{\rho_2}^2 = \sup_v \frac{(\mathbb{E}[\nabla_1 v + \Gamma(X) v])^2}{\mathbb{E}[\Sigma(X)^{-1} \mathbb{E}[v \mid X]^2]}.$$

Let $\nu(Y_2)$ be the basis approximating Y_2. Suppose we view that $v^\star$ is well approximated by $\nu(Y_2)'\beta$, and that $\mathbb{E}[\cdot \mid X]$ is well approximated by projection onto a basis $\lambda(x)$, then the above definition of $v^\star$ yields a finite-dimensional problem that we may solve in closed form to obtain the following.

Consider the following quantities

$$F = \mathbb{E}[\nabla_1 \nu(Y_2) + \Gamma(X) \nu(Y_2)] \text{ and } R = \mathbb{E}\left[\Sigma(X)^{-1} \mathbb{E}[\nu \mid X] \mathbb{E}[\nu \mid X]'\right].$$

This then implies that $v^\star = \nu' R^- F$. In sample, this amounts to

$$\hat{F} = \frac{1}{n} \sum_i \left[\nabla_1 \nu(y_{2i}) + \hat{\Gamma}(x_i) \nu(y_{2i})\right] \text{ and } \hat{R} = \frac{1}{n} \sum_i \left[\hat{\Sigma}(x_i)^{-1} P_\lambda(x_i) \nu(y_{2i}) (P_\lambda(x_i) \nu(y_{2i}))'\right] \tag{21}$$

These can then be used to obtain $\hat{v}^\star$ and the influence function correction term

$$\kappa_{\text{EIF}}(X) = \Gamma(X) - \mathbb{E}[v^\star(Y_2) \mid X] \Sigma(X)^{-1}.$$

3.3 Inference for P-ISMD, OP-OSMD, IS, ES

We now discuss how to compute standard errors and confidence intervals—again in broad strokes—for the estimating algorithms P-ISMD, OP-OSMD, IS, ES. In a nutshell, for the score estimators IS and ES, the estimator $\hat{\theta}$ is a sample mean of *estimated* influence functions, and its sample variance is directly the properly normalized variance of the influence functions. As a result, under appropriate conditions, a sample variance of the estimated influence functions is consistent for the

variance of the influence functions, leading to consistent estimation of standard errors. For the estimators IS and ES, practitioners can therefore compute the standard errors without adjusting for the estimation of the nuisance parameters.

Similarly, estimating the standard errors for the P-ISMD and OP-OSMD estimators amounts to estimating the variance of the influence function. One approach is to simply use the influence function estimates from IS and ES, and leverage the fact that (P-ISMD, IS) and (OP-OSMD, ES) are respectively asymptotically equivalent.

Another approach is to estimate the variance of the influence functions directly, without necessarily estimating the influence functions themselves. The details are stated in Section 2, and we may turn the theory into estimators by "putting hats on parameters": replacing unknown functions with their finite-dimensional sieve approximations, conditional expectation with sieve projections, and expectations and variance with their sample counterparts. For convenience, we reproduce the calculation here:

1. P-ISMD: Consider

$$w^\star(Y_2) = \arg\min_w \left\{ \mathbb{E}\left[(\mathbb{E}[w(Y_2) \mid X])^2 \right] + (1 + \mathbb{E}[\nabla_1 w(Y_2)])^2 \right\}$$

   which is the same as (11) and (18). Let $D_{w^\star}(X) = [-1 - \mathbb{E}[\nabla_1 w^\star], \mathbb{E}[w^\star \mid X]]'$. Then the asymptotic variance is

$$V = \frac{\mathbb{E}[\|D_{w^\star}(X)\|^2]^2}{\mathbb{E}\left[\|D_{w^\star}(X)\|^2 (Y_1 - h_0(Y_2))^2\right]}$$

2. OP-OSMD: The inverse of the asymptotic variance is

$$V^{-1} = \min_w \left\{ \mathbb{E}\left[\Sigma(X)^{-1} (\mathbb{E}[w(Y_2) \mid X])^2 \right] + \left(\frac{1 + \mathbb{E}[\nabla_1 w(Y_2) + \Gamma(X) w(Y_2)]}{\sqrt{\mathrm{Var}\left[\nabla_1 h_0 - \Gamma(X)(Y_1 - h_0(Y_2)) - \theta_0\right]}} \right)^2 \right\}$$

   which corresponds to the objective function in (10)

A third approach, which in our experience seems more accurate than analytic standard errors, is a multiplier bootstrap for the SMD estimators. The bootstrap simply replaces the residual $y_1 - h(y_2)$ in (13) with the weighted residuals $\omega(y_1 - h(y_2))$ where $\omega = \mathrm{diag}(\omega_1, \ldots, \omega_n)$ are such that $\omega_i \overset{\text{i.i.d.}}{\sim} F_\omega$, independently of data, for some positively supported distribution F_ω with unit mean and variance (e.g. the standard Exponential distribution). Given a realization of the bootstrap weights ω, the estimation routines P-ISMD and OP-OSMD would yield an estimate for θ. Repeating this procedure a large number of times would generate a large number of bootstrapped estimates, whose percentiles form confidence interval boundaries.

3.4 Partially linear or partially additive SMD estimators

Assume h_0 is partially linear in its first argument, or, additionally, partially additive in subsets of its arguments. Since h_0 is linear in its first argument, the slope on that argument is the average derivative θ_0. Therefore, under such a restriction, h_0 can be identified with the pair (θ_0, ϑ_0) where ϑ_0 is some nuisance parameter governing the rest of the function.

As in the case with SMD estimators in the nonparametric case, we solve the SMD problem (13), while constraining $\mathcal{H}$ to conform to the functional form assumptions made. The parameter θ_0 is estimated via direct plug-in, since a solution $\hat{h} = (\hat{\theta}, \hat{\vartheta})$ for (13) naturally produces an estimator $\hat{\theta}$ for θ_0 (Ai and Chen, 2003).

3.5 Implementation of neural networks

We now provide a brief recipe on working with neural networks. A feedforward neural network is a composition of *layers* of the form[10]

$$f_{\sigma,W,b} : \mathbb{R}^m \to \mathbb{R}^n \quad x \mapsto \sigma(Wx + b) \quad \sigma : \mathbb{R} \to \mathbb{R} \text{ is applied entry-wise.}$$

for some conformable matrix W, vector b, and nonlinear *activation function* σ:, i.e. a k-hidden-layer neural network has the representation

$$h_\eta : \mathbb{R}^m \to \mathbb{R}^n \quad h = b_{k+1} + W_{k+1} \cdot (f_{\sigma_k,W_k,b_k} \circ \cdots \circ f_{\sigma_1,W_1,b_1})$$

where we collect the *learnable parameters* $\{W_j, b_j : j = 1, \ldots, k+1\}$ as η. The gradient $\nabla_\eta h_\eta(y_2)$ can be computed efficiently using the celebrated backpropagation algorithm, and, as a result, in practice, neural networks are often optimized via first-order methods such as (stochastic) gradient descent or its variants, such as the popular Adam algorithm (Kingma and Ba, 2014) in the machine learning community. Optimization with neural networks is easiest with an unconstrained, differentiable objective, for which numerous computational frameworks exist. We use PyTorch (Paszke *et al.*, 2017) in this paper[11]. In particular, (13) is an unconstrained, differentiable objective function, and we may optimize over η since the overall gradient may be decomposed into components that are efficiently computed: By the chain rule,

$$\nabla_\eta L(h, y_1, y_2, x) = \nabla_h L \cdot \nabla_\eta h,$$

where $L(\cdot, \cdot, \cdot, \cdot)$ denote the objective function (13).

[10] For instance, a ReLU layer is a function of the form

$$x \mapsto \max(0, Wx + b)$$

for W a conformable matrix and b a conformable vector.

[11] See https://pytorch.org/

Compared to conventional numerical linear algebra packages such as NumPy or MATLAB, PyTorch offers two computational advantages particularly suited for deep learning: automatic differentiation and Graphical Processing Unit (GPU) integration. PyTorch tracks the history of computation steps taken to produce a certain output, and automatically computes analytic gradients of the output with respect to its inputs (See Listing 1 for an example). Autodifferentiation allows gradient descent methods to be carried out conveniently, without the user supplying analytical or numerical gradient calculations manually.

PyTorch also allows arithmetic operations to be computed on GPUs, which have computing architecture that allows for large-scale parallelization of simple operations. For instance, multiplying two $k \times k$ matrices is of order $O(k^3)$ with a naive algorithm, which can be viewed as k^2 dot products of size k; GPUs allow for parallelized computing of the k^2 dot product operations, in contrast to CPUs, where the level of parallelism is determined by the number of CPU cores. For optimization, we use the Adam algorithm (Kingma and Ba, 2014), which is an enhancement of basic gradient descent by estimating higher order gradients.

Listing 1: Example of automatic differentiation in PyTorch

```
>>> import torch
>>> a = torch.tensor([1.], requires_grad=True)
>>> b = torch.tensor([2.], requires_grad=True)
>>> c = (a * b)
>>> c # we expect c = a * b = 2
tensor([2.], grad_fn=<MulBackward0>)
>>> c.backward() # Compute dc/da and dc/db
>>> a.grad # dc/da = 2
tensor([2.])
>>> b.grad # dc/db = 1
tensor([1.])
```

4 Monte Carlo Studies and Empirical Illustrations

4.1 Monte Carlo Designs

We have considered many Monte Carlo designs. For the sake of space, we report three Monte Carlo designs in the paper.

Monte Carlo 1. *We consider the following data-generating process, which may be viewed as an augmentation of the Monte Carlo example in Chen (2007):*

$$Y = X_1\theta + h_{01}(R) + h_{02}(X_2) + h_{03}(\tilde{X}) + U, \quad \mathbb{E}[U \mid X_1, X_2, X_3, \tilde{X}] = 0, \quad \theta_0 = 1$$

where we generate

$$
\begin{aligned}
h_{01} : \mathbb{R} \to \mathbb{R} \quad & t \mapsto \frac{1}{1+\exp(-t)} \\
h_{02}(t) : \mathbb{R} \to \mathbb{R} \quad & t \mapsto \log(1+t) \\
h_{03} : \mathbb{R}^{d_{\tilde{x}}} \to \mathbb{R} \quad & \tilde{x} \mapsto 5\tilde{x}_1^3 + \tilde{x}_2 \cdot \max_{j=1,\ldots,d_{\tilde{x}}} (\tilde{x}_j \vee 0.5) + 0.5\exp(-\tilde{x}_{d_{\tilde{x}}}) \\
X_1, X_2, X_3 &\sim \mathrm{Unif}[0,1] \\
U \mid X_1, X_2, X_3 &\sim \mathcal{N}\left(0, \frac{1}{3}(X_1^2 + X_2^2 + X_3^2)\right) \\
\epsilon &\sim \mathcal{N}(0, 0.1) \\
R &= X_1 + X_2 + X_3 + 0.9U + \epsilon.
\end{aligned}
$$

The process generating $\tilde{X}$ is somewhat complex. First, we generate a covariance matrix $\Sigma \propto (I + Z'Z)$, normalized to unit diagonals, where Z's entries are i.i.d. standard Normal. The seed generating the covariance matrix is held fixed over different samples, and so Σ should be viewed as fixed a priori. Next, let $\rho \in [-1, 1]$ denote a correlation level and we let

$$
\tilde{X} = \Phi\left(\rho(X_1 + X_2 + X_3) + \sqrt{1-\rho^2}T\right) \quad T \sim \mathcal{N}(0, \Sigma), \tag{22}
$$

where $\Phi(\cdot)$ is the standard Normal CDF, and $\Phi(\cdot)$ and addition are applied elementwise. In the exercises reported, we use $\rho \in \{0, 0.5\}$ for correlation levels. This Monte Carlo design becomes identical to the one used in Chen (2007) when $\tilde{X}$ is an empty vector. We increase the dimension of $\tilde{X}$ to make the estimation problem more difficult.

To connect with the notation in the implementation section, here $Y_{1i} = Y$ and $Y_{2i} = [X_1, R, X_2, \tilde{X}]$, and $X_i = [X_1, X_2, X_3, \tilde{X}]$, where the moment restriction is

$$
\mathbb{E}[Y_{1i} - h_0(Y_{2i}) \mid X_i] = 0,
$$

and we treat the parameter as $\theta_0 = \mathbb{E}\left[\frac{\partial h_0}{\partial X_1}\right]$, without assuming any additive structure. Note here that this design allows for correlation among regressors both endogenous and exogenous. It also allows for heteroskedasticity and possibly large dimensions by increasing the dimension of $\tilde{X}$.

Monte Carlo 2. *Our second Monte Carlo design is an augmentation of that in Chen and Qiu (2016). The data-generating process is*

$$
Y = R_1\theta + h_{01}(R_2) + h_{02}(X_2) + h_{03}(\tilde{X}) + U \quad \mathbb{E}[U \mid X_1, X_2, X_3, \tilde{X}] = 0 \quad \theta_0 = 1
$$

where Φ is the CDF of the standard normal distribution. Moreover, the law of the data is described

by

$$
\begin{aligned}
\{h_{0j} : j = 1, 2, 3\} \text{ are the same as in Monte Carlo 1} \\
R_1 = X_1 + 0.5U_2 + V \quad R_2 = \Phi(V_3 + 0.5U_3) \\
X_2 \sim \text{Unif}[0, 1] \quad X_1 = \Phi(V_2) \quad X_3 = \Phi(V_3) \\
U = \frac{U_1 + U_2 + U_3}{3} \cdot \sigma(X_1, X_2, X_3) \\
\sigma(X_1, X_2, X_3) = \sqrt{\frac{X_1^2 + X_2^2 + X_3^2}{3}} \\
U_\ell, V_k \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, 1), \quad \ell = 1, 2, 3, k = 2, 3 \\
V \sim \mathcal{N}(0, (\sqrt{0.1})^2).
\end{aligned}
$$

$\tilde{X}$ is generated as in (22), with the correlation ρ set to $\{0, 0.5\}$.

Monte Carlo 3. *We modify Monte Carlo 2 with two changes*

(a) R_1 enters quadratically as R_1^2 in the structural equation. Note that $\theta_0 = \mathbb{E}[2R_1] = 1$.

(b) R_1 enters through $R_1^2/2 + R_1 \frac{f(a(X_2-b))}{2C}$, where

$$f(t) = h_{01}(t)(1 - h_{01}(t)) \quad h_{01}(t) = \frac{1}{1 + e^{-t}}.$$

and $C = \int_0^1 f(a(r-b))\, dr$, $a = -1$, and $b = 16$. As a result, the average derivative is

$$\mathbb{E}\left[R_1 + \frac{f(a(X_2 - b))}{2C}\right] = \frac{1}{2} + \frac{1}{2} = 1.$$

4.2 Walkthrough of implementation details

We detail here the choice of estimators that we used in addition to details about how the standard errors were constructed. A overview is presented in Table 5.

1. Figure 1 reports Monte Carlo means and standard deviations for the design in Monte Carlo 1, with SMD estimators for different choices of tuning parameters. In particular, we make the following choices.

   (a) Identity-weighted SMD with simple plug-in: $\hat{\theta}_{\text{SP}}\left(\hat{h}_{\text{ISMD}}\right)$ defined in Section 3.1. We specify choices of the linear sieve basis $\phi(\cdot)$

      i. $\phi(X) = [\phi_1(X_1, X_2, X_3), \phi_2(X, \tilde{X})]$, where $\phi_1(X_1, X_2, X_3)$ follows the basis choice made in Chen (2007) (p.5581–5582),[12] and $\phi_2(X, \tilde{X}) = [\tilde{X}, \tilde{X}^2, (X_i \tilde{X}_j)_{i,j}]$ contains

[12] i.e. $\phi_1(X_1, X_2, X_3) = [1, X_1, X_1^2, X_1^3, X_1^4, (X_1 - 0.5)_+^4, X_2, \ldots, X_2^4, (X_2 - 0.5)_+^4, X_3, \ldots, X_3^4, (X_3 - 0.1)_+^4, (X_3 - 0.25)_+^4, (X_3 - 0.5)_+^4, (X_3 - 0.75)_+^4, (X_3 - 0.9)_+^4, X_1 X_3, X_2 X_3, X_1 (X_3 - 0.25)_+^4, X_2 (X_3 - 0.25)_+^4, X_1 (X_3 - 0.75)_+^4, X_2 (X_3 - 0.75)_+^4 \cdot]$, where $(\cdot)_+ = \max(\cdot, 0)$.

second-order polynomials for $\tilde{X}$ and interactions $X_i\tilde{X}_j$.

(b) Optimally-weighted SMD with orthogonalized plug-in: $\hat{\theta}_{\text{OP}}\left(\hat{h}_{\text{OSMD}}, \hat{\Gamma}\right)$. We specify estimation details for the nuisances Σ_0, Γ_0:

i. Form the squared residuals from the identity-weighted estimator $v \equiv (y_1 - \hat{h}_{\text{ISMD}}(y_2))^2$ and estimate k-nearest neighbors.

ii. $\hat{\Gamma}(\cdot)$: Since we know how to estimate $\hat{\Sigma}$, it suffices to estimate

$$\mathbb{E}[(\nabla_1 h_0(Y_2) - \theta_0)(Y_1 - h_0(Y_2)) \mid X]$$

We form $u \equiv \left[\nabla_1 \hat{h}_{\text{OSMD}}(y_2) - \hat{\theta}_{\text{SP}}\left(\hat{h}_{\text{ISMD}}\right)\right]\left(y_1 - \hat{h}_{\text{OSMD}}(y_2)\right)$ and project on $\phi(X)$: i.e. $[\hat{\Gamma}(x_1), \ldots, \hat{\Gamma}(x_n)]' \equiv (P_\phi(\hat{\Sigma}^{-1}u))$

2. Figure 2 reports Monte Carlo means and standard deviations for the design in Monte Carlo 2, using SMD estimators under a variety of assumptions.

(a) The first column of Figure 2 reports results where the SMD estimators do not assume any special structure of the structural function h. Estimator choices are the same as in Figure 1.

(b) The next three columns of Figure 2 follow Section 3.4 in that we maintain the partially linear structure. In particular, the second column maintains that the structural function is of the form $R_1\theta + h(R_2, X_2, \tilde{X})$, and the third and fourth columns maintain that it is of the form $R_1\theta + h_1(R_2) + h_2(X_2) + h_3(\tilde{X})$. The estimation of $\hat{\Sigma}(\cdot)$ required for OSMD is the same as in Figure 1.

(c) The third column uses neural networks to approximate the scalar functions h_1, h_2, whereas the fourth column uses splines to approximate h_1, h_2.

3. Figures 3 and 4 reports Monte Carlo means and standard deviations for a wide class of estimators (not limited to SMD) for Monte Carlo 2.

(a) Neural net SMD: Follow Item 1.

(b) Spline SMD:

i. Let $\lambda(x)$ be a spline basis for the instruments and let $\nu(y_2)$ be a spline basis for the endogenous regressors. Both λ and ν are of the form where each entry expands into a Spline$(k, 2)$ basis,[13] and pairwise interactions (of the form $x_i x_j$) are included in lieu of tensor product splines. The choice of k for λ is 1 more than that for ν.

[13] This notation is for a spline with 2 knots, where, between adjacent knots, the spline function is a polynomial of order $k - 1$.

ii. Given λ, ν, we estimate P-ISMD, OP-OSMD exactly as in Item 1, where we optimize over candidate structural functions of the form $\nu(\cdot)'\gamma$.

(c) Influence function estimators: Let $\lambda(x), \nu(y_2)$ be the bases used for the spline SMD estimators

i. IS:

A. Estimate $\hat{h}_{\text{ISMD}}$ as in 1(a).

B. Given $\hat{v}^\star(y_2)$, we estimate $\hat{\kappa}_{\text{ID}}$ with $\hat{\kappa}_{\text{ID}}(x) = P_\lambda \cdot \hat{v}^\star(y_2)$.

C. $v^\star(y_2)$ can be computed by solving (18). To do so, we approximate $w^\star(y_2)$ with $\nu(y_2)\beta$ for some coefficients β, and the $\mathbb{E}[\cdot \mid X]$ operator with P_λ. Doing so makes (18) a least-squares problem. In fact, the closed form solution is

$$\hat{\beta} = -\left(\frac{1}{n}\nu(y_2)'P_\lambda\nu(y_2) + \frac{1}{n^2}\nabla_1\nu(y_2)11'\nabla_1\nu(y_2)\right)^{-1}\left(\frac{1}{n}[\nabla_1\nu(y_2)]'1\right),$$

where $\nabla_1\nu(y_2) \in \mathbb{R}^{n\times d_\nu}$ takes the partial derivative entry-wise. Therefore $\nu(y_2)\hat{\beta}$ is the estimator for $w^\star$, and this gives an estimator for $v^\star$ by plugging in.

ii. ES:

A. Estimate $\hat{h}, \hat{\Gamma}$ with Item 1b.

B. Estimate $v^\star$ by (21).

C. Form

$$\hat{\kappa}_{\text{EIF}}(x) = \hat{\Gamma}(x) - P_\lambda[\hat{v}^\star(y_2)]\hat{\Sigma}(x)^{-1}$$

D. Finally, plug $\hat{\kappa}_{\text{EIF}}(x)$ and $\hat{h}$ to the influence function and compute $\hat{\theta}_{\text{EIF}}$.

(d) AGMM: Uses the implementation of Dikkala, Lewis, Mackey and Syrgkanis (2020).

4.3 Monte Carlo Results

We have implemented many more simulation results. Due to the length of the paper, we report representative simulation results in a sequence of figures below. Figure 1 plots the performance of various ANN SMD estimators in terms of mean ± one (Monte Carlo) standard deviation across 1000 replications for Monte Carlo 1, in which the first element of Y_2 is exogenous (X_1). As a reminder, P-ISMD is the simple plug in estimator of θ with identity weighting, while OP-OSMD is the orthogonalized plug in with optimal weighting for the SMD objective. As we can see across layers and activation function, and whether we have a low dimensional regime in the left hand side columns or large dimensional regimes in the right hand columns, or whether there is correlation across regressors (denoted by Y(es) or N(o) on top of each column), the behavior of these ANN estimator is similar and adequate. All the intervals are more or less centered on top of the truth, $\theta_0 = 1$, while the efficient estimator OP-OSMD is slightly less biased.

The rest of the figures correspond to more difficult Monte Carlo designs 2 and 3 where the first element of Y_2 is endogenous (R_1). Figure 2 reports the performance of various ANN SMD estimators for θ in Monte Carlo 2. The top display plots the results for $n = 1000$ and the bottom for $n = 5000$. Note here that the columns correspond to various assumptions we maintain on what the econometrician knows about the true structure of $h_0(.)$ in the model $\mathbb{E}[Y_1 - h_0(Y_2)|X] = 0$. The true design is partially additive, and the first column, NP, assumes that the econometrician has no knowledge of the true structure. As we can see, across all implementations (the rows), most of the ANN SMD estimators perform well, which indicates that ANNs seem able to adapt to the unknown structure of h_0. The second column labeled PL (for partially linear) assumes that $h_0(Y_2)$ is partially linear (i.e., $h_0(Y_2) = \theta R_1 + h(R_2, X_2, \tilde{X})$) while the third column labeled PA assumes the correct additive structure (i.e., $h_0(Y_2) = \theta R_1 + h_1(R_2) + h_2(X_2) + h_3(\tilde{X})$) in the Monte Carlo design is known to the econometrician (but the functions h_1, h_2, h_3 within it are of course not known). PA column corresponds to the case where we use ANN sieves to learn all the unknown functions h_1, h_2, h_3 although h_1, h_2 are functions of scalar random variable. Its performance slightly deteriorates as compared to the NP and PL columns. Notice here that for comparison, the last column for the PA case uses splines to approximate the two scalar valued unknown functions h_1 and h_2 while h_3 is always estimated via ANN (since it is of higher dimensions (at least when $\dim(\tilde{X}) > 0$). We see that the spline results are in line with the PL and NP results, and are adequate here.

In Figures 3 and 4, we compare various implementations of ANN estimators and spline estimators in Monte Carlo 2. In Figure 3, we compare identity-weighted estimators (IS, P-ISMD, AGMM, IS-X). In Figure 4, we compare optimally-weighted estimators that are semiparametrically efficient under suitable regularity conditions (ES, OP-OSMD, ES-X).[14] It is important at the outset to keep in mind that all ANN implementations require some non-negligible tuning as the optimization problem is non-convex and the problem itself with endogeneity, correlation among the regressors, and high dimensions is not easy to tune. Also, currently and for NPIV models, there is no theory for data driven approaches to picking width, depth, or activation functions and finite sample behavior in our design varied (For linear splines, there are data-driven choice of sieve terms, see Chen, Christensen and Kankanala, 2021).[15] Also, P-ISMD and OP-OSMD are the plug in and optimal plug in SMD estimators. The results across various combinations of $\dim(\tilde{X})$ and correlations for $n = 1000, 5000$ indicate first that ANN OP-OSMD and especially spline estimators seem to behave best. In particular, spline estimators require little tuning and are more stable than all ANN based estimators. The SMD ANN estimators are adequate with slight bias for the single-layer, varying-width case. IS and ES ANN estimators are generally less biased and slightly higher variance than P-ISMD and OP-OSMD ANN estimators, but we note that good performance of ES (in the ANN case) is quite sensitive to the choice of $\Sigma(X)^{-1}$ in the score—Figure 10 compares the performances

[14] As a reminder, we consider the following estimators: IS or identity weighted score estimator, ES or the efficient score estimator, while IS-X and ES-X are score estimators with two-fold cross fitting.

[15] although we have not implemented any data-driven choice of spline sieve terms in our paper.

of a variety of choices for $\hat{\Sigma}(X)$.

In Figure 5 we examine various standard error approaches for a set of estimators in Monte Carlo 2. For each of these, we compute the MC standard deviation, a feasible estimator based on the estimator variance derived from theory, and a bootstrapped standard error. Overall, the theory and bootstrapped standard errors are adequate. In unreported results, criterion (SMD) based bootstrap confidence intervals showed reasonable coverage performance.

In Figure 6, we report results for the various estimators for Monte Carlo 3, where the unknown function h_0 is now nonlinear in the endogenous R_1 (the first element of Y_2). In the top panel (a), we report results for the case with $R^2/2$ and panel (b) reports results for the case where the unknown function is $R_1^2/2 + R_1 f(X_2)$ where now the derivative depends on the regressor X_2 in a nonlinear way (as the function f is highly nonlinear). Both results are for $n = 1000, 5000$. For panel (a) we see that the spline estimator remain well behaved across all designs (across rows), the single-hidden layer (1L) sigmoid ANN estimators remain adequate while both versions of the AGMM estimators exhibit some bias. In panel (b), spline remains well behaved and so are the 1L sigmoid ANN estimators. In Figure 7 we show estimates of the partial derivative evaluated at various fixed values for some regressors. Though the estimators do not track the function well, especially in the tails in the bottom display, the average derivative is estimated well. Interestingly, 1L sigmoid ANN seems to estimate the derivative function marginally better than splines, perhaps since ANNs are able to automatically generate rich interaction behavior, whereas specifying tensor products for spline sieves is somewhat onerous.

Finally, Tables 3 and 4 provides various inference statistics for the ANN SMD estimators P-ISMD and OP-OSMD for Monte Carlo 2, without assuming any semiparametric structure on $h(\cdot)$ beyond smoothness. In particular, we report bootstrapped confidence intervals for relu and sigmoid and for depths 1 and 3 when the dimension of the nuisance variables $\tilde{X}$ ranges from 0 to 10. The results are also given for sample sizes $n = 1000$ and $n = 5000$. Across all specifications, the two ANN estimators perform adequately.

Overall, it seems that ANN methods are useful in approximating potentially high dimensional functions in NPIV models. Also, in the class of models we investigated, choices of layers, widths or activation functions are not very consequential in terms of finite sample performance. On the other hand, ANN based estimators in these non-standard NPIV models are hard to tune with, and a researcher needs to choose many tuning parameters. These ANN estimators are also unstable in some runs as they are based on highly complex (and non-convex) optimization programs. In addition, ANNs are not as effective in estimating univariate functions. Finally, to our surprise, We find that various plug-in spline SMD estimators to be stable, less biased generally and can outperform ANNs for NPIV models even in high dimensional cases with 13 continuous regressors.

5 Two Empirical Illustrations: Averaged Price Derivatives of Consumer Demand Curves

5.1 National Household Travel Survey Demand

We use data on gasoline demand from the 2001 National Household Travel Survey (Blundell, Horowitz and Parey, 2012, 2017). The sample we use include 4,812 observations in the full sample provided by Chen and Christensen (2018). We estimate an NPIV analogue of the model (11) in Blundell, Horowitz and Parey (2012), where the outcome variable is log gasoline demand, the endogenous variable is log price, and the included covariates follow Column (3) in Table 2 of Blundell, Horowitz and Parey (2012). The instrument is the distance from the Gulf coast, following Chen and Christensen (2018). We define the estimand as the average price derivative of the unknown structural function, which has an average elasticity interpretation.

Table 1: Estimates of price elasticity for gasoline in National Household Travel Survey data (Blundell, Horowitz and Parey, 2012)

	P-ISMD	OP-OSMD	IS
Sigmoid [1L]	-1.28	-1.24	-1.12
	[-1.69, -0.9]	[-1.64, -0.87]	(0.22)
Sigmoid [3L]	-1.24	-1.28	
	[-1.65, -0.9]	[-1.64, -0.87]	
ReLU [3L]	-1.27	-1.25	
	[-1.65, -0.9]	[-1.64, -0.87]	
Spline(3, 2)	-1.17	-1.2	
	[-1.57, -0.8]	[-1.6,-0.8]	
	Blundell *et al.* (2012) OLS	OLS	TSLS
	-0.83	-0.85	-1.24
	(0.148)	(0.15)	(0.2)

Notes. The covariates included are log gasoline price, log income, household size, driver, household age, number working, public transit distance. We instrument gasoline price with distance to Gulf of Mexico. □

Table 1 shows our estimates for the average elasticity. Broadly speaking, these estimates point to a similar range of values and are similar to a parametric two-stage least-squares specification. Across estimator classes, the neural network-based estimates are slightly larger in magnitude than the spline estimates and the IS estimates. Within the neural network SMD estimator class, architecture choices of the networks do not appear to matter much for the result.

5.2 Strawberry demand

Table 2: Estimate of demand average derivatives from Nielsen strawberry demand data (Compiani, 2019)

Non-organic

	IS	P-ISMD	OP-OSMD
Sigm [1L]	-1.649	-1.530	-1.747
	(0.04)	(0.04)	(0.03)
		[-1.8, -1.7]	[-2.3, -1.8]
Relu [1L]	-1.648	-1.590	-1.706
	(0.04)	(0.04)	(0.04)
		[-1.9, -1.7]	[-2.3, -1.8]
Relu [3L]	-1.648	-1.634	-1.659
	(0.04)	(0.04)	(0.06)
		[-1.9, -1.55]	[-2.2, -1.5]
Spline(3,2)	-1.611	-1.648	-1.676
	(0.04)	(0.04)	(0.04)

Organic

	IS	P-ISMD	OP-OSMD
Sigm [1L]	-3.235	-2.409	-3.382
	(0.07)	(0.09)	(0.06)
		[-2.7, -2.44]	[-4.3, -3.5]
Relu [1L]	-3.236	-2.197	-2.129
	(0.07)	(0.06)	(0.08)
		[-2.4, -2.11]	[-2.4, -2.06]
Relu [3L]	-3.232	-2.206	-2.122
	(0.07)	(0.07)	(0.14)
		[-3.1, -2.08]	[-2.36, -2.06]
Spline(3,2)	-3.194	-3.232	-3.124
	(0.06)	(0.07)	(0.06)

Notes. We include the following 6 covariates (Y_2): strawberry prices (non-organic, organic), income, lettuce demand (taste for organic proxy), state-level sale of non-strawberry fresh fruits, average outside good price. The excluded instruments are 3 Hausman IV (prices in neighbouring markets)+ 2 strawberry spot prices (marginal cost measures). □

We also consider a setting where consumers choose two substitutable goods. We use the Nielsen dataset from Compiani (2019),[16] where consumers in California choose from strawberries, organic strawberries, and an outside option.[17] We observe the market share of each type of product, their prices, and a variety of covariates at the market (store-week) level. In the analysis, we consider NPIV model $\mathbb{E}[Y_1 - h_0(Y_2) \mid X] = 0$ where Y_1 is the log market share of a type of good (non-organic or organic strawberries) and Y_2 is a vector of 6 random variables, including endogenous prices for both types of strawberries and the outside good, and other market-level covariates. The instruments X include Hausman instruments as well as cost shifters such as measurements of consumer taste and income at the market level. We focus on the target parameter $\theta_0 = \mathbb{E}[\nabla_1 h_0]$, which is the average derivative of h with respect to the own-price in logs, which we interpret as a version of price elasticity.[18]

We present the results in Table 2. As is perhaps expected from a casual intuition, estimates

[16] Our results do not necessarily represent the views of the Nielsen Company

[17] For a detailed description of the data, see Appendix G of https://www.tse-fr.eu/sites/default/files/TSE/documents/sem2019/eee/compiani.pdf.

[18] Under a model of the demand where the NPIV condition $\mathbb{E}[Y_1 - h_0(Y_2) \mid X] = 0$ defines the demand function h_0, we can understand θ_0 as a price elasticity. However, this model—which implicitly assumes that endogeneity is additive—may not be consistent with microfoundations of consumer behavior (Berry and Haile, 2016), and so care should be taken in interpreting θ_0 as an elasticity. Nevertheless, for purposes of our illustration here, we may continue to view θ_0 as some well-defined function of the distribution of the data. For a more detailed implementation of demand in this setting, see Compiani (2019).

of θ_0 are negative across both products, and more negative for the more price-sensitive product (organic strawberry). Moreover, results are broadly similar across estimation methods (SMD vs. score) and sieve choices (spline vs. neural net), with perhaps more variability for neural networks in organic strawberries. The estimates for non-organic strawberries hover around -1.5, and are reasonably stable across choices of tuning parameters and estimators (IS vs. SMD estimators). The estimates for organic strawberries are more variable across specification of nuisance parameters and neural architectures, but seem to be around -2 and -3, and larger in magnitude than the own-price elasticity estimate for non-organic strawberries.

These estimates are qualitatively similar to Compiani (2019)'s estimates, which reports median own-price elasticities of -1.4 (0.03) for non-organic strawberries and -5.5 (0.7) for organic strawberries.[19] Our estimates are more dissimilar for organic strawberries, for which we offer a few conjectures. First, Compiani (2018) reports estimates following Berry and Haile (2016)'s approach to demand estimation, that accounts for price endogeneity differently. Under his assumptions, it is possible that our estimator is consistent for a different parameter than his. Second, organic strawberry market shares are very small, and hence fluctuates more on a log scale, thereby resulting in worse estimation precision.

6 Conclusion

In this paper, we present two semiparametric efficient estimators for weighted average derivatives of nonparametric instrumental variables regressions of moderate and high dimensional endogenous and exogenous regressors. We have conducted detailed comparisons of finite sample performance of the inefficient and efficient estimators using various ANN nonlinear sieves of regular expectation functionals of semi-nonparametric functions of endogenous and exogenous variables via fixed finite layer ANN sieves. The simulation studies and empirical applications confirm the theoretical advantage of ANN estimation of unknown functions of high dimensional variables, after some tuning of hyperparameters. But surprisingly, simple spline estimators perform very well in terms of finite sample biases and variances. There seems gaps between approximation theory and finite sample computational performance in applying flexible ANN nonlinear sieves to estimate nonparametric models with endogeneity.[20]

[19] Interestingly, our estimates are closer to estimates from BLP that Compiani (2019) reports in Figure 4, which are also around -2 to -3.

[20] In a concurrent work, Chen, Liao and Wang (2021) consider ANN sieve quasi likelihood ratio inference on possibly irregular functionals of semi-nonparametric conditional moment restrictions for time series data, including nonparametric quantile instrumental variables regressions with high dimensional regressors as a leading example.

References

Ai, C. and Chen, X. (2003). Efficient estimation of models with conditional moment restrictions containing unknown functions. *Econometrica*, **71** (6), 1795–1843.

— and — (2007). Estimation of possibly misspecified semiparametric conditional moment restriction models with different conditioning variables. *Journal of Econometrics*, **141** (1), 5–43.

— and — (2012). The semiparametric efficiency bound for models of sequential moment restrictions containing unknown functions. *Journal of Econometrics*, **170** (2), 442–457.

Athey, S., Imbens, G. W., Metzger, J. and Munro, E. M. (2019). *Using Wasserstein Generative Adversarial Networks for the Design of Monte Carlo Simulations*. Tech. rep., National Bureau of Economic Research.

Berry, S. and Haile, P. (2016). Identification in differentiated products markets. *Annual Review of Economics*, **8** (1), 27–52.

Bickel, P. J., Klaassen, C. A., Bickel, P. J., Ritov, Y., Klaassen, J., Wellner, J. A. and Ritov, Y. (1993). *Efficient and adaptive estimation for semiparametric models*, vol. 4. Johns Hopkins University Press Baltimore.

Blundell, R., Horowitz, J. and Parey, M. (2017). Nonparametric estimation of a nonseparable demand function under the slutsky inequality restriction. *Review of Economics and Statistics*, **99** (2), 291–304.

—, Horowitz, J. L. and Parey, M. (2012). Measuring the price responsiveness of gasoline demand: Economic shape restrictions and nonparametric demand estimation. *Quantitative Economics*, **3** (1), 29–51.

Brown, B. W. and Newey, W. K. (2002). Generalized method of moments, efficient bootstrapping, and improved inference. *Journal of Business & Economic Statistics*, **20** (4), 507–517.

Chamberlain, G. (1992a). Comment: sequential moment restrictions in panel data. *Journal of Business and Economic Statistics*, **10**, 20–26.

— (1992b). Comment: Sequential moment restrictions in panel data. *Journal of Business & Economic Statistics*, **10** (1), 20–26.

Chen, X. (2007). Large sample sieve estimation of semi-nonparametric models. *Handbook of econometrics*, **6**, 5549–5632.

—, Christensen, T. and Kankanala, S. (2021). Adaptive estimation and uniform confidence bands for nonparametric iv. *arXiv preprint arXiv:2107.11869*.

— and Christensen, T. M. (2018). Optimal sup-norm rates and uniform inference on nonlinear functionals of nonparametric iv regression. *Quantitative Economics*, **9** (1), 39–84.

—, Hong, H. and Shum, M. (2007). Nonparametric likelihood ratio model selection tests between parametric likelihood and moment condition models. *Journal of Econometrics*, **141** (1), 109–140.

— and Liao, Z. (2015). Sieve semiparametric two-step gmm under weak dependence. *Journal of Econometrics*, **189**, 163–186.

— and Pouzo, D. (2015). Sieve wald and qlr inferences on semi/nonparametric conditional moment models. *Econometrica*, **83** (3), 1013–1079.

—, — and Powell, J. L. (2019). Penalized sieve gel for weighted average derivatives of nonparametric quantile iv regressions. *Journal of Econometrics*, **213** (1), 30–53.

— and Qiu, Y. J. J. (2016). Methods for nonparametric and semiparametric regressions with endogeneity: A gentle guide. *Annual review of economics*, **8**, 259–290.

— and White, H. (1999). Improved rates and asymptotic normality for nonparametric neural network estimators. *IEEE Transactions on Information Theory*, **45** (2), 682–691.

Chernozhukov, V., Chetverikov, D., Demirer, M., Duflo, E., Hansen, C., Newey, W. and Robins, J. (2018). Double/debiased machine learning for treatment and structural parameters.

—, Escanciano, J. C., Ichimura, H., Newey, W. K. and Robins, J. M. (2021). Locally robust semiparametric estimation. *arXiv preprint arXiv:1608.00033*.

Compiani, G. (2018). Nonparametric demand estimation in differentiated products markets. *Available at SSRN 3134152*.

— (2019). Market counterfactuals and the specification of multi-product demand: A nonparametric approach. *Available at SSRN*.

Dikkala, N., Lewis, G., Mackey, L. and Syrgkanis, V. (2020). Minimax estimation of conditional moment models. *arXiv preprint arXiv:2006.07201*.

Donald, S. G., Imbens, G. W. and Newey, W. K. (2003). Empirical likelihood estimation and consistent tests with conditional moment restrictions. *Journal of Econometrics*, **117** (1), 55–93.

Farrell, M. H., Liang, T. and Misra, S. (2018). Deep neural networks for estimation and inference. *arXiv preprint arXiv:1809.09953*.

Hartford, J., Lewis, G., Leyton-Brown, K. and Taddy, M. (2017). Deep iv: A flexible approach for counterfactual prediction. In *International Conference on Machine Learning*, pp. 1414–1423.

Hornik, K., Stinchcombe, M. and White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural networks*, **2** (5), 359–366.

Kingma, D. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Newey, W. K. and Powell, J. L. (2003). Instrumental variable estimation of nonparametric models. *Econometrica*, **71** (5), 1565–1578.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L. and Lerer, A. (2017). Automatic differentiation in pytorch.

Santos, A. (2012). Inference in nonparametric instrumental variables with partial identification. *Econometrica*, **80** (1), 213–275.

Schmidt-Hieber, J. (2019). Deep relu network approximation of functions on a manifold. *arXiv preprint arXiv:1908.00695*.

Severini, T. and Tripathi, G. (2013). Semiparametric efficiency bounds for microeconometric models: A survey. *Foundations and Trends® in Econometrics*, **6** (3–4), 163–397.

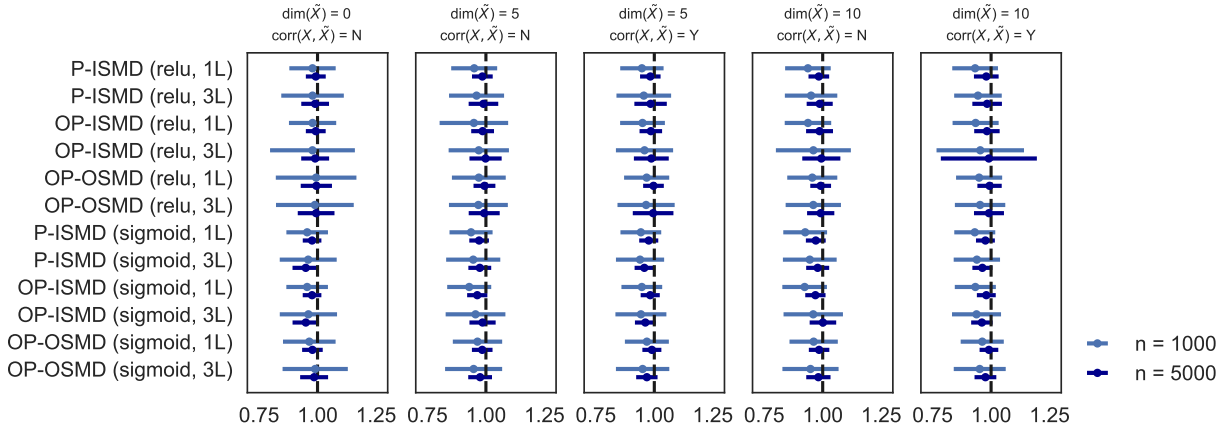
Stinchcombe, M. B. and White, H. (1998). Consistent specification testing with nuisance parameters present only under the alternative. *Econometric theory*, pp. 295–325.

Van der Vaart, A. W. (2000). *Asymptotic statistics*, vol. 3. Cambridge university press.

Wooldridge, J. M. and White, H. (1988). Some invariance principles and central limit theorems for dependent heterogeneous processes. *Econometric theory*, pp. 210–230.

Yarotsky, D. (2017). Error bounds for approximations with deep relu networks. *Neural Networks*, **94**, 103–114.

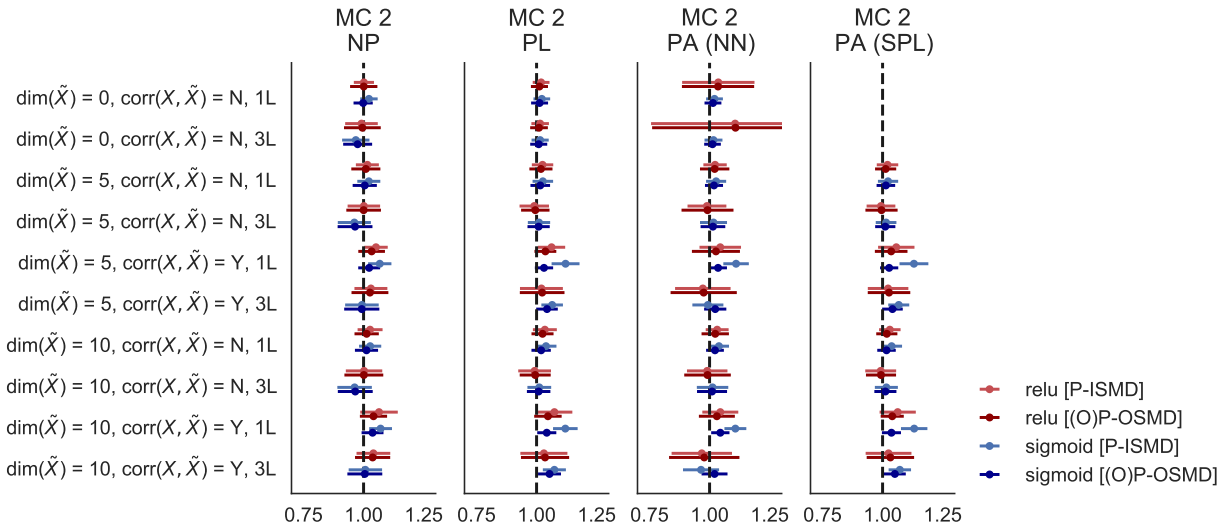
Figure 1: ANN SMD estimators for the average derivative parameter in Monte Carlo 1



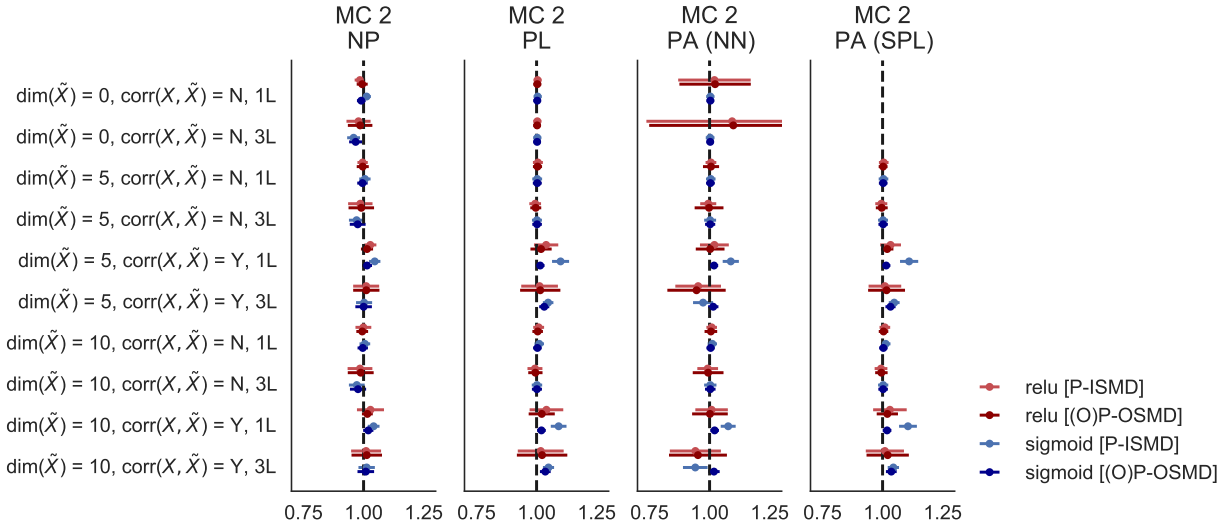
Notes. Monte Carlo Mean ±1 Monte Carlo standard deviation across 1,000 replications. □

Figure 2: ANN SMD estimators for Monte Carlo 2

(a) $n = 1000$



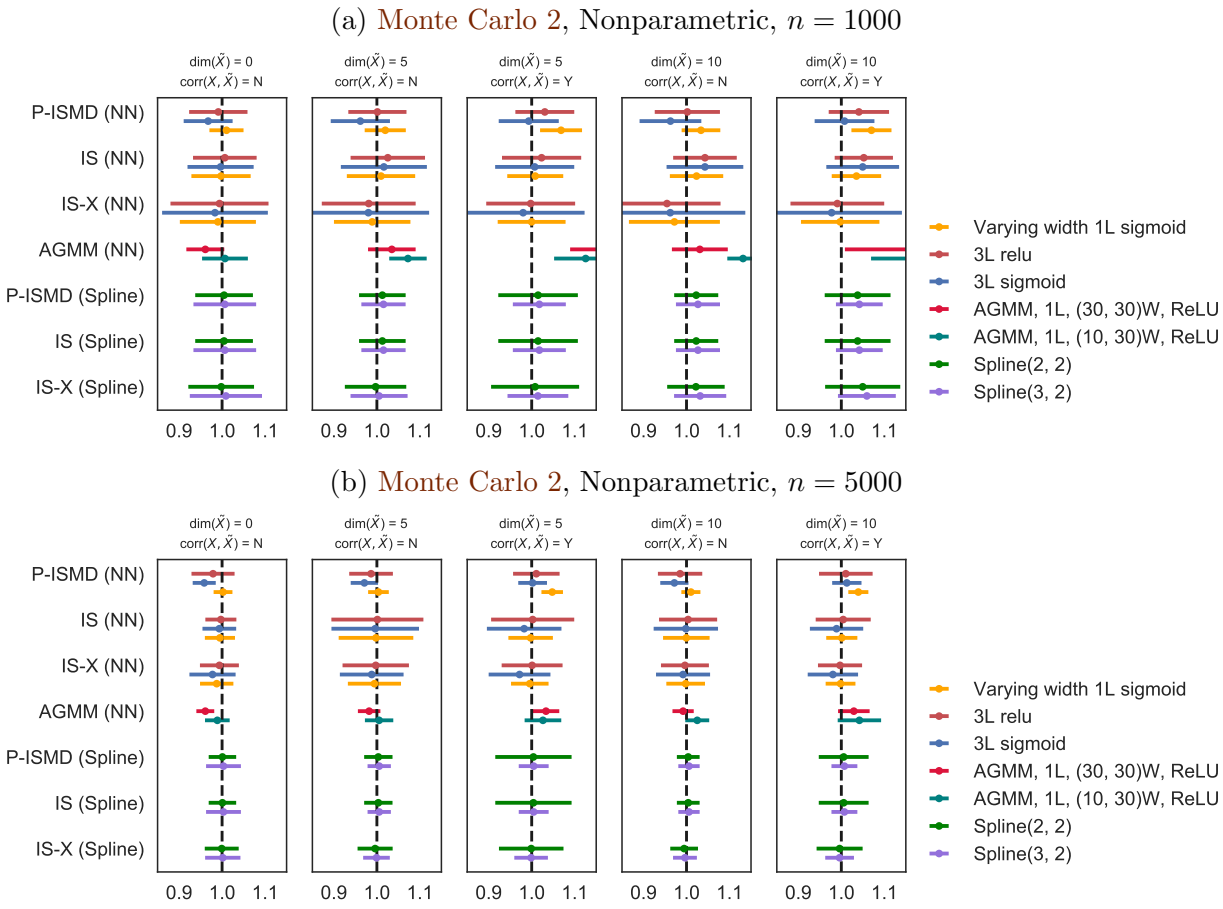
(b) $n = 5000$



Notes. Monte Carlo Mean ±1 Monte Carlo standard deviation across 1,000 replications.
The columns are estimators where different correct assumptions of the data-generating process are placed. The first column (NP: *nonparametric*) shows estimated average derivative of an NPIV model, where the unknown function $h(Y_2)$ is not assumed to have separable structure. The second column (PL: *partially linear*) assumes $h(Y_2) = \theta R_1 + h_1(R_2, X_2, \tilde{X})$. The third and fourth columns (PA: *partially additive*) assumes $h(Y_2) = \theta R_1 + h_1(R_2) + h_2(X_2) + h_3(\tilde{X})$. The third column uses neural networks to approximate the scalar functions h_1, h_2, and the fourth column uses splines to approximate h_1, h_2 (while h_3 is always estimated via ANN).

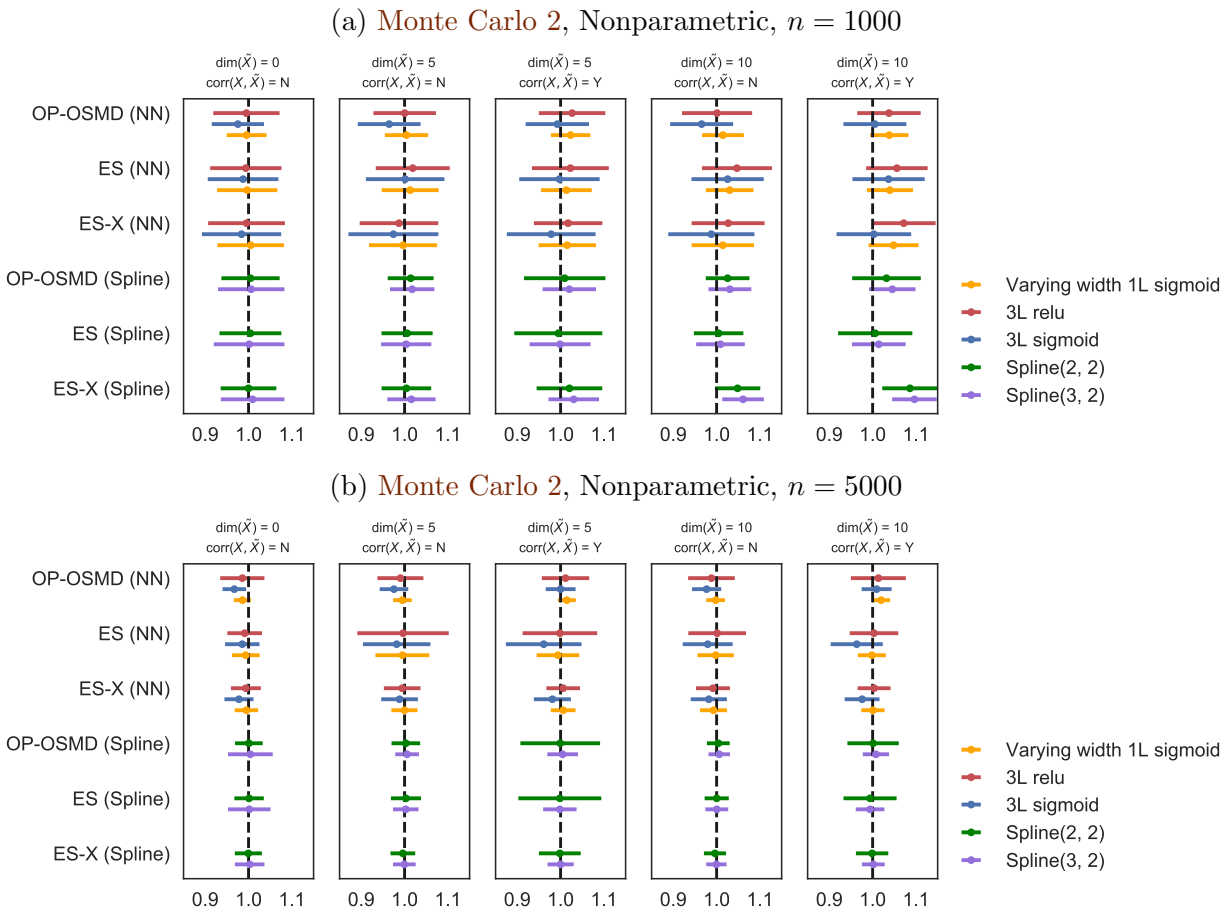
For each type of assumption placed on the true $h_0(Y_2)$, we vary the data-generating process by varying the dimension of $\tilde{X}$ and the level of correlation between (X_1, X_2, X_3) and $\tilde{X}$. We also vary the network architecture by $\{\text{ReLU}, \text{Sigmoid}\} \times \{\text{1L}, \text{3L}\} \times \{\text{10W}\}$. Lastly, we vary the type of estimator used from simple plug-in with the identity-weighted SMD estimator to orthogonalized plug-in with the optimally-weighted SMD estimator. □

Figure 3: Estimation quality of average derivative parameter in Monte Carlo 2 across a variety of *identity-weighted* estimators



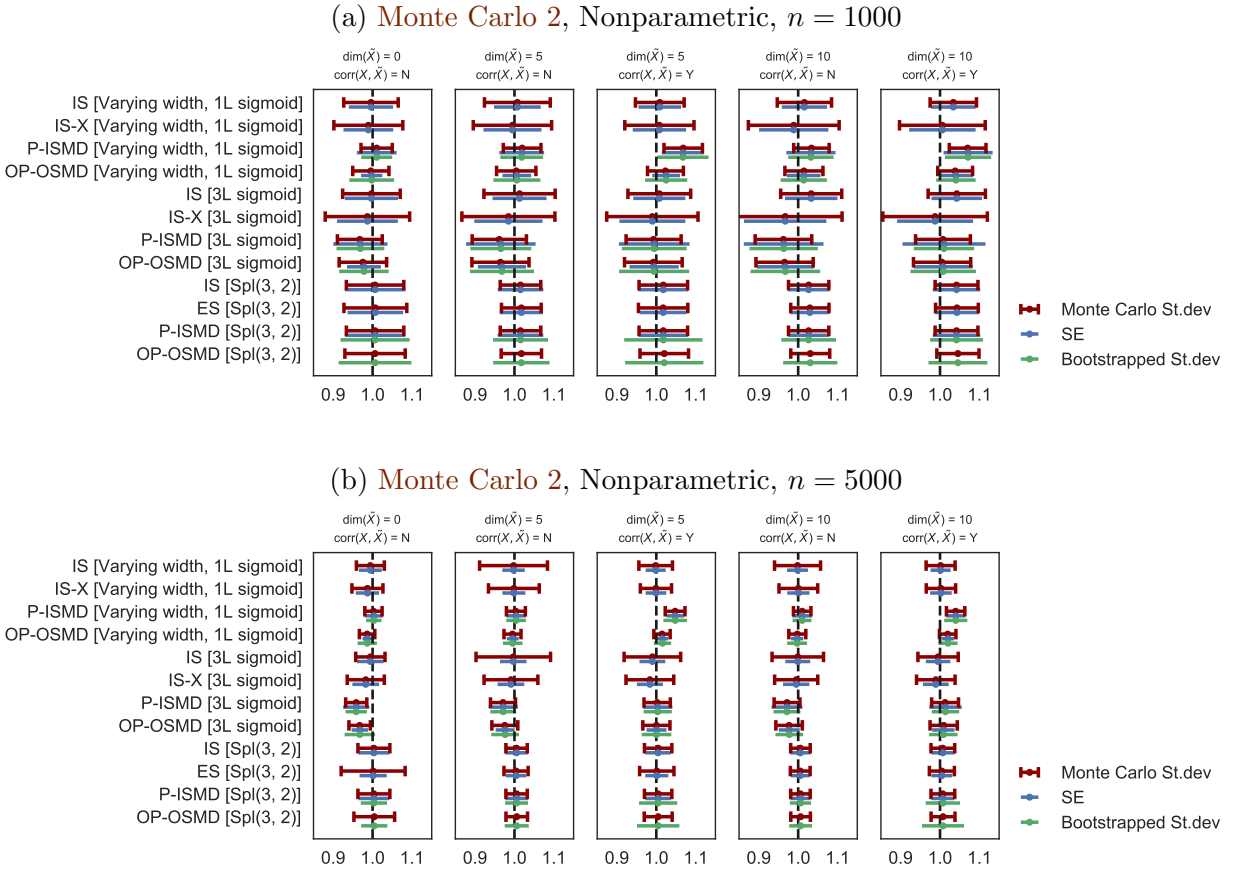
Notes. Monte Carlo Mean ±1 Monte Carlo standard deviation across 1,000 replications.
We consider a few different estimation strategies and vary over choice of tuning parameters for nuisance parameters in these estimation strategies.
In terms of estimation strategies, IS stands for identity score estimators, detailed in Item 3, whereas IS-X stands for the score estimators, but with two-fold cross-fitting. AGMM uses the adversarial GMM estimation algorithm in Dikkala *et al.* (2020) to compute $\hat{h}$, and outputs the simple plug-in estimator for θ. P-ISMD estimators follow Item 1.
In terms of neural architecture and spline parameter choices, *varying width 1L sigmoid* refers to using 1-layer sigmoid network, but vary the width of the network according to $\dim(\tilde{X})$, as opposed to fixing the width at 10. The two AGMM architecture choices refer to different widths for the network estimating h and the adversarial network approximating the instrument test functions, where (10,30)W refers to using width-10 for h and width-30 for the instruments. Lastly, Spline(a, b) is a spline basis for approximating h such that each spline function is an $(a-1)$-degree piecewise polynomial that have b knots, where we include pairwise interactions in lieu of tensor products. In the spline scenarios, Spline$(a+1, b)$ is used as a spline basis for the instruments. Tuning parameter choices for estimation of additional nuisance parameters are detailed in Table 5. □

Figure 4: Estimation quality of average derivative parameter in Monte Carlo 2 across a variety of *optimally weighted* estimators



Notes. Monte Carlo Mean ± 1 Monte Carlo standard deviation across 1,000 replications.
We consider a few different estimation strategies and vary over choice of tuning parameters for nuisance parameters in these estimation strategies.
In terms of estimation strategies, ES stands for efficient score estimators, detailed in Item 3, whereas ES-X stands for the score estimators, but with two-fold cross-fitting. OP-OSMD estimators follow Item 1.
In terms of neural architecture and spline parameter choices, *varying width 1L sigmoid* refers to using 1-layer sigmoid network, but vary the width of the network according to $\dim(\tilde{X})$, as opposed to fixing the width at 10. Lastly, Spline(a, b) is a spline basis for approximating h such that each spline function is an $(a-1)$-degree piecewise polynomial that have b knots, where we include pairwise interactions in lieu of tensor products. In the spline scenarios, Spline$(a+1, b)$ is used as a spline basis for the instruments. Tuning parameter choices for estimation of additional nuisance parameters are detailed in Table 5. □

Figure 5: Inference quality of average derivatrive parameter in Monte Carlo 2 across a variety of estimators



Notes. Monte Carlo Mean ±1{Monte Carlo st. dev., estimated s.e., bootstrapped s.e.} across 1,000 replications. Bootstrap SEs are based on one realization of the data. □

Figure 6: Estimation quality of average derivative parameter in Monte Carlo 3 across a variety of estimators

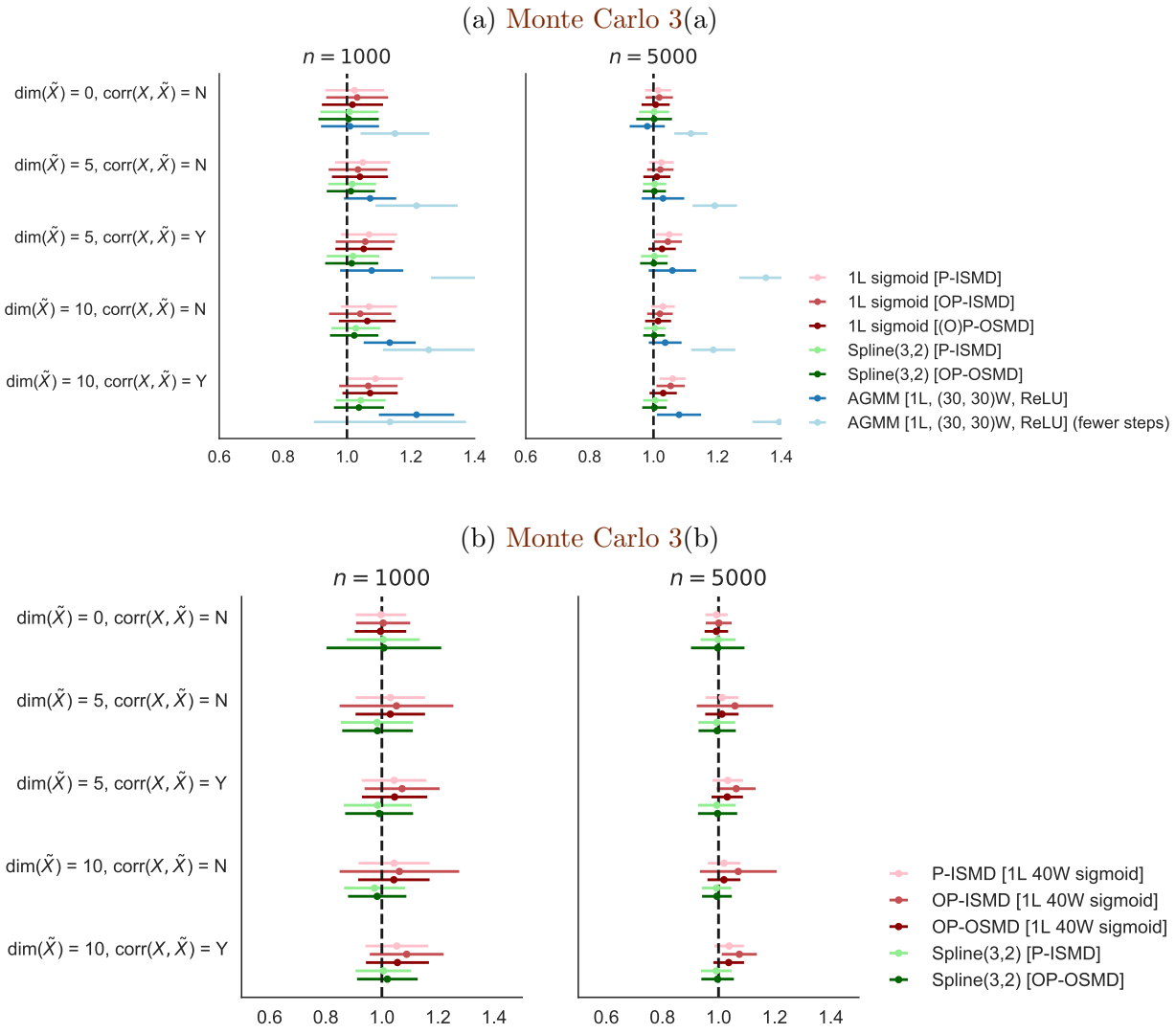
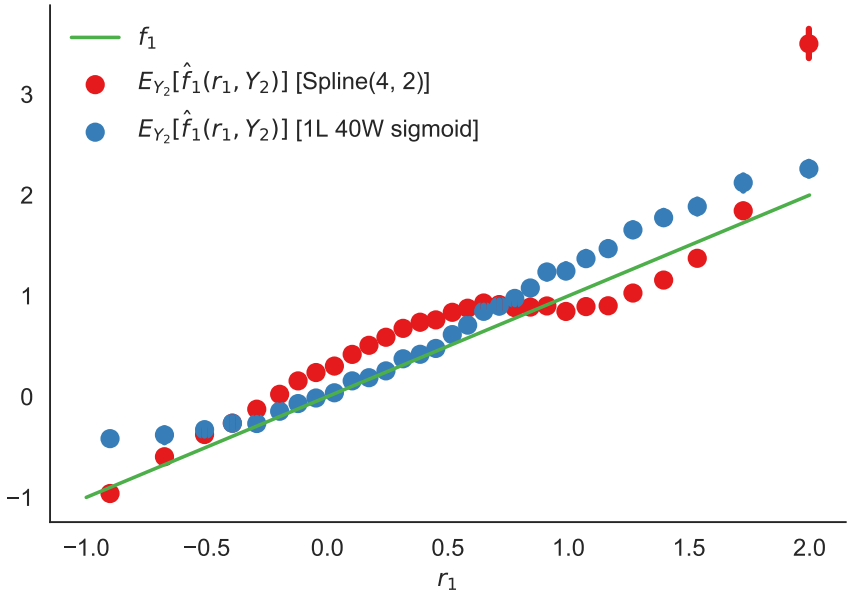
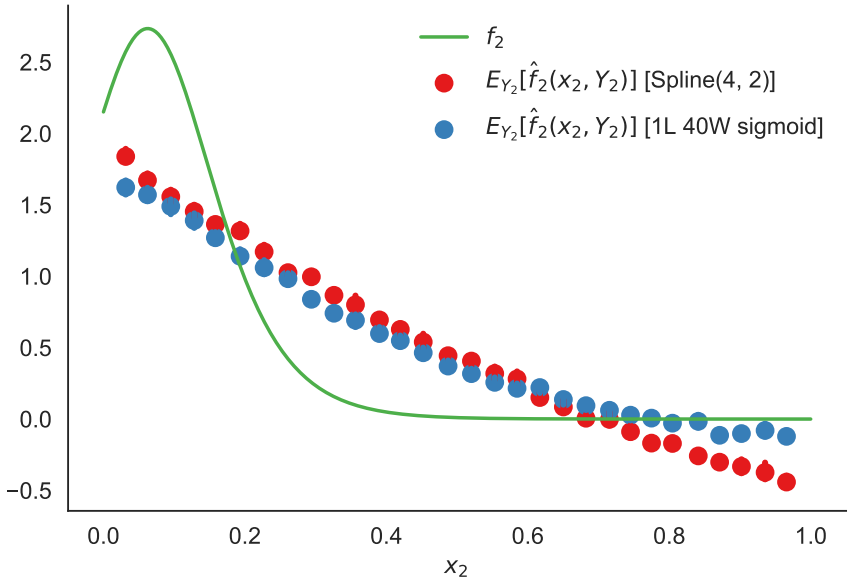


Figure 7: Estimation quality of the partial derivative function in Monte Carlo 3(b) across a variety of estimators

(a) Estimated f_1 versus true f_1. Single sample for $N = 10,000$



(b) Estimated f_2 versus true f_2. Single sample for $N = 10,000$



Notes. In the DGP Monte Carlo 3(b), the partial derivative $\nabla_1 h_0$ is of the form $f_1(R_2) + f_2(X_2)$, and we evaluate performance estimating f_1, f_2. Estimated f_1 is calculated by taking $\nabla_1 \hat{h} - f_2(x_2)$. We plot expectation marginalizing over variables other than r_1. Estimated f_2 is calculated by taking $\nabla_1 \hat{h} - f_1(r_1)$. We plot expectation marginalizing over variables other than x_2. □

Table 3: SMD Inference Results for P-ISMD and OP-OSMD, $N = 1000$

				P-ISMD					OP-OSMD				
Nui. Dim	$\text{Corr}(X, \tilde{X})$	Depth	Activation	Mean	Std	Med. Est. SE	Boot. LB	Boot. UB	Mean	Std	Med. Est. SE	Boot. LB	Boot. UB
0	0.0000	1	relu	1.001	0.042	0.045	0.948	1.077	1.001	0.057	0.029	0.909	1.102
			sigmoid	1.022	0.036	0.046	0.978	1.118	0.998	0.040	0.026	0.913	1.108
		3	relu	0.991	0.068	0.053	0.913	1.087	0.995	0.076	0.033	0.840	1.235
			sigmoid	0.968	0.057	0.069	0.871	1.107	0.975	0.060	0.043	0.883	1.126
5	0.0000	1	relu	1.016	0.048	0.054	0.889	1.200	1.009	0.061	0.040	0.870	1.218
			sigmoid	1.022	0.048	0.057	0.873	1.103	1.005	0.050	0.037	0.893	1.133
		3	relu	1.001	0.068	0.061	0.878	1.085	1.000	0.072	0.043	0.875	1.096
			sigmoid	0.962	0.069	0.088	0.713	1.024	0.964	0.072	0.061	0.722	1.048
	0.5000	1	relu	1.052	0.049	0.052	0.946	1.159	1.033	0.055	0.038	0.906	1.142
			sigmoid	1.067	0.048	0.053	0.921	1.163	1.023	0.046	0.035	0.904	1.116
		3	relu	1.030	0.068	0.060	0.895	1.143	1.026	0.077	0.043	0.905	1.167
			sigmoid	0.993	0.070	0.090	0.740	1.054	0.992	0.073	0.062	0.722	1.072
10	0.0000	1	relu	1.027	0.052	0.061	0.915	1.090	1.013	0.051	0.045	0.908	1.104
			sigmoid	1.027	0.046	0.062	0.900	1.111	1.012	0.048	0.042	0.940	1.157
		3	relu	1.002	0.076	0.070	0.749	1.179	1.001	0.081	0.052	0.753	1.184
			sigmoid	0.963	0.072	0.101	0.670	1.015	0.965	0.073	0.072	0.682	1.026
	0.5000	1	relu	1.064	0.078	0.064	0.968	1.147	1.041	0.057	0.045	0.959	1.148
			sigmoid	1.071	0.047	0.064	0.931	1.133	1.037	0.046	0.044	0.935	1.135
		3	relu	1.041	0.070	0.073	0.865	1.251	1.038	0.073	0.052	0.867	1.278
			sigmoid	1.007	0.070	0.105	0.749	1.041	1.005	0.073	0.075	0.740	1.056

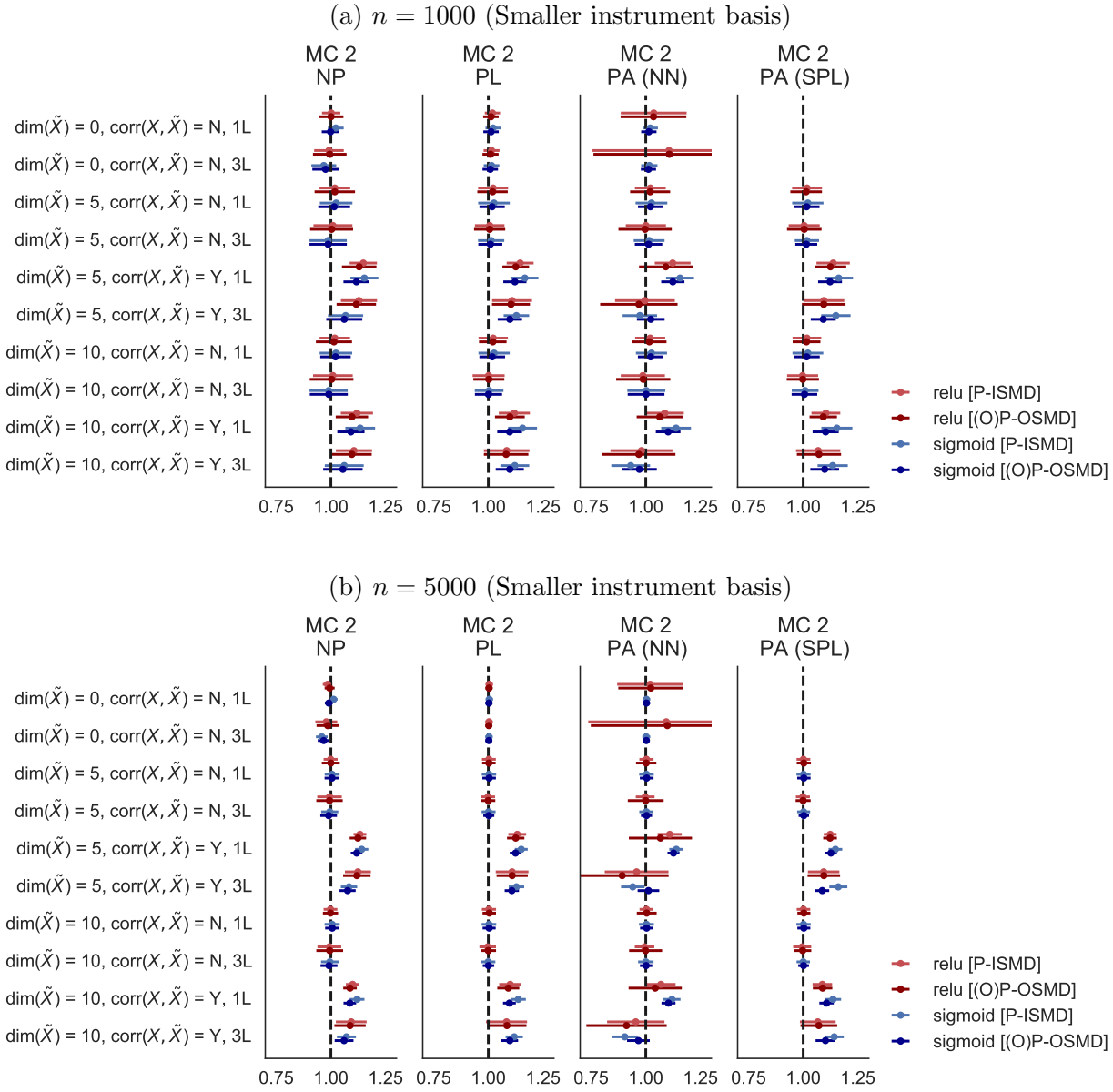
Notes. 1000 Monte Carlo replications. Bootstrap CIs based on a single replication. □

Table 4: SMD Inference Results for P-ISMD and OP-OSMD, $N = 5000$

				P-ISMD					OP-OSMD				
Nui. Dim	$\text{Corr}(X, \tilde{X})$	Depth	Activation	Mean	Std	Med. Est. SE	Boot. LB	Boot. UB	Mean	Std	Med. Est. SE	Boot. LB	Boot. UB
0	0.0000	1	relu	0.985	0.022	0.021	0.966	1.036	0.994	0.023	0.013	0.897	1.031
			sigmoid	1.013	0.018	0.021	0.976	1.054	0.990	0.018	0.011	0.942	1.026
		3	relu	0.979	0.050	0.023	0.933	1.012	0.986	0.051	0.014	0.850	1.056
			sigmoid	0.958	0.027	0.033	0.876	0.978	0.967	0.027	0.021	0.881	1.035
5	0.0000	1	relu	0.996	0.022	0.024	0.968	1.079	0.997	0.025	0.017	0.912	1.076
			sigmoid	1.006	0.024	0.025	0.964	1.065	0.995	0.022	0.015	0.940	1.034
		3	relu	0.986	0.051	0.026	0.922	1.041	0.990	0.053	0.018	0.914	1.043
			sigmoid	0.971	0.032	0.037	0.938	1.071	0.976	0.033	0.024	0.933	1.078
	0.5000	1	relu	1.028	0.025	0.023	0.976	1.071	1.014	0.025	0.016	0.907	1.052
			sigmoid	1.046	0.024	0.022	1.002	1.117	1.014	0.021	0.014	0.960	1.049
		3	relu	1.011	0.054	0.025	0.942	1.045	1.011	0.055	0.018	0.923	1.038
			sigmoid	1.002	0.033	0.037	0.927	1.066	1.000	0.034	0.025	0.908	1.060
10	0.0000	1	relu	0.999	0.033	0.025	0.959	1.044	0.994	0.025	0.018	0.933	1.031
			sigmoid	1.005	0.022	0.025	0.969	1.064	0.997	0.022	0.016	0.936	1.032
		3	relu	0.985	0.052	0.027	0.848	1.050	0.988	0.054	0.019	0.848	1.055
			sigmoid	0.972	0.033	0.039	0.930	1.064	0.977	0.034	0.027	0.932	1.068
	0.5000	1	relu	1.029	0.056	0.025	0.981	1.065	1.016	0.023	0.018	0.956	1.047
			sigmoid	1.042	0.025	0.024	1.008	1.113	1.020	0.021	0.016	0.973	1.065
		3	relu	1.010	0.062	0.028	0.911	1.126	1.013	0.063	0.020	0.901	1.122
			sigmoid	1.013	0.034	0.041	0.937	1.073	1.009	0.035	0.028	0.927	1.065

Notes. 1000 Monte Carlo replications. Bootstrap CIs based on a single replication. □

Figure 8: ANN SMD estimators for Monte Carlo 2 with smaller instrument basis



Notes. Monte Carlo Mean ±1 Monte Carlo standard deviation across 1,000 replications.
The columns are estimators where different correct assumptions of the data-generating process are placed. The first column (NP: *nonparametric*) shows estimated average derivative of an NPIV model, where the unknown function $h(Y_2)$ is not assumed to have separable structure. The second column (PL: *partially linear*) assumes $h(Y_2) = \theta R_1 + h_1(R_2, X_2, \tilde{X})$. The third and fourth columns (PA: *partially additive*) assumes $h(Y_2) = \theta R_1 + h_1(R_2) + h_2(X_2) + h_3(\tilde{X})$. The third column uses neural networks to approximate the scalar functions h_1, h_2, and the fourth column uses splines to approximate h_1, h_2.

For each type of assumption placed on the true $h_0(Y_2)$, we vary the data-generating process by varying the dimension of $\tilde{X}$ and the level of correlation between (X_1, X_2, X_3) and $\tilde{X}$. We also vary the network architecture by $\{\text{ReLU}, \text{Sigmoid}\} \times \{\text{1L}, \text{3L}\}$. Lastly, we vary the type of estimator used from simple plug-in with the identity-weighted SMD estimator to orthogonalized plug-in with the optimally-weighted SMD estimator. □

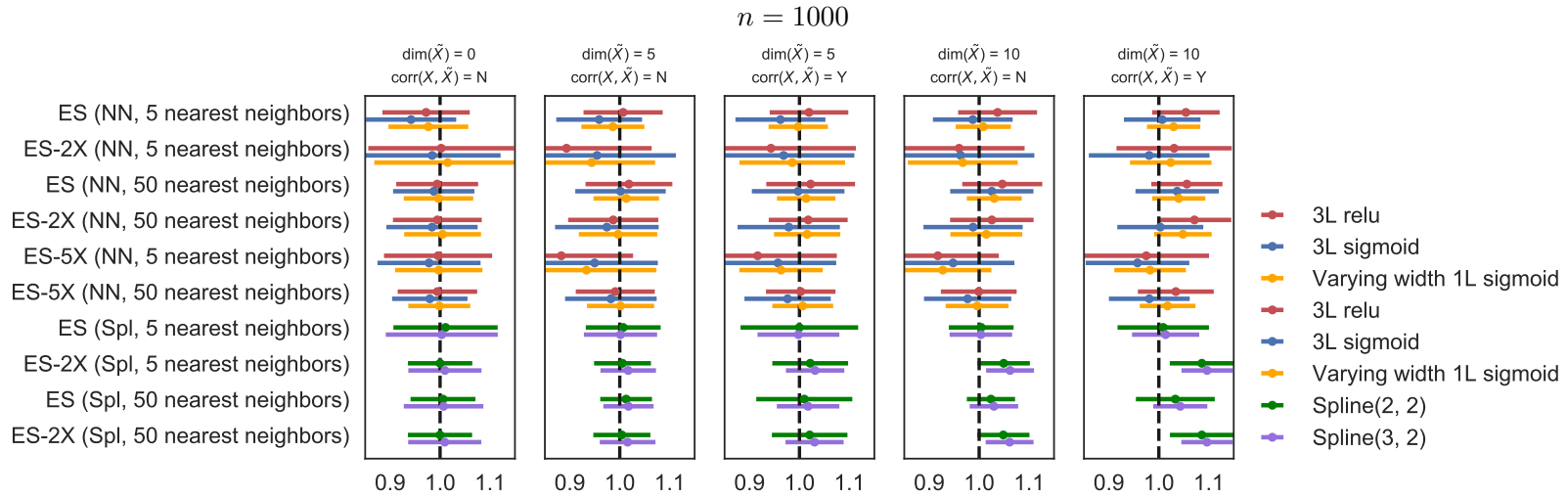


Figure 9: Performance of ES with different estimators for $\Sigma(X)^{-1}$ in the score expression

Notes. Monte Carlo Mean ± 1 Monte Carlo standard deviation across 1,000 replications.
"k-nearest neighbors": Use k-nearest neighbors to estimate $\Sigma(X)$ in the score (in $\mathbb{E}[v^\star \mid X]\Sigma(X)^{-1}$).
"True inverse variance": Plug in the true $\Sigma(X)$ for that in the score.
"Plug in identity": Plug in the identity matrix for $\Sigma(X)$ in the score.
"Projection": Use the projection of the squared residuals onto spline bases for $\Sigma(X)$.
"Estimate $w^\star$": Instead of estimating $v^\star$ with sieves, we estimate $w^\star$ with sieves and form $v^\star$ via plugging in estimates of other nuisance parameters. □

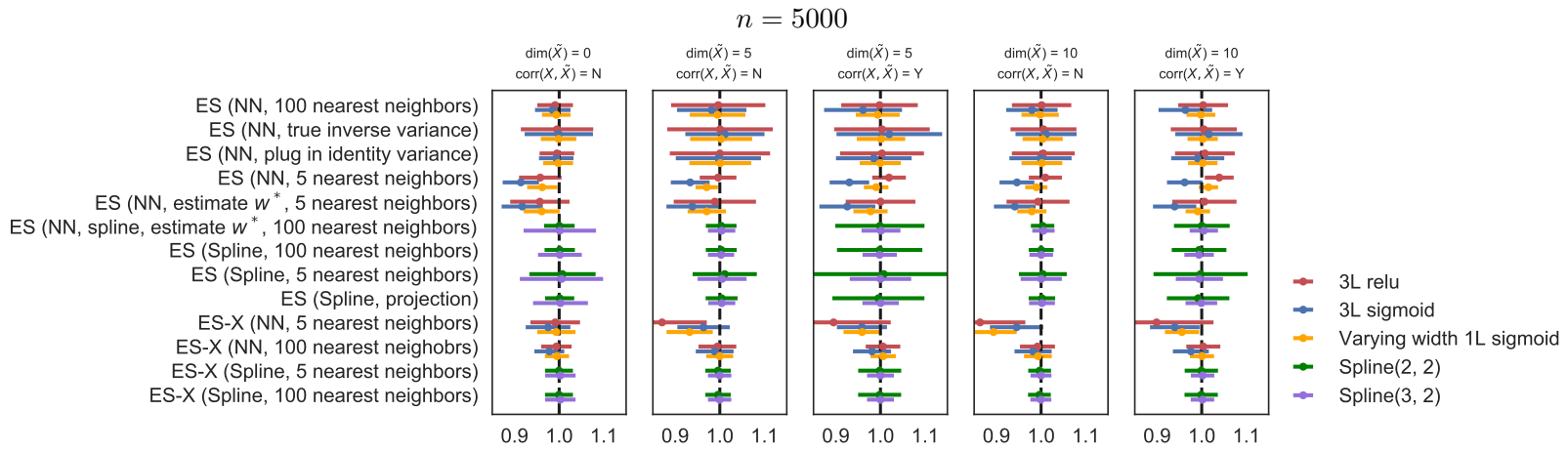


Figure 10: Performance of ES with different estimators for $\Sigma(X)^{-1}$ in the score expression

Notes. Monte Carlo Mean ± 1 Monte Carlo standard deviation across 1,000 replications.
"k-nearest neighbors": Use k-nearest neighbors to estimate $\Sigma(X)$ in the score (in $\mathbb{E}[v^\star \mid X]\Sigma(X)^{-1}$).
"True inverse variance": Plug in the true $\Sigma(X)$ for that in the score.
"Plug in identity": Plug in the identity matrix for $\Sigma(X)$ in the score.
"Projection": Use the projection of the squared residuals onto spline bases for $\Sigma(X)$.
"Estimate $w^\star$": Instead of estimating $v^\star$ with sieves, we estimate $w^\star$ with sieves and form $v^\star$ via plugging in estimates of other nuisance parameters. □

Estimator type	$\Sigma(X)$ [SMD]	$\Gamma(X)$	$\Sigma(X)$ [Score]	$v^\star$
P-ISMD [NN] OP-OSMD [NN]	5 nearest neighbors	Projection of demeaned $(\nabla\hat{h}-\overline{\nabla\hat{h}})((y-\hat{h})-\overline{(y-\hat{h})})$ on instrument basis used in SMD estimation. Then multiply $\Sigma(X)^{-1}$ estimate for SMD. Project the result onto the sieve basis for the instruments.		
IS [NN]				Sieve calculation of (18) with Spline(3, 2)
ES [NN]	Same as OP-OSMD [NN]	Same as OP-OSMD [NN]	50 nearest neighbors for $n = 1000$, 100 for $n = 5000$	Sieve calculation of (21) with Spline(3, 2)
IS/ES-X [NN]	Both scores take the form of $\nabla\hat{h} - \lambda(x)'\hat{\xi}\cdot(y-\hat{h})$ where $\lambda(x)$ is a sieve basis (Spline(3, 2)). The sample is split so that $\hat{h}$ and $\hat{\xi}$ are estimated from one half and the score is computed on the other. The roles of the two subsamples are then exchanged.			
P-ISMD [Spl] OP-OSMD [Spl]	Projection onto sieve basis for the instruments	Projection of demeaned $(\nabla\hat{h}-\overline{\nabla\hat{h}})((y-\hat{h})-\overline{(y-\hat{h})})$ on instrument basis used in SMD estimation. Then multiply $\Sigma(X)^{-1}$ estimate from SMD. Project the result onto the sieve basis for the instruments.		
IS [Spl]				Sieve calculation of (18) with same spline basis as the instruments
ES [Spl]	Same as OP-OSMD [Spl]	Same as OP-OSMD [Spl]	50 nearest neighbors for $n = 1000$, 100 for $n = 5000$	Sieve calculation of (21) with same spline basis as the instruments
IS/ES-X [Spl]	Both scores take the form of $\nu(y_2)'\hat{\beta} - \lambda(x)'\hat{\xi}\cdot(y-\hat{h})$ where $\lambda(x), \nu(y_2)$ are sieve bases. The sample is split so that $\hat{\beta}$ and $\hat{\xi}$ are estimated from one half and the score is computed on the other. The roles of the two subsamples are then exchanged.			

Table 5: Estimation of additional nuisance parameters