Yale University

EliScholar – A Digital Platform for Scholarly Publishing at Yale

Cowles Foundation Discussion Papers | Cowles Foundation

12-31-2021

Efficient Estimation of Average Derivatives in NPIV Models: Simulation Comparisons of Neural Network Estimators

Jiafeng Chen

Xiaohong Chen
Yale University

Elie Tamer

Follow this and additional works at: https://elischolar.library.yale.edu/cowles-discussion-paper-series

Part of the Economics Commons

Recommended Citation

Chen, Jiafeng; Chen, Xiaohong; and Tamer, Elie, "Efficient Estimation of Average Derivatives in NPIV Models: Simulation Comparisons of Neural Network Estimators" (2021). *Cowles Foundation Discussion Papers*. 2673.
https://elischolar.library.yale.edu/cowles-discussion-paper-series/2673

This Discussion Paper is brought to you for free and open access by the Cowles Foundation at EliScholar – A Digital Platform for Scholarly Publishing at Yale. It has been accepted for inclusion in Cowles Foundation Discussion Papers by an authorized administrator of EliScholar – A Digital Platform for Scholarly Publishing at Yale. For more information, please contact elischolar@yale.edu.

EFFICIENT ESTIMATION OF AVERAGE DERIVATIVES IN NPIV MODELS: SIMULATION COMPARISONS OF NEURAL NETWORK ESTIMATORS

By

Jiafeng Chen, Xiaohong Chen, and Elie Tamer

December 2021

COWLES FOUNDATION DISCUSSION PAPER NO. 2319

COWLES FOUNDATION FOR RESEARCH IN ECONOMICS
YALE UNIVERSITY
Box 208281
New Haven, Connecticut 06520-8281

http://cowles.yale.edu/

Efficient Estimation of Average Derivatives in NPIV Models: Simulation Comparisons of Neural Network Estimators *

Jiafeng Chen[†] Xiaohong Chen[‡] Elie Tamer[§]

First draft: September 2019 - Revised draft: December 2021

Abstract

Artificial Neural Networks (ANNs) can be viewed as *nonlinear sieves* that can approximate complex functions of high dimensional variables more effectively than linear sieves. We investigate the computational performance of various ANNs in nonparametric instrumental variables (NPIV) models of moderately high dimensional covariates that are relevant to empirical economics. We present two efficient procedures for estimation and inference on a weighted average derivative (WAD): an **o**rthogonalized **p**lug-in with **o**ptimally-weighted **s**ieve **m**inimum **d**istance (OP-OSMD) procedure and a sieve efficient score (ES) procedure. Both estimators for WAD use ANN sieves to approximate the unknown NPIV function and are root-n asymptotically normal and first-order equivalent. We provide a detailed practitioner's recipe for implementing both efficient procedures. This involves the choice of tuning parameters for the unknown NPIV, the conditional expectations and the optimal weighting function that are present in both procedures but also the choice of tuning parameters for the unknown Riesz representer in the ES procedure. We compare their finite-sample performances in various simulation designs that involve smooth NPIV function of up to 13 continuous covariates, different nonlinearities and covariate correlations. Some Monte Carlo findings include: 1) tuning and optimization are more delicate in ANN estimation; 2) given proper tuning, both ANN estimators with various architectures can perform well; 3) easier to tune ANN OP-OSMD estimators than ANN ES estimators; 4) stable inferences are more difficult to achieve with ANN (than spline) estimators; 5) there are gaps between current implementations and approximation theories. Finally, we apply ANN NPIV to estimate average partial derivatives in two empirical demand examples with multivariate covariates.

JEL Classification: C14; C22

Keywords: Artificial neural networks; Relu; Sigmoid; Nonparametric instrumental variables; Weighted average derivatives; Optimal sieve minimum distance; Efficient influence; Semiparametric efficiency; Endogenous demand.

*We are grateful to Vanya Klenovskiy for excellent research assistance in trying our code and producing the empirical Strawberry demand estimates and to Josh Purtell in partially checking our code. We thank Denis Chetverikov for insightful discussions at the Gary Chamberlain online seminar. We also thank A. Babii, S. Bonhomme, E. Ghysels, S. Han, G. Imbens, P. Kline, O. Linton, W. Newey, M. Pelger, P. Sant'Anna, Y. Sun, A. Timmermann, D. Xiu, Y. 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 a variety of ANN architectures has been established by Hornik, Stinchcombe and White (1989) and many 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 computer vision, natural language processing and other areas, with complex nonlinear relations among many covariates and huge data sets of high quality.[1] Many problems where deep layer ANNs are extremely effective involve prediction problems (i.e. estimating conditional means or densities)—or problems in which nuisance parameters are themselves predictions. Recently, Farrell *et al.* (2018) and Athey *et al.* (2019), among others, have applied multi-layer ReLU ANNs to estimate average treatment effects under unconfoundedness and demonstrated their good performance in estimating unknown conditional means and densities of multivariate covariates. It remains to be seen whether ANNs are similarly effective for structural estimation problems with nonparametric endogeneity.

To that end, we consider semiparametric efficient estimation and inference for a weighted average (partial) derivative (WAD) of a nonparametric instrumental variables regression (NPIV) via ANN sieves. Specifically, we assume an unknown structure function h satisfies the NPIV model: $\mathbb{E}[Y_1 - h(Y_2) \mid X] = 0$, where Y_2 is a continuous random vector of moderately high dimension (including endogenous regressors that are excluded from X), and X is a vector of moderately high dimensional conditioning variables. We are interested in efficient estimation and inference for a WAD parameter of the smooth NPIV function $h(Y_2)$, without assuming that $h(Y_2)$ is a known sparse function of a moderately high dimensional covariates Y_2. WADs of structural relationships are linked to (cross) elasticities of endogenous demand systems in economics, finance, and business. It is essentially a treatment effect parameter under confounding and endogenous continuous treatment. Although there is a huge literature on efficient estimation of the average treatment effect and other causal parameters under unconfoundedness, there are far fewer results on efficient estimation and inference on the average treatment effect in nonparametric models with endogenous continuous treatment.

This paper makes three contributions. First, we present two classes of efficient estimators for WADs of NPIV models where unknown $h_0(Y_2)$ is approximated by ANN sieves: the optimally weighted sieve minimum distance estimators and the efficient score-based estimators. Under some regularity conditions both types of estimators are root-n asymptotically normal, semiparametrically efficient, and hence are first-order equivalent. Second, we detail a *practitioner's recipe* that include

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

a step by step guide for implementing these two classes of estimators. Third, and perhaps most importantly, we present a large set of Monte Carlo results on finite-sample performances of various ANN estimators. These are implemented using increasingly complex designs, such as NPIV function containing up to 13 continuous covariates (including endogenous regressors), various nonlinearities and correlations among the covariates.

We now briefly introduce the two classes of efficient estimation procedures that we consider. Both procedures are inspired by the semiparametric efficiency bound characterization in Ai and Chen (2012) (henceforth AC12) for the WAD of the unknown $h(Y_2)$ in a NPIV model $\mathbb{E}[Y_1 - h(Y_2) \mid X] = 0$. The first procedure is based on minimizing an optimal criterion, the optimally-weighted orthogonalized *sieve minimum distance* (SMD) criterion. This procedure is numerically equivalent to a semiparametric two-step procedure, where the unknown NPIV function $h(\cdot)$ is estimated via an optimally weighted SMD in the first step, and the WAD of $h(\cdot)$ is estimated using a sample analogue of an orthogonalized unconditional moment (Chamberlain, 1992) in the second step, with the unknown h substituted by the optimally weighted SMD estimator from the first step. This procedure will be denoted as OP-OSMD in our paper. AC12 already introduced this procedure and presented a small Monte Carlo study demonstrating its finite-sample performance using a spline SMD in the first step when the unknown $h(\cdot)$ is a function of a scalar endogenous variable Y_2. It is unclear how this procedure will perform when Y_2 could be a continuous random vector of higher dimension.

The second procedure is based on the efficient score (equivalently, efficient influence function).[2] AC12 derived a characterization of the efficient influence (or equivalently, efficient score) for the WAD of a NPIV model. It is also the asymptotic influence function of the OP-OSMD estimator.[3] The efficient influence is the sum of the orthogonalized unconditional moment (the one used for the OP-OSMD estimator) and the Riesz adjustment piece accounting for plugging-in estimated $h(Y_2)$. Compared to simpler settings, e.g. estimating average treatment effect under unconfoundedness, the Riesz representer here has no closed-form expression, but is characterized as one solution to an optimization over an infinite-dimensional Hilbert space constructed by a norm connected to the optimally weighted minimum distance objective. The components of the efficient influence function can nonetheless be consistently estimated via sieve approximations. The procedure using the sample estimated efficient influence (i.e., efficient score) will be denoted as ES in our paper. As far as we know there is no published work on theory or simulation on the finite-sample performance of any ES estimator for the WAD in a NPIV model yet.

[2] The efficient score/influence function approach to efficient estimation has a long history in semiparametrics. See, e.g., Bickel *et al.* (1993), Section 25.8 of Van der Vaart (2000) and references therein, for an introduction.

[3] This is not surprising since the efficient influence function is unique.

In this paper we investigate the finite-sample performance of both efficient procedures when the unknown function $h(Y_2)$ is estimated via various ANN SMDs and when $h(Y_2)$ depends on moderately high dimensional continuous regressors Y_2 (some of which are endogenous). Our simulations reveal that the ANN OP-OSMD is more stable and easier to implement than ANN ES for estimation of the average (partial) derivative in a NPIV model with unknown conditional variance $\Sigma(X) \equiv \mathrm{Var}(Y_1 - h(Y_2) \mid X)$.

In practice, it could be appealing to report simpler inefficient estimators that are still consistent and $\sqrt{n}$-asymptotically normal. It is possible that computationally simpler inefficient estimators may perform better than the efficient estimators in finite samples. For the sake of comparison, we include two first-order asymptotically equivalent inefficient estimators of the WAD of a NPIV function, denoted by P-ISMD and IS respectively. The P-ISMD is a simple plug-in identity-weighted SMD estimator that was proposed in Ai and Chen (2007) (henceforth AC07). The IS is what we call "inefficient score" estimator that is based on sample analog of the asymptotic influence function of the P-ISMD estimator (derived in AC07).[4] We note that both P-ISMD and IS are asymptotically efficient for a WAD of a nonparametric regression $\mathbb{E}[Y_1|Y_2]$. However, they are no longer efficient for the WAD of a NPIV function $h(Y_2)$ identified by the conditional moment restriction $\mathbb{E}[Y_1 - h(Y_2) \mid X] = 0$ (for $Y_2 \neq X$).

We compare the finite sample performance of these efficient (OP-OSMD, ES) and inefficient (P-ISMD, IS) estimation procedures 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 $h(Y_2)$ 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 $h(Y_2)$ with respect to an endogenous variables using various ANN sieves and spline sieves. Our Monte Carlo experiments allow for comparisons along several dimensions:

- For ANN estimators, how much does ANN architecture (activation, depth, width) matter? How much do other tuning parameters matter?
- Across types of estimation procedures, how do ANN SMD estimators compare to ANN score estimators, along with alternative procedures like adversarial GMM (Dikkala *et al.*, 2020)?

[4] Different inefficient estimators of the WAD can have different asymptotic influence functions and hence different asymptotic variances. That is why we define the IS estimator based on the asymptotic influence function of the P-ISMD estimator of AC07, so that they will have the same asymptotic variance.

[5] Since both score-based estimators ES and IS are based on orthogonal moments, we also provide comparison with their cross-fitted versions. The cross-fitting orthogonal moments estimators have become very popular following (Chernozhukov, Chetverikov, Demirer, Duflo, Hansen, Newey and Robins, 2018; Chernozhukov, Escanciano, Ichimura, Newey and Robins, 2021) and others, although no published work has applied cross-fit to efficient estimation of WAD in NPIV yet.

- Within a type of estimation procedure, do ANN estimators exhibit superior finite-sample performance compared to linear sieve (e.g., spline) estimators, when dimension of Y_2 is moderately high?[6]

The main takeaways from our Monte Carlo experiments are as follows:

- Choices of hyperparameters in optimization—learning rate, stopping criterion—are delicate and can affect performance of ANN-based estimators. Nonconvex optimization could lead to unstable performances. However, certain values of the hyperparameters do result in good performance of ANN based estimators.

- We do not empirically observe systematic differences in finite-sample performances as a function of ANN architecture, within the feedforward neural network family. The importance of ANN 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.

- ANN OP-OSMD and ANN IS have smaller biases than ANN P-ISMD for the average derivative parameter.

- ANN ES and ANN cross-fitted ES are sensitive (in terms of bias) to the estimation of the optimal weighting $\Sigma^{-1}(X)$ in Riesz representer adjustment part. ANN OP-OSMD is not sensitive to the poor estimation of the optimal weighting $\Sigma^{-1}(X)$.

- Spline OP-OSMD , spline P-ISMD , spline IS and spline ES for the average derivative parameter are less biased, stable and accurate, and can outperform their ANN counterparts, even when the NPIV function $h(Y_2)$ depends on moderately high-dimensional continuous covariates Y_2 (as high as thirteen in the simulation studies).

- Generally, there seems to be gaps between intuitions suggested by approximation theory and current implementation.

Lastly, as applications to real data, we apply ANN sieve NPIV to estimate average price elasticity of a gasoline demand using the data set of Blundell, Horowitz and Parey (2012), and to estimate average derivatives of a price-quantity relation in differentiated product markets using the data set of Compiani (2019). Both applications involve nonparametric structure functions of multi-dimensional covariates (including endogenous price), and our ANN applications do not impose any semiparametric shape restrictions.

[6] To be clear, we are not speaking of "high dimension" in the $\dim(Y_2)/n \not\to 0$ sense.

Related literature on ANN NPIVs. We view various ANNs as examples of *nonlinear* sieves, which, compared to linear sieves, 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,[7] the asymptotic properties of estimation and inference based on the ANN sieve could be established by applying the general theory of sieve-based methods. The nonparametric convergence rates in Ai and Chen (2003, 2007) (henceforth, AC03, 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. Due to the small sample size and computational limitation, earlier applications in econometrics have focused on single-hidden layer (or what is now called "shallow") ANNs. For instance, Chen and Ludvigson (2009) applied single hidden layer sigmoid ANN SMD to estimate the unknown habit function in a semi-nonparametric asset pricing conditional moment model with a time series sample size of about 200 quarterly observations. To the best of our knowledge, Hartford, Lewis, Leyton-Brown and Taddy (2017) is the first paper to apply multi-layer (2 hidden layer) ANNs to estimate NPIV function. Also see the follow-up work on adversarial GMM (Dikkala, Lewis, Mackey and Syrgkanis, 2020) and the references there in. However, as documented in our simulation studies, the WAD parameter estimated via plugging in the fancy adversarial GMM estimator of $h(\cdot)$ can be biased.[8] In a project that started after our first draft, Chen, Liao and Wang (2021b) established rate of convergence for multi-layer ANN optimally weighted SMD estimation of general nonparametric conditional moment restrictions for time series data, and proposed ANN sieve quasi likelihood ratio inference for possibly slower-than-root-n estimable linear functionals. However, they do not consider efficient variance estimation for root-n estimable linear functionals of NPIV such as the WAD parameter, which is our parameter of interest.

Our simulation studies and empirical applications indicate that, although multi-layer ANNs can perform well after careful choice of tuning parameters, they have no clear advantage over single hidden layer ANNs or spline sieves for efficient estimation of WAD in a NPIV model when the unknown structure function $h(Y_2)$ is a relatively smooth function of multi-dimensional Y_2, which is likely the case in economic endogenous demand estimation. Just like the simulation paper by Lee

[7] Different ANN sieves have different approximation error rates for different function classes. See, for example, Barron (1993) and Chen and White (1999) for approximation errors rates for single hidden layer ANNs for Barron class; Yarotsky (2017), Shen *et al.* (2021b), Shen *et al.* (2021a) for approximation error rates of multi layer ReLU ANNs for typical smooth function class; Schmidt-Hieber (2019) for approximation error rates of deep layer ReLU ANNs for composition function classes.

[8] This is not surprising since the tuning parameter choice for nonparametric estimation of $h(\cdot)$ is different from that for the efficient estimation of the WAD.

et al. (1993) about the performance of single-hidden layer ANNs on testing nonlinear regression models, our paper documents that ANNs can also be one promising tool in efficient estimation and inference for causal parameters in NPIV models.

The rest of the paper is organized as follows. Section 2 introduces the model, and the two classes of efficient estimation procedures. 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 briefly concludes.

2 Efficient Estimation Procedures for Average Derivatives in NPIV Models

We first present the model and recall the semiparametric efficiency bound characterization. We then present two classes of efficient estimation procedures.

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 is the partial derivative w.r.t. the first argument and the unknown real-valued function $h_0 \in \mathcal{H}$ is identified via a conditional moment restriction[9]

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

Previously, Ai and Chen (2007) (AC07) presented a root-n consistent asymptotically normally distributed identity-weighted SMD estimator of θ_0, nonlinear sieves such as single hidden layer ANN sieve is allowed for in their sufficient conditions. AC12 presented the semiparametric efficiency bound of θ_0 and an efficient estimator based on orthogonalized optimally weighted SMD (see their section 4.2).[10] 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 $\sqrt{n}$-asymptotically normal estimator of linear functionals of NPIV in Santos (2011) fails to achieve the efficiency bound. Chen, Pouzo and Powell (2019) proposed efficient estimation of weighted average derivatives of nonparametric quantile IV regression via penalized linear sieve GEL procedure, without providing any simulation results on how their

[9] See, e.g., Newey and Powell (2003), Blundell *et al.* (2007), Andrews (2017) for identification of a NPIV model.

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

procedure performs in finite samples.

Since weighted average treatment effects under confounding and endogenous continuous treatments can be regarded as an example of the WAD in a NPIV model, it is important to conduct some detailed Monte Carlo studies to compare finite-sample performance of various efficient estimators of θ_0 when $h_0(Y_2)$ depends on multi-dimensional covariates Y_2. In this paper we present large scale simulation studies focusing on the performance of several estimators of θ_0 when $h_0(Y_2)$ is approximated via various ANN sieves and Y_2 is up to 13-dimensional vector of continuous covariates.

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 the full parameter vector as $\alpha_0 \equiv (\theta_0, h_0) \in \Theta \times \mathcal{H} \equiv \mathcal{A}$. The model can be written as the following sequential moment restriction

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

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:

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

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

We now specialize the results in AC12 to the plug-in model:

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

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

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

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

We recall the efficiency bound characterization for WAD of a NPIV model from AC12 (see their Example 3.3) 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 $J_0 \equiv \mathrm{Var}(S^*)$.

(1) If $J_0 = 0$, then θ_0 cannot 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:

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

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 optimally weighted quadratic distance of the expectation of the two moment conditions

$$m(X,\alpha) = \begin{bmatrix} \mathbb{E}[\varepsilon_1(Z,\alpha)] \\ \mathbb{E}[Y_1 - h(Y_2) \mid X] \end{bmatrix} = \begin{bmatrix} \mathbb{E}[a(Y_2)\nabla_1 h(Y_2) - \theta - \Gamma(X)(Y_1 - h(Y_2))] \\ \mathbb{E}[Y_1 - h(Y_2) \mid X] \end{bmatrix}$$

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

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

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:

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

This is important because solving the model sequentially while maintaining efficiency suggests a simple way to compute the estimators.

A sieve minimum distance estimator for $\alpha_0 = (h_0, \theta_0)$ 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 $W_0(.)$ respectively; see Sections 3 and 4 below for examples of different estimators. Let $\mathcal{H}_n$ be a sieve parameter space 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

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

This is an estimator proposed in AC12.

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 B. Either approach will lead to the following asymptotic efficient influence function expansion:

$$\sqrt{n}(\hat{\theta}-\theta_0) = \frac{1}{\sqrt{n}}\sum_{i=1}^{n}\left[a(Y_2)\nabla_1 h_0(Y_{2i}) - \theta_0 - \left(\Gamma(X_i) - \frac{\mathbb{E}[v_h^\star \mid X]}{\Sigma(X)}\right)(Y_{1i} - h_0(Y_{2i}))\right] + o_p(1). \quad (8)$$

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 the optimal direction as the following:

$$w^\star = \arg\min_w \mathbb{E}\left[\frac{1}{\sigma_0^2}(\mathbb{E}[1 + a(Y_2)\nabla_1 w + \Gamma(X)w])^2 + \frac{1}{\Sigma(X)}(\mathbb{E}[w \mid X])^2\right]. \quad (9)$$

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 (9) 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 (2021b) results can be applied for multi-layer ANN sieves.

3 Implementation of the estimators

In this section, we describe in broad strokes the implementation of the eventual estimators for the average derivative of a NPIV, 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.

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 model is:

$$\mathbb{E}\left[Y_1 - h_0(Y_2) \mid X\right] = 0. \tag{10}$$

Let $Z = [Y_1, Y_2, X]$ collect the observable random variables (in the population). The parameter of interest is $\theta_0 = \mathbb{E}\left[\nabla_1 h_0(Y_2)\right]$, where $\nabla_1 h_0(Y_2)$ is the partial derivative of h_0 with respect to its first argument, evaluated at Y_2.

We also set up notation for objects related to the sample. Let there be a random 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 $f(x)$.[11] 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 $f(x)$.

[11] 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.).

Quick map of estimation procedures We provide a simple map that connects the above model and estimation approaches to the estimators we implement 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 as in (8), and take the sample variance. (Section 3.3)
3. Score estimators [IS, ES]: Estimate the components of the influence functions as in (6). 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 $h_0(\cdot)$. We also conclude the section with a brief discussion of software implementation issues.

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 (10), 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 $P_\phi(y_1 - h(y_2))$:

$$\hat{h} = \underset{h \in \mathcal{H}_n}{\arg\min} \frac{1}{n} \big\| P_\phi[y_1 - h(y_2)] \big\|^2. \tag{11}$$

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 $\Sigma(X) \equiv \mathbb{E}[(Y_1 - h_0(Y_2))^2 \mid X]$ 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 (11) 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**:

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

See AC (2007) for the root-n asymptotic normality of this estimator, and its asymptotic linear expansion is of the form:

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

where

$$v^\star_{h,\text{id}} = \frac{-w^\star_{\text{id}}}{1 + \mathbb{E}[\nabla_1 w^\star_{\text{id}}]} \quad w^\star_{\text{id}} = \arg\min_w \left\{\mathbb{E}[\mathbb{E}[w(Y_2) \mid X]^2] + (1 + \mathbb{E}[\nabla_1 w(Y_2)])^2\right\}. \tag{12}$$

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** estimator, 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 $Y_1 - h_0(Y_2)$:

$$\Gamma(X) \equiv \mathbb{E}[(\nabla_1 h_0(Y_2) - \theta_0)(Y_1 - h_0(Y_2)) \mid X]\Sigma^{-1}(X). \tag{13}$$

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—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 $\hat{h} = \arg\min_{h \in \mathcal{H}_n} \frac{1}{n}\left\|P_\phi[y_1 - h(y_2)]\right\|_2^2$.

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 identity-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 $\hat{h} = \arg\min_{h \in \mathcal{H}_n} \frac{1}{n}\left\|P_\phi[y_1 - h(y_2)]\right\|_{\hat{W}}^2$.

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) 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

$$\psi(Z, \theta, h, \kappa) = \nabla_1 h(Y_2) - \kappa(X)(Y_1 - h(Y_2)) - \theta. \tag{14}$$

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*.

$$\kappa_{\text{ID}}(X) = \mathbb{E}[-v^\star(Y_2) \mid X] \tag{15}$$

$$v^\star(Y_2) = \frac{-w^\star(Y_2)}{1 + \mathbb{E}[\nabla_1 w^\star(Y_2)]} \tag{16}$$

$$w^\star(Y_2) = \arg\min_w \left\{ \mathbb{E}\left[(\mathbb{E}[w(Y_2) \mid X])^2 \right] + (1 + \mathbb{E}[\nabla_1 w(Y_2)])^2 \right\}. \tag{17}$$

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 (13), and

$$v^\star(Y_2) = \frac{-w^\star}{\mathbb{E}[1 + \nabla_1 w^\star + \Gamma(X) w^\star(Y_2)]} \operatorname{Var}\left[\nabla_1 h_0 - \theta_0 - \Gamma(X)(Y_1 - h_0(Y_2))\right] \tag{18}$$

$$w^\star(Y_2) = \arg\min_w \left\{ \mathbb{E}\left[\Sigma(X)^{-1} \left(\mathbb{E}[w(Y_2) \mid X]\right)^2 \right] + \left(\frac{1 + \mathbb{E}[\nabla_1 w(Y_2) + \Gamma(X) w(Y_2)]}{\sqrt{\operatorname{Var}\left[\nabla_1 h_0 - \Gamma(X)(Y_1 - h_0(Y_2)) - \theta_0\right]}} \right)^2 \right\} \tag{19}$$

are the same as (9), which are also weighted analogues of the identity weighted $w^\star$, (17).

The above formulation writes $v^\star$ as a function of $w^\star$; alternatively, we may follow the strategy in Appendix B 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

$$\hat{F} = \frac{1}{n}\sum_i \left[\nabla_1 \nu(y_{2i}) + \hat{\Gamma}(x_i)\nu(y_{2i})\right] \text{ and } \hat{R} = \frac{1}{n}\sum_i \left[\hat{\Sigma}(x_i)^{-1} P_\lambda(x_i)\nu(y_{2i})(P_\lambda(x_i)\nu(y_{2i}))'\right] \quad (20)$$

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 (12) and (17). 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

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

which corresponds to the objective function in (9)

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 (11) 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 (11), 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 (11) 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[12]

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

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[13]. In particular, (11) 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 (11).

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 ar-

[12]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.

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

chitecture 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(\cdot) # Compute dc/da and dc/db
>>> a.grad # dc/da = 2
tensor([2.])
>>> b.grad # dc/db = 1
tensor([1.])
```

4 Monte Carlo Studies

We present three Monte Carlo designs in the first subsection. We then describe exactly how we estimated the various components that are needed for the estimators in the next subsection. The last subsection discusses some Monte Carlo results. These results and additional ones are contained in Figures and Tables in the Appendix.

4.1 Design Descriptions

We consider a set of Monte Carlo experiments that combine simple but relevant designs that include high dimensional regressors. These designs are also relevant to the kinds of empirical models that are of interests to economists. We describe the three Monte Carlo designs below. They differ in whether allowing for endogeneity and various nonlinearities.

Monte Carlo 1. The first Monte Carlo Data Generating Process (DGP) is an augmentation of the design in Chen (2007). This design has a simpler functional form of h_0:

$$Y_1 = h_0(Y_2) + U = X_1 + h_{01}(R) + h_{02}(X_2) + h_{03}(\tilde{X}) + U, \quad \mathbb{E}[U \mid X_1, X_2, X_3, \tilde{X}] = 0,$$

where we generate

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

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

$$
\tilde{X} = \Phi\left(\rho(X_1 + X_2 + X_3) + \sqrt{1-\rho^2}T\right) \quad T \sim \mathcal{N}(0, \Sigma), \tag{21}
$$

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 5 and 10 to make the estimation problem more difficult. Note 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 $\tilde{X}$. We have also tried different conditional variance of U, the simulation results are similar.

To connect with the notation in the previous sections, let $Y_2 = [X_1, R, X_2, \tilde{X}]$ and $X = [X_1, X_2, X_3, \tilde{X}]$. The parameter of interest is $\theta_0 = \mathbb{E}\left[\frac{\partial h_0(Y_2)}{\partial X_1}\right] = 1$.

Monte Carlo 2. The second Monte Carlo DGP is:

$$
Y_1 = h_0(Y_2) + U = R_1 + h_{01}(R_2) + h_{02}(X_2) + h_{03}(\tilde{X}) + U, \quad \mathbb{E}[U \mid X_1, X_2, X_3, \tilde{X}] = 0,
$$

where

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

In addition, $\tilde{X}$ is generated as in (21), with the correlation ρ set to $\{0, 0.5\}$. In the high dimensional design, we set the dimension of $\tilde{X}$ to be 10 and so the model will have 13 continuous regressors. To connect with the notation in the previous sections, let $Y_2 = [R_1, R_2, X_2, \tilde{X}]$ and $X = [X_1, X_2, X_3, \tilde{X}]$. The parameter of interest is $\theta_0 = \mathbb{E}\left[\frac{\partial h_0(Y_2)}{\partial R_1}\right] = 1$.

Monte Carlo 3. We modify Monte Carlo 2 with two changes that allows for some nonlinearity of h_0 in R_1. In particular:

(a) R_1 enters $h_0(\cdot)$ through R_1^2. The parameter of interest is $\theta_0 = \mathbb{E}\left[\frac{\partial h_0(Y_2)}{\partial R_1}\right] = \mathbb{E}[2R_1] = 1$.

(b) R_1 enters $h_0(\cdot)$ 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 $b = 16$. The parameter of interest is

$$
\theta_0 = \mathbb{E}\left[\frac{\partial h_0(Y_2)}{\partial R_1}\right] = \mathbb{E}\left[R_1 + \frac{f(a(X_2 - b))}{2C}\right] = \frac{1}{2} + \frac{1}{2} = 1.
$$

Next, we provide a step by step guidance on how to implement the estimators.

4.2 Implementation details

We explain here the exact choice of estimators that we used for these Monte Carlo designs. A detailed overview is presented in Table 4. Various ANN SMD estimators for h have additional tuning parameters regarding nonlinear optimization, which are described in Table 1.

Monte Carlo	Learning rate	# steps
1	0.001	1500–2000
2	0.01	3000–5000
3	0.01	7000–10000

Table 1: Optimizer parameter choices for ANN SMD for h. The number of steps is of the form (minimum number of steps)–(maximum number of steps), where an ad hoc stopping rule is used when the step size is in between, based on how much progress the optimization procedure is making. In our experience, in practice, the optimizer stops at near the minimum number of steps. We use PyTorch's implementation of the Adam optimizer (`torch.optim.Adam`) throughout our experiments.

1. Figure 1 reports Monte Carlo means and standard deviations for the design in Monte Carlo 1, with ANN SMD estimators of different network architectures. In particular, we make the following choices for estimation of various nuisance parameters.

   (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 $\phi(\cdot)$ for instruments

      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),[14] and $\phi_2(X, \tilde{X}) = [\tilde{X}, \tilde{X}^2, (X_i\tilde{X}_j)_{i,j}]$ contains second-order polynomials for $\tilde{X}$ and interactions $X_i\tilde{X}_j$.

   (b) Optimally-weighted SMD with orthogonalized plug-in: $\hat{\theta}_{\text{OP}}\left(\hat{h}_{\text{OSMD}}, \hat{\Gamma}\right)$ defined in Section 3.1. We specify estimation details for the nuisance functions $\Sigma(X), \Gamma(X)$:

      i. $\hat{\Sigma}(\cdot)$: Form the squared residuals from the identity-weighted estimator $v \equiv (y_1 - \hat{h}_{\text{ISMD}}(y_2))^2$ and estimate Σ by $k = 5$-nearest neighbors.

      ii. $\hat{\Gamma}(\cdot)$: Given an estimate $\hat{\Sigma}$, it suffices to estimate

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

      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 it 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 ANN SMD estimators under a variety of model specifications on true h_0. The instrument sieve basis used is the same as in 1(a)i.

[14] i.e. $\phi_1(X_1, X_2, X_3) = [1, X_1, X_1^2, X_1^3, X_1^4, (X_1 - 0.5)_+^4, X_2, \ldots, X_2^4, (X_2 - 0.5)_+^4, X_3, \ldots, X_3^4, (X_3 - 0.1)_+^4, (X_3 - 0.25)_+^4, (X_3 - 0.5)_+^4, (X_3 - 0.75)_+^4, (X_3 - 0.9)_+^4, X_1X_3, X_2X_3, X_1(X_3 - 0.25)_+^4, X_2(X_3 - 0.25)_+^4, X_1(X_3 - 0.75)_+^4, X_2(X_3 - 0.75)_+^4.]$, where $(\cdot)_+ = \max(\cdot, 0)$.

(a) The first column of Figure 2 reports results where the ANN SMD estimators are computed assuming h_0 is fully nonparametric. Estimator choices are the same as in Item 1 for Figure 1.

(b) The second column of Figure 2 follows Section 3.4 in that we assume a partially linear structure on h_0, which is of the form $R_1\theta + h(R_2, X_2, \tilde{X})$. The estimator of $\hat{\Sigma}(\cdot)$ required for the OSMD is the same as in Item 1(b) for Figure 1.

(c) The third and the fourth columns of Figure 2 follow Section 3.4 in that we maintain the partially additive structure on h_0, which is of the form $R_1\theta + h_1(R_2) + h_2(X_2) + h_3(\tilde{X})$, where the unknown $h_3(\cdot)$ is approximated via ANNs. We use ANN sieves to approximate the scalar functions h_1, h_2 in the 3rd column, whereas the fourth column uses spline sieves to approximate h_1, h_2. The estimator of $\hat{\Sigma}(\cdot)$ required for the OSMD is the same as in Item 1(b) for Figure 1.

3. Figures 5 and 6 reports Monte Carlo means and standard deviations for a wide class of estimators (not limited to ANN SMD) for Monte Carlo 2.

   (a) ANN SMD: Follow Item 1 for Figure 1.

   (b) Spline SMD:

      i. Let $\lambda(x)$ be a spline basis for the instrument space of X, and let $\nu(y_2)$ be a spline basis for the structural function $h(\cdot)$. Both λ and ν are of the forms where each entry expands into a Spline$(k, 2)$ basis,[15] and pairwise interactions (of the form $x_i x_j$) are included in lieu of tensor product splines. The choice of order k for $\lambda(x)$ is 1 more than that for $\nu(y_2)$.

      ii. Given λ, ν, we estimate P-ISMD, OP-OSMD as in Item 1, where we optimize over candidate structural functions of the form $\nu(\cdot)'\gamma$, and estimate Σ and Γ by least squares projections onto the instrument sieve λ.

   (c) Score/influence function estimators: Let $\lambda(x), \nu(y_2)$ be the spline bases used for the spline SMD in Item 3(b).

      i. IS:

         A. Estimate $\hat{h}_{\text{ISMD}}$ as in Item 1(a) for ANN ISMD and as in Item 3(b) for spline ISMD.

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

B. $v^\star(y_2)$ can be computed by solving (17). 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 (17) 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.

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

D. Plug $\hat{\kappa}_{\text{ID}}(x)$ and $\hat{h}$ to the inefficient influence function and compute $\hat{\theta}_{\text{IS}}$.

ii. ES:

A. Estimate $\hat{h}, \hat{\Gamma}$ as in Item 1(b) for ANN OSMD and as in Item 3(b) for spline OSMD.

B. Estimate $v^\star$ by (20).

C. Estimate Σ

D. Form $\hat{\kappa}_{\text{EIF}}(x) = \hat{\Gamma}(x) - P_\lambda[\hat{v}^\star(y_2)]\hat{\Sigma}(x)^{-1}$, where Σ estimated via $k(n)$-nearest neighbors, with $k(n) > 5$.

E. Plug $\hat{\kappa}_{\text{EIF}}(x)$ and $\hat{h}$ to the efficient influence function and compute $\hat{\theta}_{\text{ES}}$.

(d) AGMM: First we apply Dikkala, Lewis, Mackey and Syrgkanis (2020) code to estimate structural function h_0 by $\hat{h}_{\text{AGMM}}$. Then compute the simple plug-in $\hat{\theta}_{\text{SP}}\left(\hat{h}_{\text{AGMM}}\right)$ defined in Section 3.1.

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 and tables below.

4.3.1 Performance of point estimates in terms of (Monte Carlo) bias and variance

Figure 1 plots the performance of various ANN SMD estimators in terms of mean $\pm$ 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 estimators 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 Figure 3, 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 nonlinearly (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 (Dikkala *et al.*, 2020) exhibit some bias. In panel (b), spline remains well behaved and so are the 1L sigmoid ANN estimators. In Figure 4 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.

In Figures 5 and 6, we compare various implementations of ANN estimators and spline estimators in Monte Carlo 2. In Figure 5, we compare identity-weighted estimators (IS, P-ISMD, AGMM, IS-X). Note that P-ISMD and OP-OSMD are the plug in and optimal plug in SMD estimators. In Figure 6, we compare optimally-weighted estimators that are semiparametrically efficient under suitable regularity conditions (ES, OP-OSMD, ES-X).[16] 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, 2021a).[17] 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 we use. 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 very sensitive to the choice of $\hat{\Sigma}(X)^{-1}$ in the "optimally-weighted" Riesz representer estimation. Figures 7 and 8 compare the performances of ES in a variety of choices for $\hat{\Sigma}(X)$. It is interesting that the poor choice of $\hat{\Sigma}(X)^{-1}$ leads to biased estimation of ES and its cross-fitted versions.

See Online Appendix A for additional Monte Carlo results for sensitivity checks about the choice of instrument sieve bases, as well as regularization.

4.3.2 Performance of inference statistics

Tables 5 and 6 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.

In Figure 9, we examine various standard error approaches for a set of estimators in Monte

[16] 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.

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

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.

4.3.3 Overall simulation findings

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, and a researcher needs to choose many smoothing 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 appear stable, less biased generally and can outperform ANNs for NPIV models even in high dimensional cases with 13 continuous regressors.

5 Empirical Illustrations

We present two empirical applications of estimating average derivatives with respect to endogenous price of a nonparametric demand $h_0(Y_2)$ for some non-durable goods. We apply ANN sieves to approximate $h_0(\cdot)$ nonparametrically when its argument Y_2 consists of 7 covariates (for gasoline demand) and 6 covariates (for strawberry demand). In the existing literature researchers have used both data sets to estimate unknown $h_0(\cdot)$ in the model $\mathbb{E}[Y_1 - h_0(Y_2) \mid X] = 0$ by assuming h takes some parametric or semiparametric (such as partially linear) form to avoid the "curse of dimensionality" of Y_2. Although served as illustrations, our applications below are the first to estimate the endogenous demand function $h_0(\cdot)$ fully nonparametrically when $\dim(Y_2) > 5$.

5.1 Gasoline demand

We use data on gasoline demand from the 2001 National Household Travel Survey (Blundell, Horowitz and Parey, 2012). The sample we use include 4,812 observations in the full sample as in Chen and Christensen (2018). We estimate an NPIV analogue of the model (11) in Blundell, Horowitz and Parey (2012), $\mathbb{E}[Y_1 - h_0(Y_2) \mid X] = 0$ where Y_1 is the log gasoline demand, and Y_2 is a vector of 7 random variables consisting of the log gasoline price (possibly endogenous) and the other included covariates following Column (3) in Table 2 of Blundell, Horowitz and Parey (2012).

The instrument is the distance from the Gulf coast. We define the estimand as the average price derivative of the unknown structural function $h_0(\cdot)$, which has an average elasticity interpretation. blundell2012measuring via OLS, and

Table 2: 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	-1.11
	[-1.65, -0.9]	[-1.64, -0.87]	(0.22)
ReLU [3L]	-1.27	-1.25	-1.14
	[-1.65, -0.9]	[-1.64, -0.87]	(0.22)
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 7 included covariates (Y_2) 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 2 shows our estimates for the average price elasticity (and bootstrapped 0.95 confidence intervals). 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 ANN SMD estimates are slightly larger in magnitude than the spline SMD estimates and the ANN IS estimates. Within the ANN SMD estimator class, architecture choices of the networks do not appear to matter much for the result.

5.2 Strawberry demand

We also consider a setting where consumers choose two substitutable goods. We use the Nielsen dataset from Compiani (2019),[18] where consumers in California choose from strawberries, organic strawberries, and an outside option.[19] 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

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

[19] 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.

Table 3: 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. The 6 included covariates (Y_2) are: 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). A market is defined at the store-week level and there are $N = 38,800$ markets. □

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.[20]

We present the results in Table 3. As is perhaps expected from a casual intuition, estimates 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

[20]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) where in principle one can also use the neural networks based implementation in this paper in a natural way.

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.[21] 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 classes of semiparametric efficient estimators for weighted average derivatives (WADs) of nonparametric instrumental variables regressions (NPIV) of moderate and high dimensional endogenous and exogenous regressors. We have conducted detailed Monte Carlo comparisons of finite sample performance of various inefficient and efficient estimators of the WADs using various ANN sieves. The simulation studies and empirical applications confirm the theoretical advantage of ANN approximation of unknown continuous functions of moderately high-dimensional variables, after some tuning of hyper-parameters. Perhaps the most practical findings from our large amount of reported and unreported simulation studies using moderate sample sizes are as follows: the ANN efficient SMD estimators have smaller biases than those of the ANN inefficient SMD estimators, and are less sensitive to the tuning parameters than those of the ANN efficient score estimators. In addition, simple spline based estimators of WADs of NPIVs perform very well in terms of finite sample biases and variances. More research is needed to close the gap between approximation theory and finite sample computational performance in applying flexible ANNs to nonparametric models with endogeneity.

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

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.

Andrews, D. W. (2017). Examples of l2-complete and boundedly-complete distributions. *Journal of econometrics*, **199** (2), 213–220.

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.

Barron, A. R. (1993). Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information theory*, **39** (3), 930–945.

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., Chen, X. and Kristensen, D. (2007). Semi-nonparametric iv estimation of shape-invariant engel curves. *Econometrica*, **75** (6), 1613–1669.

—, 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.

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

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

—, Christensen, T. and Kankanala, S. (2021a). 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.

—, Liao, Y. and Wang, W. (2021b). *Neural Network Inference on Nonparametric Conditional Moment Restrictions with Weakly Dependent Data*. Tech. rep.

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

— and Ludvigson, S. C. (2009). Land of addicts? an empirical investigation of habit-based asset pricing models. *Journal of Applied Econometrics*, **24** (7), 1057–1093.

— 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 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*.

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*.

Lee, T.-H., White, H. and Granger, C. W. (1993). Testing for neglected nonlinearity in time series models: A comparison of neural network methods and alternative tests. *Journal of Econometrics*, **56** (3), 269–290.

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. (2011). Instrumental variable methods for recovering continuous linear functionals. *Journal of Econometrics*, **161** (2), 129–146.

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.

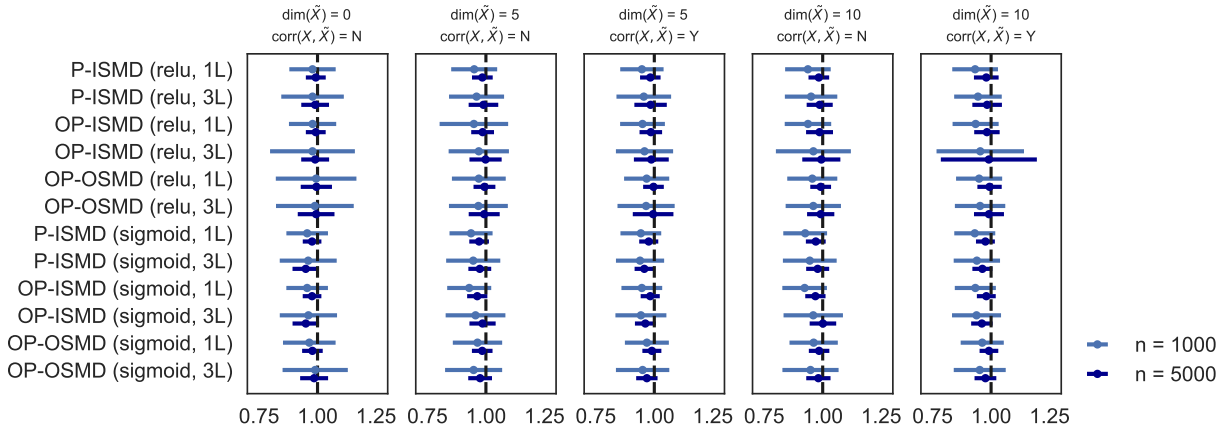
Shen, Z., Yang, H. and Zhang, S. (2021a). Neural network approximation: Three hidden layers are enough. *Neural Networks*, **141**, 160–173.

—, — and — (2021b). Optimal approximation rate of relu networks in terms of width and depth. *arXiv preprint arXiv:2103.00502*.

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

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

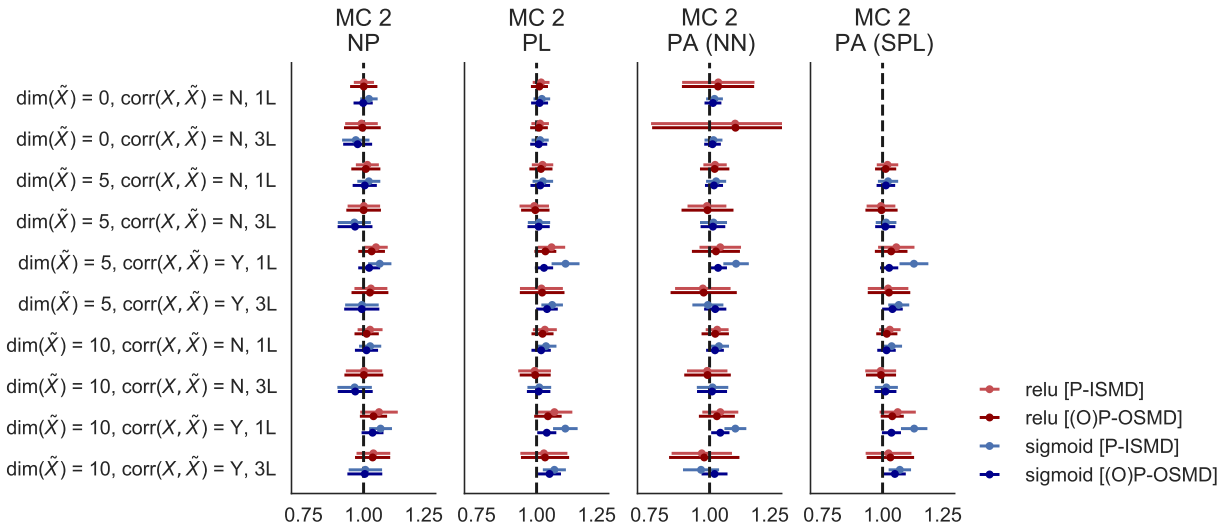
Figure 1: ANN SMD estimators for the average derivative parameter in Monte Carlo 1



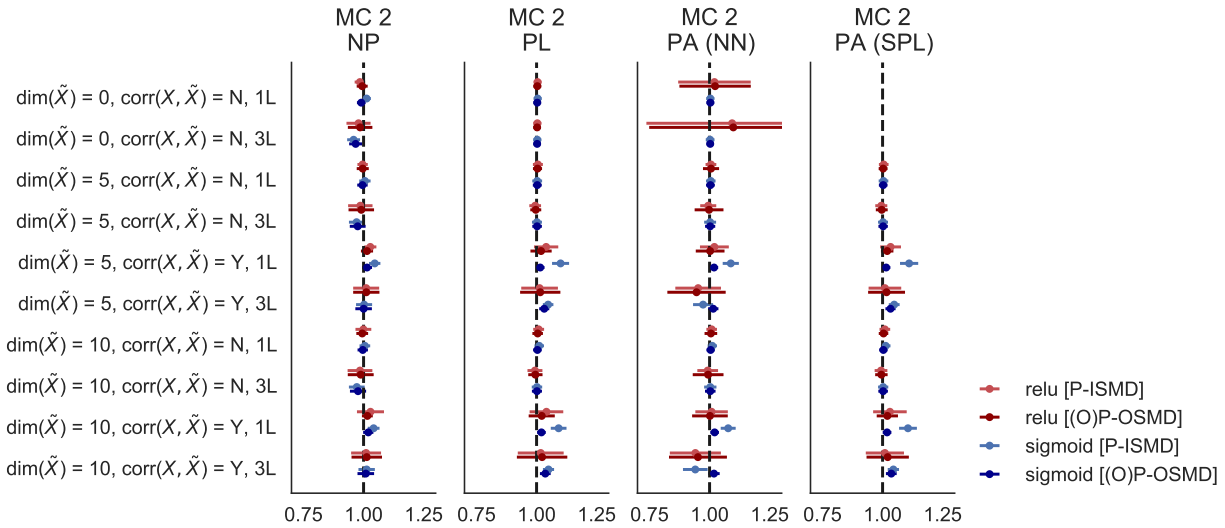
Notes. Monte Carlo Mean ±1 Monte Carlo standard deviation across 1,000 replications. □

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 $\{\text{ReLU}, \text{Sigmoid}\} \times \{\text{1L, 3L}\} \times \{\text{10W}\}$. 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 3 across a variety of NPIV estimators

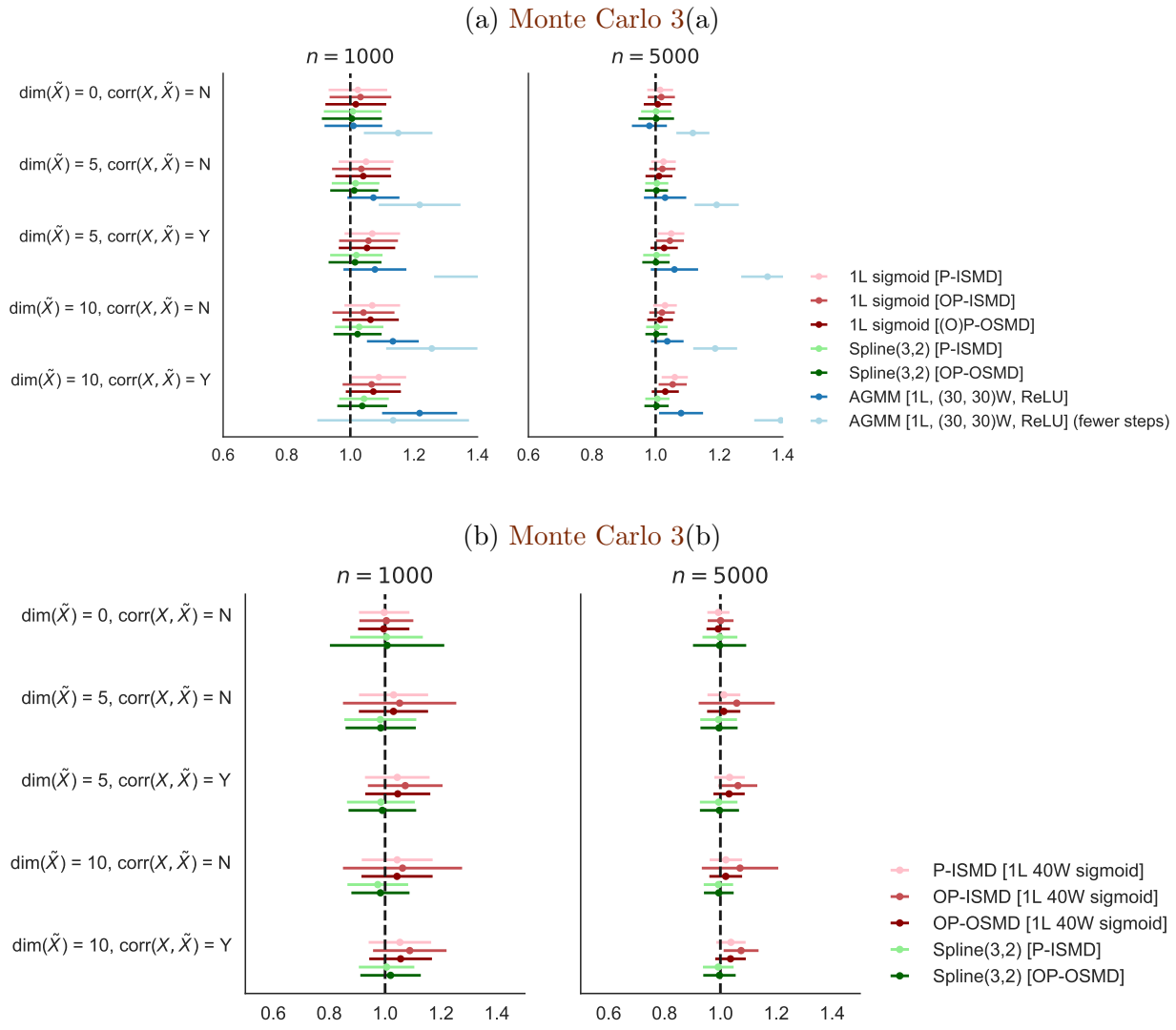
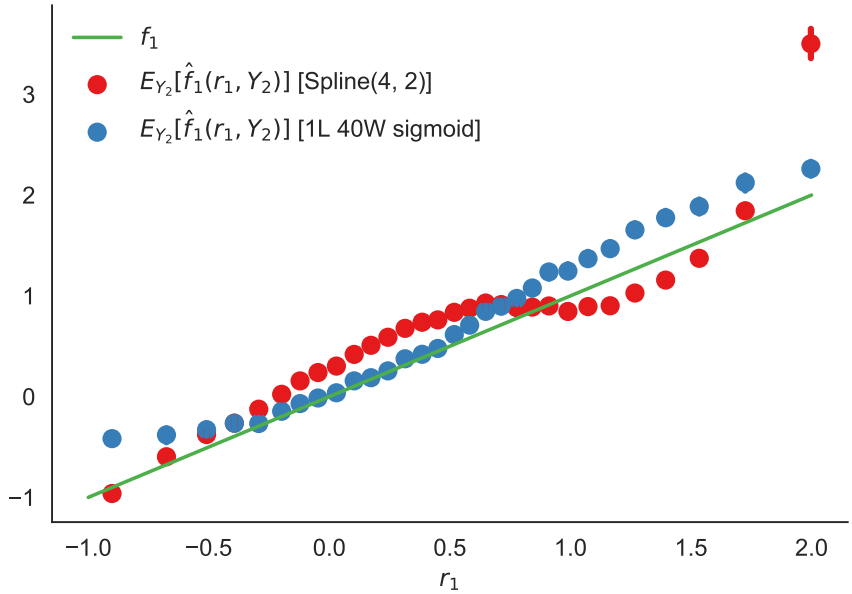
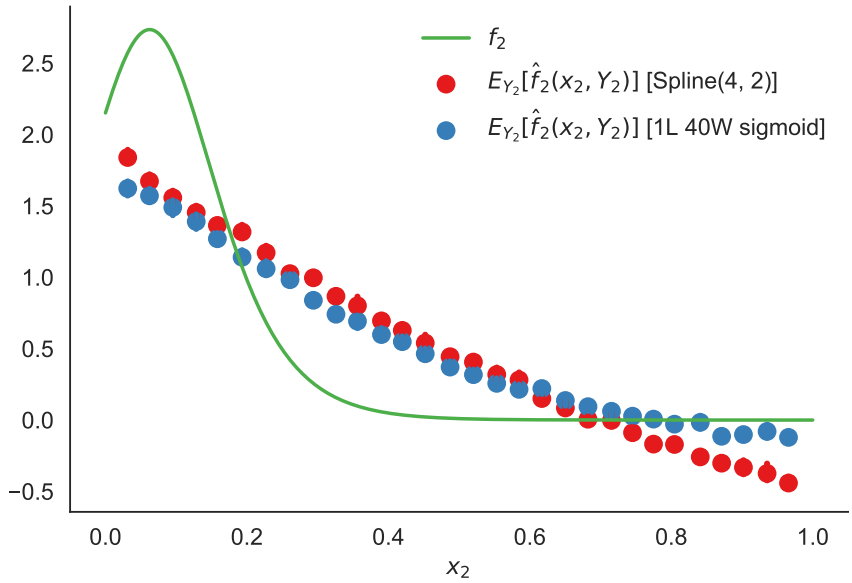


Figure 4: Estimation quality of the partial derivative function in Monte Carlo 3(b) across a variety of estimators

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

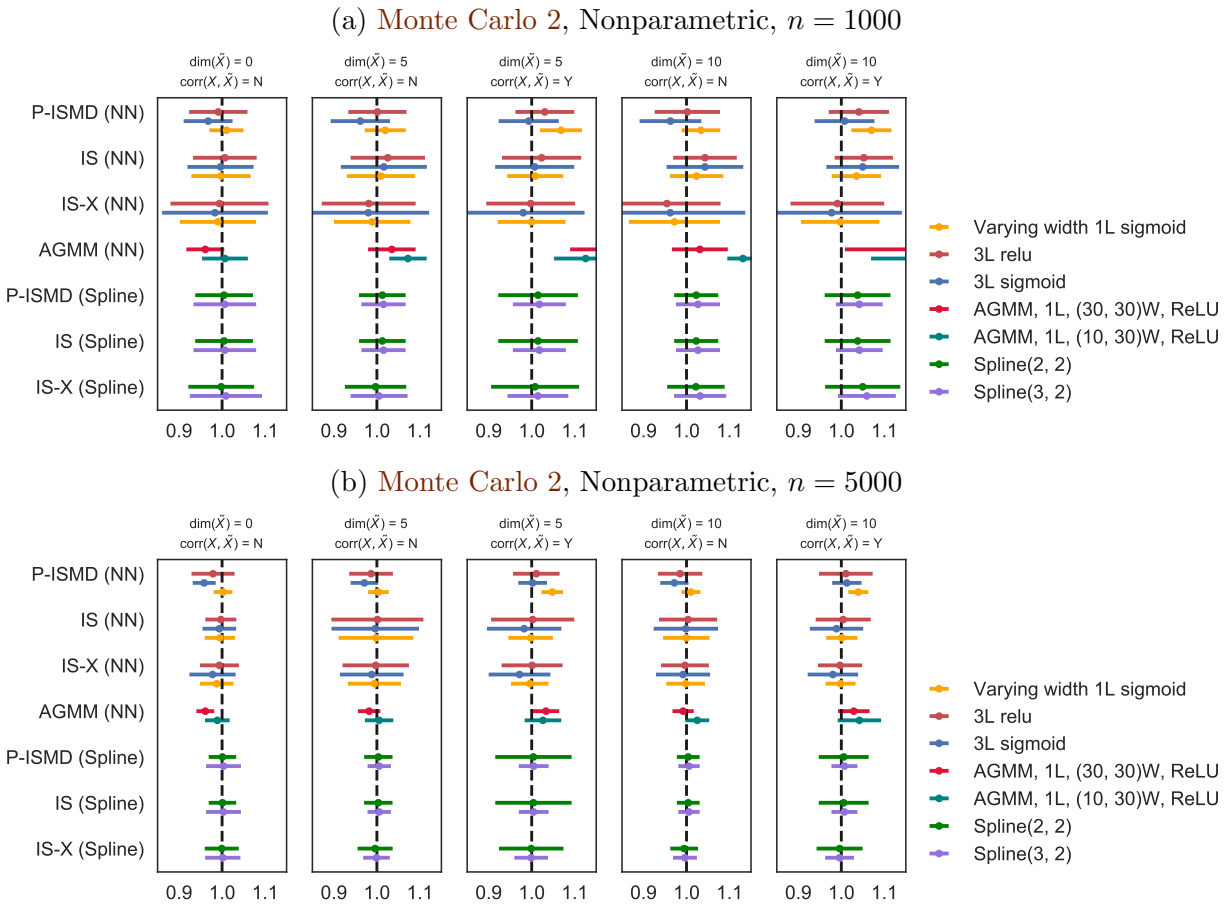


(b) Estimated f_2 versus true f_2. Single sample for $N = 10,000$



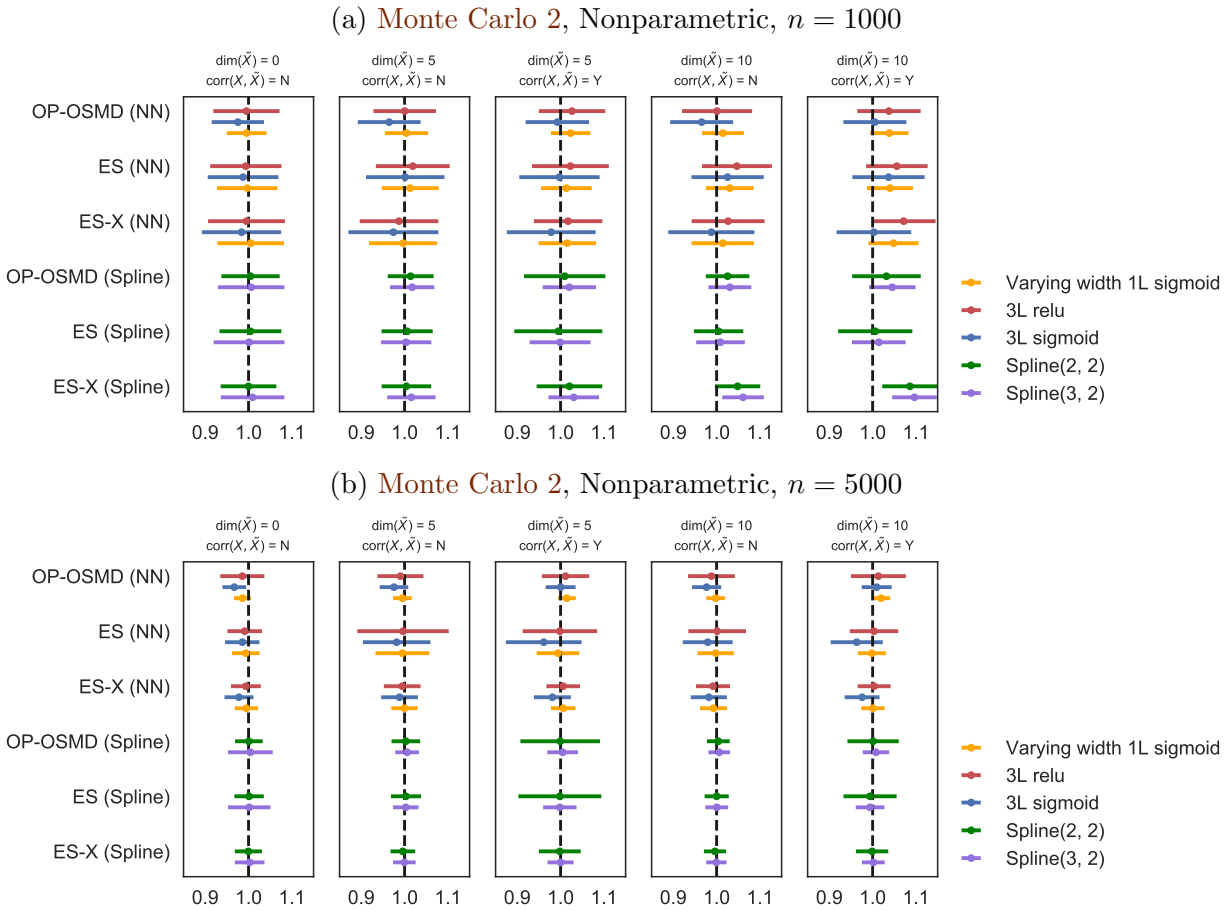
Notes. In the DGP Monte Carlo 3(b), the partial derivative $\nabla_1 h_0$ is of the form $f_1(R_2) + f_2(X_2)$, and we evaluate performance estimating f_1, f_2. Estimated f_1 is calculated by taking $\nabla_1 \hat{h} - f_2(x_2)$. We plot expectation marginalizing over variables other than r_1. Estimated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

Figure 5: 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 4. □

Figure 6: 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 4. □

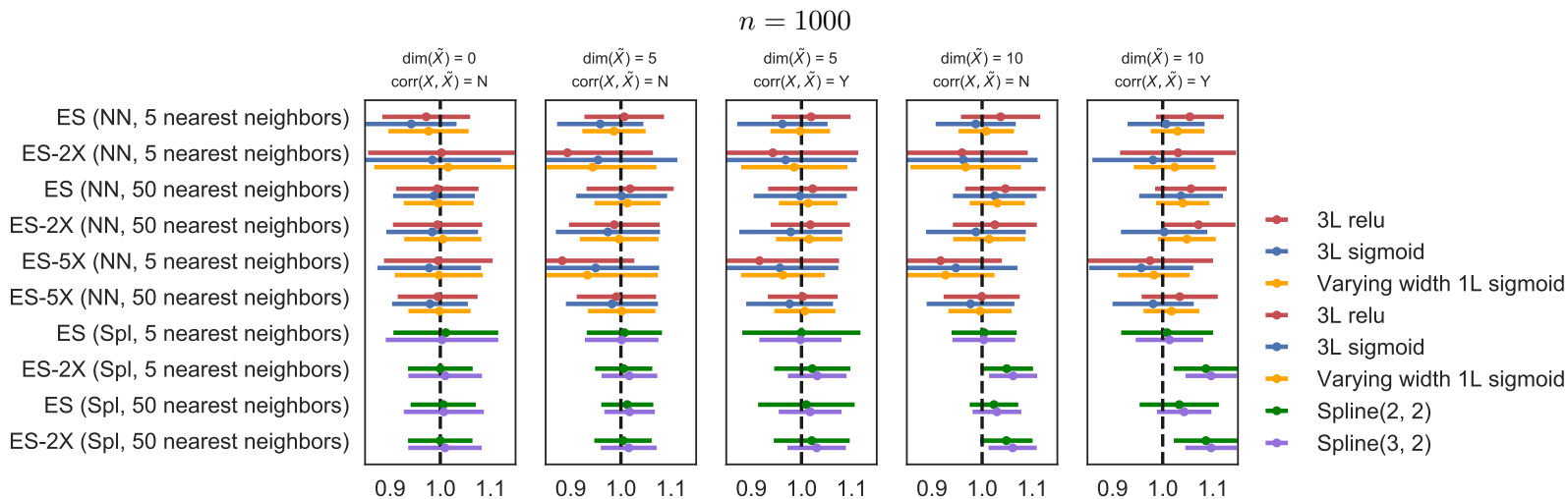


Figure 7: Performance of ES with different estimators for $\Sigma(X)^{-1}$ in the score expression

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. □

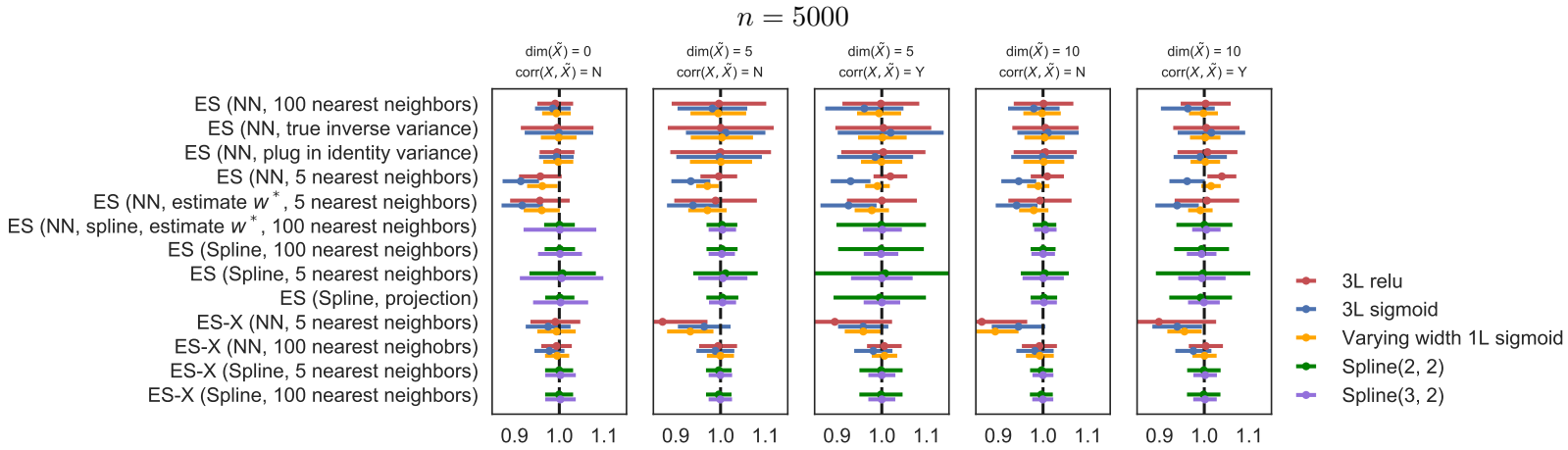


Figure 8: Performance of ES with different estimators for $\Sigma(X)^{-1}$ in the score expression

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]	$v^\star$ [Score]
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 (ϕ) 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 (17) with Spline(3, 2) (λ)
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 (20) with Spline(3, 2) (λ)
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 (λ) for the instruments	Projection of demeaned $(\nabla\hat{h}-\overline{\nabla\hat{h}})((y-\hat{h})-\overline{(y-\hat{h})})$ on instrument basis (λ) 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 (17) with same spline basis (λ) 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 (20) with same spline basis (λ) 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 4: Estimation of additional nuisance parameters

Table 5: SMD inference results for P-ISMD and OP-OSMD average derivative parameter in Monte Carlo 2, $n = 1000$

				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. □

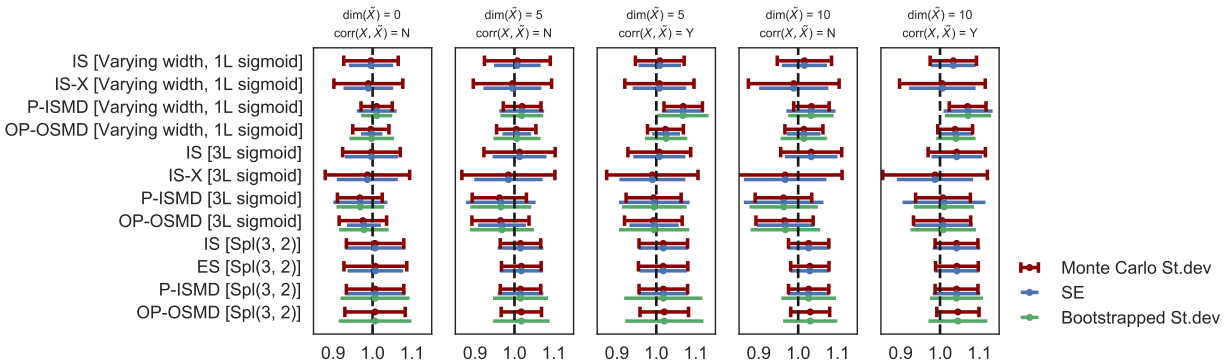
Table 6: SMD inference results for P-ISMD and OP-OSMD average derivative parameter in Monte Carlo 2, $n = 5000$

				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

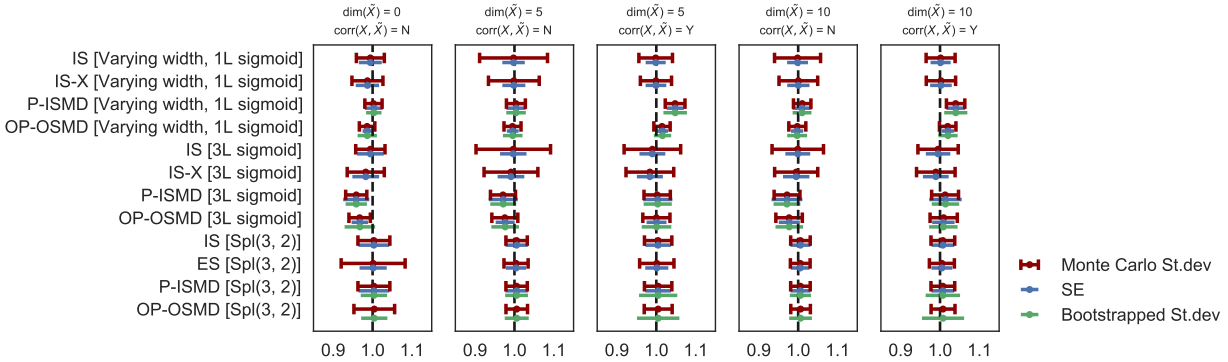
Notes. 1000 Monte Carlo replications. Bootstrap CIs based on a single replication. □

Figure 9: Inference quality of average derivatrive parameter in Monte Carlo 2 across a variety of estimators

(a) Monte Carlo 2, Nonparametric, $n = 1000$



(b) Monte Carlo 2, Nonparametric, $n = 5000$



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. □

A Appendix: Additional Monte Carlo Results

In this Appendix we provide additional sensitivity checks against choice of instrument sieve basis and regularization in ANN SMD estimation of the unknown NPIV function h.

Instrument basis. Figures 10 and 11 are replicates of Figures 1 and 2 respectively, except for slightly smaller instrument sieve bases. Specifically, we only use ϕ_1 in item 1(a)i in Section 4.2 as the basis, as opposed to using both ϕ_1, ϕ_2.

We see that the ANN SMD estimates for the simpler Monte Carlo Design 1 are not sensitive to the choice of instrument sieves, while the ANN SMD estimates for Monte Carlo Design 2 are

slightly more sensitive to the choice of instrument sieves.

Regularization. We also investigated the role that regularization via penalization schemes[22]. This is complicated in our settings since for instance for the optimal approaches, we need to estimate various unknown functions (in addition to $h(.)$, we have the Riesz representer and the skedastic function $\Sigma(X)$). In recent work, (Chernozhukov *et al.*, 2021, 2018) there is concern about the bias that is induced by large levels of regularization. Regularization on the other hand introduces additional tuning parameters that are not easy to tune in our setups.

We use the design in Monte Carlo Design 2 and include two kinds of regularizations. First, we add a penalty on the weights for net with strength λ. For the score based estimators, we also $(L^2))$ regularize the sieve estimation of the Riesz representer of v^* with a penalty that depends on λ. We vary the strength of both these penalties to examine the effect that these might have on performance. The results are provided for the identity-weighted and optimally-weighted ANN estimators in Figures 12 and 13 respectively[23]. We vary the level of regularization applied in the SMD estimation procedure. For large levels of regularization, plugging in from SMD estimators does seem to incur some bias, and we use cross fit procedures to try to mitigate the bias.

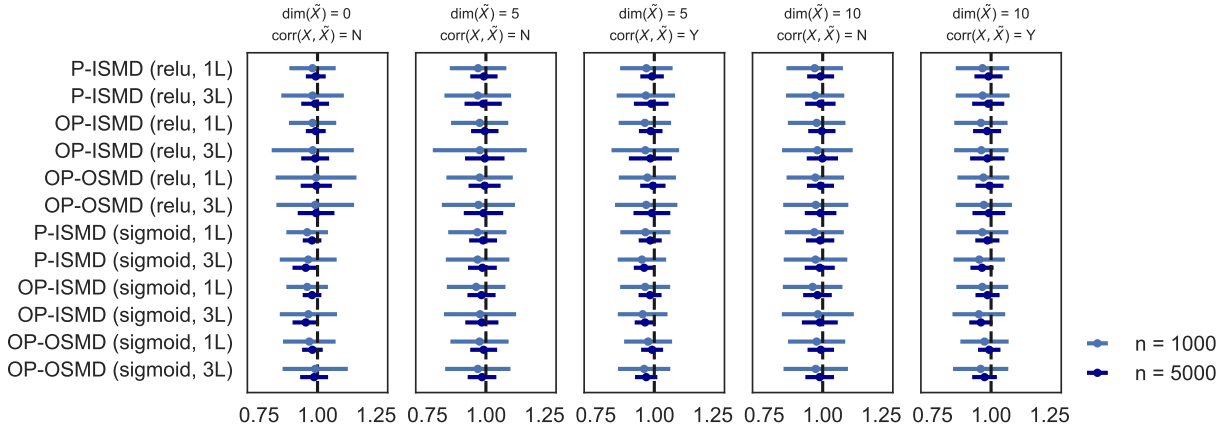
The results of inefficient (or identity-weighted) estimators are presented in Figure 12. In the large dimensional case (the last column), we see a slight bias especially with larger λ $(= 10^{-3})$ and see there that crossfit results in somewhat reduced bias. But on the whole, the performance of the estimators seem similar across designs and regularization strengths and that they also compare well with no regularization results for the same design in Figure 5.

The efficient procedures require in addition an estimator for the optimal weighting matrix and the results are given in Figure 13. The performance across different levels of regularization appear to be reasonable except that estimating $\Sigma(X)^{-1}$ with 100 nearest neighbor seems to perform better across various specification than using a 5 nearest neighbors. Also, note that the performances of cross-fitted efficient score estimators are not immune to other choices of tuning parameters, including the estimation of $\Sigma(X)^{-1}$ that is explored in Figure 8. On the other hand, for small values of regularization $(\lambda = 10^{-4})$, all estimators perform similarly as they do in Figures 6 and 8.

[22]In fact, in our experience, the regularization makes certain neural architectures more prone to optimization issues, though it may be due to bad hyperparameter settings in the optimization algorithm.

[23]Our ability to apply regularization is limited by the need for the estimation procedure to be efficiently *end-to-end trainable*. In practice, this requirement amounts to needing the regularization term to be differentiable functions in the neural network weights, whose gradient does not involve higher-order derivatives. In contrast, smoothing-spline type regularization that penalizes the smoothness of the function h directly—e.g. by penalizing $\|\nabla h\|_1$—is difficult to implement. In our experiments, we apply L^2 penalty on the neural network weights.

Figure 10: ANN SMD estimators for Monte Carlo 1 with smaller instrument basis

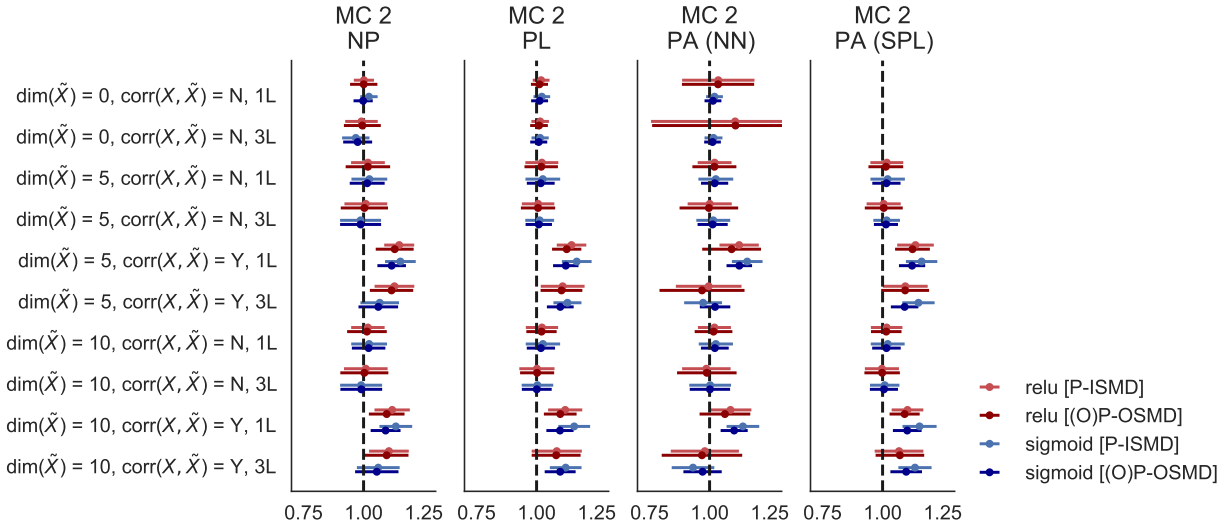


Notes. Monte Carlo Mean ±1 Monte Carlo standard deviation across 1,000 replications.

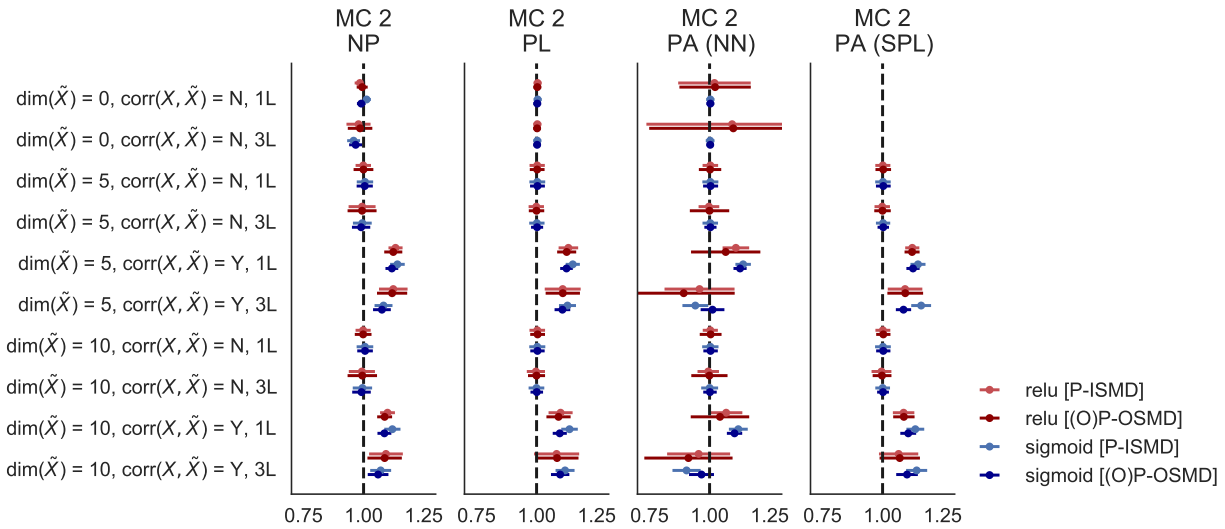
□

Figure 11: ANN SMD estimators for Monte Carlo 2 with smaller instrument basis

(a) $n = 1000$ (Smaller instrument basis)



(b) $n = 5000$ (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 $\{\text{ReLU}, \text{Sigmoid}\} \times \{1\text{L}, 3\text{L}\}$. 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. □

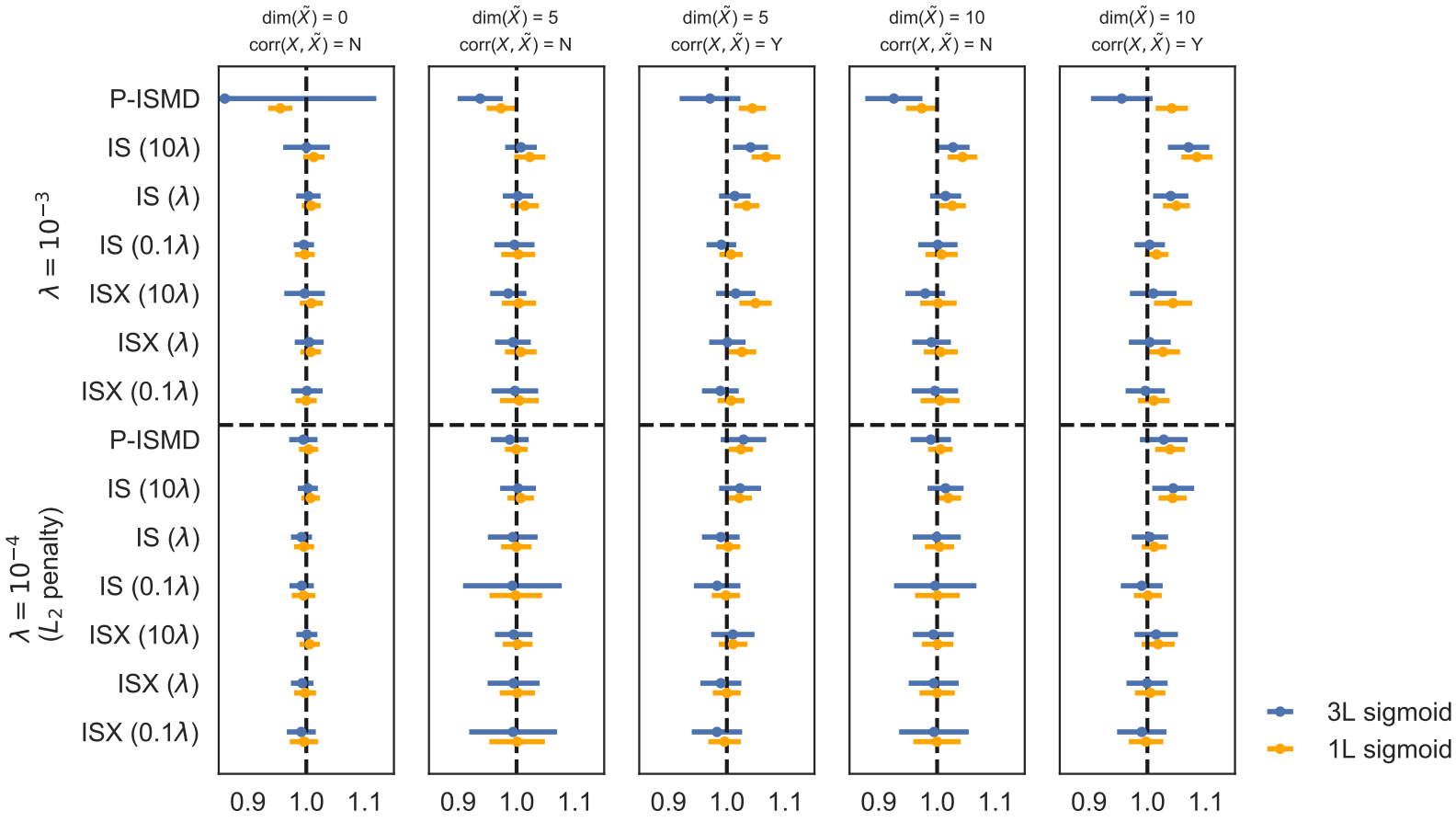


Figure 12: Performance of identity-weighted estimators with regularization

Notes. Monte Carlo Mean ±1 Monte Carlo standard deviation across 1,000 replications on $n = 5000$ sample size. Neural networks are trained with L_2-regularization on the weights with strength λ. Sieve estimation of the Riesz representer $v^\star$ is also (L^2) regularized with varying strengths relative to λ. ReLU networks encountered substantial optimization issues, and their poor performances are omitted. □

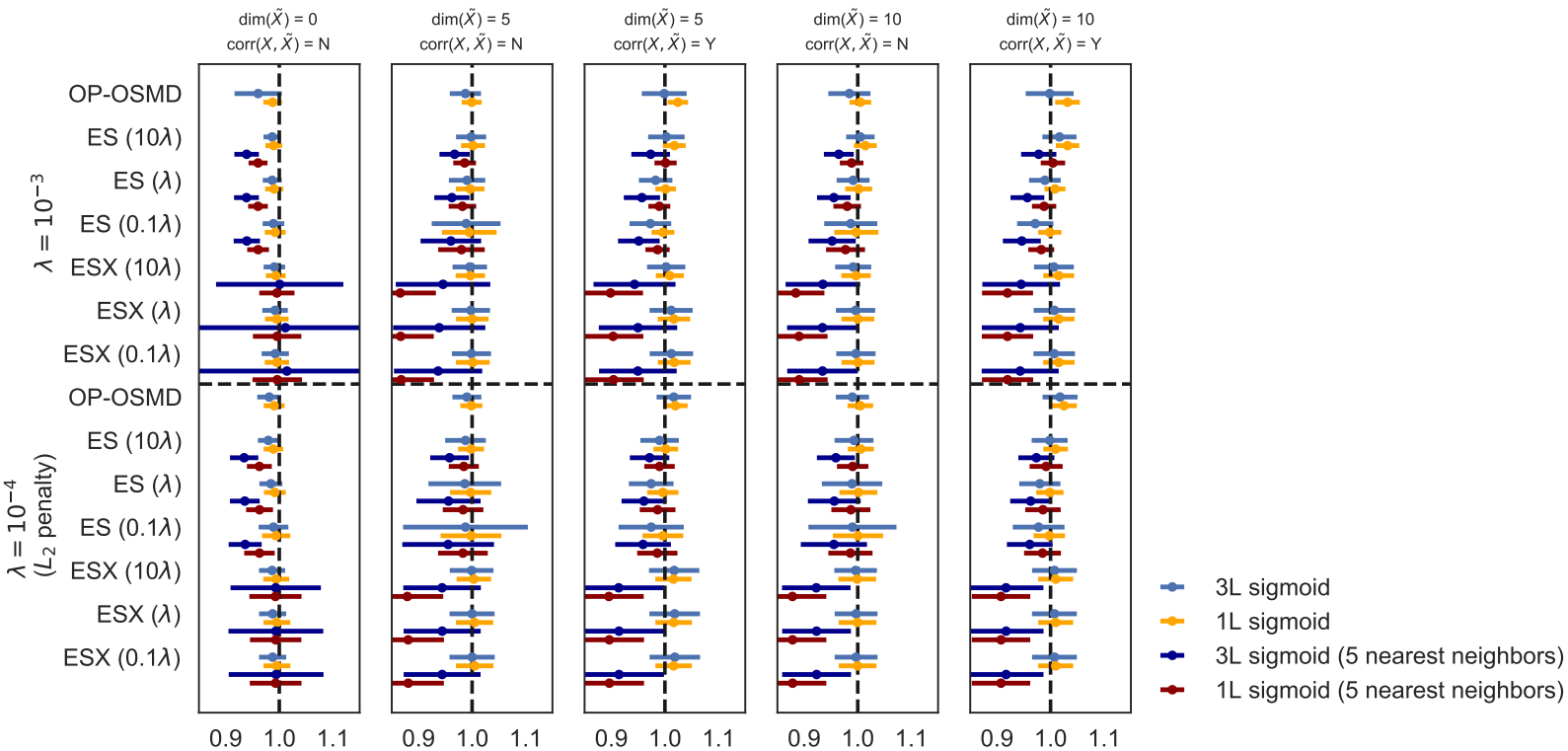


Figure 13: Performance of efficient estimators with regularization

Notes. Monte Carlo Mean ± 1 Monte Carlo standard deviation across 1,000 replications on $n = 5000$ sample size.
Neural networks are trained with L_2-regularization on the weights with strength λ. Sieve estimation of the Riesz representer $v^\star$ is also (L^2) regularized with varying strengths relative to λ. ReLU networks encountered substantial optimization issues, and their poor performances are omitted.
We also vary estimators for $\Sigma(X)^{-1}$ in the adjustment term for ES and ES-X estimators (5 vs. 100 nearest neighbors). □

B Appendix: Analysis of the optimally weighted SMD as sequential GMM

Recall that we can view the SMD estimator as a plug-in:

$$\hat{\theta} - \theta_0 = \frac{1}{n}\sum_{i=1}^{n}\left[a(Y_{2i})\nabla_1\hat{h}(Y_{2i}) - \theta_0 - \hat{\Gamma}(X_i)(Y_{1i} - \hat{h}(Y_{2i}))\right]$$

Again, we linearize

$$\hat{\theta} - \theta_0 \approx \frac{1}{n}\sum_{i=1}^{n}\left[a(Y_{2i})\nabla_1 h_0(Y_{2i}) - \theta_0 - \Gamma(X_i)(Y_{1i} - h_0(Y_{2i})) + \frac{d\mathbb{E}[\varepsilon(Z,\alpha_0)]}{dh}[\hat{h} - h_0]\right]$$

and define

$$v \mapsto \frac{d\mathbb{E}[\varepsilon(Z,\alpha_0)]}{dh}[v]$$

as a linear operator which admits a Riesz representation under the inner product for the first-step SMD estimation:

$$\langle u, v\rangle_{\rho_2} = \mathbb{E}\left[\frac{d\mathbb{E}[Y_1 - h_0(Y_2) \mid X]}{dh}[u]'\Sigma(X)^{-1}\frac{d\mathbb{E}[Y_1 - h_0(Y_2) \mid X]}{dh}[v]\right] = \mathbb{E}\left[\mathbb{E}[u \mid X]\Sigma(X)^{-1}\mathbb{E}[v \mid X]\right],$$

since the pathwise derivative is

$$\frac{d\mathbb{E}[Y_1 - h_0(Y_2) \mid X]}{dh}[v] = -\mathbb{E}[v \mid X].$$

Note that $\Sigma(X)$ is the scalar variance $\mathrm{Var}(Y_1 - h_0(Y_2)|X)$. Let $v^\star_{\rho_2}$ be the Riesz representer. Applying Chen and Liao (2015) we obtain the asymptotic influence function expansion:

$$\hat{\theta} - \theta_0 \approx \frac{1}{n}\sum_{i=1}^{n}\left[a(Y_{2i})\nabla_1 h_0(Y_{2i}) - \theta_0 - \Gamma(X_i)(Y_{1i} - h_0(Y_{2i})) + \frac{\mathbb{E}[v^\star_{\rho_2} \mid X]}{\Sigma(X)}(Y_1 - h_0(Y_2))\right], \quad (22)$$

which also verifies that $v^\star_{\rho_2} = v^\star_h$ are the same object.[24]

This alternative analysis allows us to estimate $v^\star_{\rho_2}$ directly, instead of estimating $w^\star$ as in the optimal weighed SMD case, since it shows that $v^\star_{\rho_2} = v^\star_h$ is in fact a Riesz representer on its own

[24] Assuming *completeness*: $\mathbb{E}[h(Y_2) \mid X] = 0 \iff h(Y_2) = 0$ a.s.

with respect to a different inner product. The definition of the Riesz representer is such that

$$\mathbb{E}[\mathbb{E}[v^\star_{\rho_2} \mid X]\Sigma(X)^{-1}\mathbb{E}[v \mid X]] = \mathbb{E}[a(Y_2)\nabla_1 v + \Gamma(X)v] \quad \|v^\star_{\rho_2}\|^2_{\rho_2} = \sup_v \frac{(\mathbb{E}[a(Y_2)\nabla_1 v + \Gamma(X)v])^2}{\mathbb{E}[\Sigma(X)^{-1}\mathbb{E}[v \mid X]^2]}.$$

Though this population version is difficult to characterize, it again can be approximated via linear sieve $\mathcal{V}_n = \{\nu(Y_2)'\gamma : \gamma\}$ (for instance one can think of $\nu(Y_2)$ as a power series or splines or Fourier series in Y_2)

$$\|v^\star_{\rho_2,n}\|^2_{\rho_2} = \sup_{v \in \mathcal{V}_n} \frac{(\mathbb{E}[a(Y_2)\nabla_1 v + \Gamma(X)v])^2}{\mathbb{E}[\Sigma(X)^{-1}\mathbb{E}[v \mid X]^2]}$$

then the sieve version of the Riesz representer is easy to compute. For completeness, the sieve Riesz representer is

$$v^\star_{\rho_2,n} = \nu(Y_2)'\mathbb{E}[\Sigma(X)^{-1}\mathbb{E}[\nu \mid X]\mathbb{E}[\nu \mid X]']^{-1}\mathbb{E}[a(Y_2)\nabla_1 \nu + \Gamma(X)\nu]$$

as a specialization of Chen and Liao (2015).

The root-n asymptotic normality now can be established by checking the sufficient conditions in Chen and Liao (2015).