Machine Learning Based Panel Data Models

Bingduo Yang* Wei Long† Zongwu Cai‡

January 15, 2024

Abstract

We examine nonparametric panel data regression models with fixed effects and cross-sectional dependence through a diverse collection of machine learning techniques. We add cross-sectional averages and time averages as regressors to the model to account for unobserved common factors and fixed effects respectively. Additionally, we utilize the debiased machine learning method by Chernozhukov et al. (2018) to estimate parametric coefficients followed by the nonparametric component. We comprehensively investigate three commonly used machine learning techniques - LASSO, random forests, and neural network - in finite samples. Simulation results demonstrate the effectiveness of our proposed method across different combinations of the number of cross-sectional units, time dimension sample size, and the number of regressors, irrespective of the presence of fixed effects and cross-sectional dependence. In the empirical part, we employ the proposed machine learning-based panel data model to estimate the total factor productivity (TFP) of public companies of Chinese mainland and find that the proposed machine learning methods are comparable to other competitive methods.

Keywords: Machine learning; panel data model; cross-sectional dependence; debiased machine learning

JEL classification: **C12, C22**

*School of Finance, Guangdong University of Finance and Economics, Guangzhou 510320, China.
†Department of Economics, Tulane University, New Orleans, LA 70118, United States.
‡Department of Economics, University of Kansas, Lawrence, KS, United States.

1 Introduction

Machine learning-based panel data models have garnered considerable attention in recent times and found extensive applications in empirical economics and finance, ranging from demand estimation (Bajari et al., 2015) to labor markets (Dube et al., 2020), audit quality (Yang et al., 2020), and empirical asset pricing (Gu et al., 2020; Leippold et al., 2022). Despite their increasing popularity, the unobservable nature of individual effects and common factors poses a significant challenge. Most researchers, as a result, gravitate towards pooled methods, inadvertently overlooking the joint consideration of these latent variables. Panel data models, known for their distinctive capability to mitigate endogeneity concerns by incorporating both individual effects and unobserved common factors, present a crucial aspect that should not be ignored. Consequently, employing a machine learning-based panel data model with a pooled dataset may yield biased estimators, particularly when individual effects, unobserved common factors, or both exhibit correlations with the regressors.

In this study, we aim to establish a comprehensive framework for panel data models by leveraging machine learning techniques. Our proposed method exhibits generality, as it accommodates a wide range of machine learning algorithms, including LASSO, random forests, boosted trees, deep neural networks, and ensembles or aggregated versions thereof. This stands in contrast to existing research, which often confines itself to specific machine learning tools for panel data models, such as sparse-group LASSO (Babii et al., 2022) and deep neural networks (Chronopoulos et al., 2023). Additionally, our approach explicitly addresses both fixed effects and cross-sectional dependence to mitigate endogeneity concerns. This is a difference from other works that either concentrate solely on pooled data (Gu et al., 2020; Leippold et al., 2022) or, at most, incorporate fixed effects as the predictors (Babii et al., 2022; Chronopoulos et al., 2023).

In the absence of a predetermined functional form, we employ diverse machine learning techniques to estimate a nonparametric panel data regression model encompassing both fixed effects and cross-sectional dependence. Our approach involves incorporating cross-sectional averages and time averages as regressors to respectively capture unobservable common factors and fixed effects, aligning with the methodology proposed by Huang (2013). To address concerns related to overfitting and regularization-induced bias arising from the machine learning techniques' plug-in nuisance function estimators, we adopt the

debiased machine learning method introduced by Chernozhukov et al. (2018). This entails initially estimating parametric coefficients, followed by the nonparametric component. Under the assumptions of a large number of cross-sectional units ($N \to \infty$), a substantial time dimension ($T \to \infty$), and a fixed number of regressors (d fixed), the estimators for parametric coefficients maintain unbiasedness, and the nonparametric estimators exhibit consistency.

We conduct a comprehensive investigation into the performance of three commonly utilized machine learning techniques - LASSO, random forests, and neural networks— in finite samples. In scenarios where the data generating process adheres to a sparse linear model, LASSO consistently demonstrates reasonably good in-sample estimation with favorable mean squared error and R^2, regardless of the presence of fixed effects or cross-sectional dependence. However, both random forests and neural networks tend to exhibit an overfitting issue in these settings. Conversely, when the data generating process takes a nonlinear form, random forests and neural networks outshine LASSO, yielding superior in-sample accuracy. An assessment of the three models' out-of-sample goodness-of-fit reveals that our proposed machine learning-based panel data model provides enhanced robustness and higher R^2 values across various scenarios. In summary, the simulation results underscore the effectiveness of our proposed method across diverse combinations of N, T, and d, irrespective of the presence of fixed effects and cross-sectional dependence.

In an empirical application, we employ the proposed machine learning-based panel data model to estimate total factor productivity (TFP) of public companies in the Chinese mainland. We respectively specify the production function as a nonparametric function and unobserved productivity the addition of fixed effect and unobserved factors. The empirical results demonstrates that, although our models do not utilize any proxies in the model specification, our three machine learning methods are comparable to other five popular existing methods including the pooled model, fixed effect model, and methods by Olley and Pakes (1996), Levinsohn and Petrin (2003) and Ackerberg et al. (2015).

Our contribution to the existing literature is threefold. Firstly, we enhance the non-parametric panel data literature by integrating various machine learning tools, thereby enabling the inclusion of a more extensive set of regressors. Prior researchers have predominantly relied on classical nonparametric methods to estimate unknown functions in panel data models with cross-sectional dependence. For instance, Su and Jin (2012) ad-

vocates the use of sieve estimators for nonparametric regression functions, while Huang (2013) employs kernel-based local linear regression for estimation. Both approaches incorporate cross-sectional averages to filter unobserved common factors, akin to the common correlated effects estimator (CCE) proposed by Pesaran (2006). Nevertheless, classical nonparametric methods are often constrained to one or two regressors due to the curse of dimensionality issue. For an in-depth exploration of nonparametric panel data models with cross-sectional dependence, readers are directed to survey papers such as Sun et al. (2015) and Xu et al. (2016).

Secondly, we contribute to the machine learning literature by introducing individual effects and unobserved common factors to simultaneously address issues of heterogeneity, cross-sectional dependence, and endogeneity. Our proposed method is versatile, as it accommodates various machine learning tools, in contrast to other studies that often focus on specific techniques in panel data models. For example, Babii et al. (2022) introduce a machine learning panel data regression approach for nowcasting price/earnings ratios, concentrating on fixed effects panel regressions with sparse-group LASSO (sg-LASSO) regularization. In a different vein, Chronopoulos et al. (2023) proposes a novel machine learning panel data estimator based on deep neural networks. However, these recent models are confined to either LASSO or neural networks and do not consider potential cross-sectional dependence. To the best of our knowledge, there is currently no existing literature on implementing generic machine learning techniques for panel data regressions that account for both fixed effects and cross-sectional dependence.

Third, we contribute to TFP estimation by introducing flexibility to the production function and mitigating endogeneity concerns through the incorporation of fixed effects and cross-sectional dependence, without reliance on additional proxy variables. Conventional TFP estimation methods, exemplified by Olley and Pakes (1996), Levinsohn and Petrin (2003), and Ackerberg et al. (2015), typically address endogeneity by introducing proxy variables into the control functions.

The remaining part of the paper is structured as follows. Section 2 introduces the proposed econometric model. Section 3 reports the finite-sample simulation results. Section 4 applies the proposed machine learning-based panel data model to estimate TFP. We conclude in Section 5. The technical proofs are provided in the supplementary appendix.

2 Econometric Model

We consider a nonparametric panel data regression model that encompasses both fixed effects and cross-sectional dependence ($N \to \infty$, $T \to \infty$, and d fixed). The model can be expressed as follows:

$$y_{it} = f(\mathbf{x}_{it}) + \alpha_i + u_{it}, \quad u_{it} = \gamma_i^\intercal \lambda_t + \varepsilon_{it}, \quad 1 \le i \le N, \quad 1 \le t \le T, \tag{1}$$

where y_{it} denotes the dependent variable for the i-th cross-sectional unit observed at time t. The function $f(\cdot)$, representing an unknown nonparametric function, is estimated using machine learning techniques such as LASSO, random forests, boosted trees, deep neural networks, or their ensembles. $\mathbf{x}_{it} = (x_{it,1}, \ldots, x_{it,d})^\intercal$ represents the d-dimensional regressors for the i-th cross-sectional unit at time t. The individual-specific fixed effects α_i could exhibit correlation with regressors $\mathbf{x}_{it}$. The error term u_{it} demonstrates a multifactor structure with λ_t as unobserved p-dimensional time-varying common factors accounting for cross-sectional dependence, and γ_i representing the corresponding p-dimensional factor loading for the i-th unit. In practical situations, regressor $\mathbf{x}_{it}$ and unobserved common factors λ_t could be correlated, potentially leading to endogeneity and inconsistent estimation. The individual-specific (idiosyncratic) error ε_{it} is a sequence of independent and identically distributed (i.i.d.) random variables with zero mean and finite variance, independent of $\mathbf{x}_{it}$, α_i, and λ_t for all i and t. To ensure identification, we assume $E(\alpha_i) = 0$ and $E(\lambda_t) = 0$.

Replacing $\mathbf{x}_{it}$ with $\mathbf{x}_{i(t-1)}$ transforms the model into a predictive panel data regression. When $\gamma_i = 0$ for all i, the model simplifies to a nonparametric panel data regression with fixed effects, or a random effects model if α_i is uncorrelated with $\mathbf{x}_{it}$. If the function $f(\cdot)$ is linear with respect to $\mathbf{x}_{it}$, the model aligns with the linear panel data model discussed by Pesaran (2006). Lastly, if the unknown function $f(\cdot)$ is estimated using sieve methods or kernel functions, the model becomes a special case studied by Su and Jin (2012) and Huang (2013).

To obtain the nonparametric estimator $\widehat{f}(\cdot)$, we employ cross-sectional averages, denoted as $\bar{y}_t = \frac{1}{N}\sum_{i=1}^{N} y_{it}$ and $\bar{\mathbf{x}}_t = \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_{it}$, following the approach by Pesaran (2006), to filter the unobserved common factors. Simultaneously, we utilize time averages, denoted as $\bar{y}_i = \frac{1}{T}\sum_{t=1}^{T} y_{it}$ and $\bar{\mathbf{x}}_i = \frac{1}{T}\sum_{t=1}^{T} \mathbf{x}_{it}$, following the methodology of Huang (2013),

to filter the fixed effects. The detailed proofs are provided in Appendix A. The underlying principle of this procedure is to substitute unobserved factors with observable data in the panel data model, thereby eliminating the effects of both unobserved common factors and fixed effects as N and T approach infinity. Notably, this procedure exhibits robustness concerning the number of unobserved common factors as N and T tend to infinity.

Define $\bar{y} = \frac{1}{N}\frac{1}{T}\sum_{i=1}^{N}\sum_{t=1}^{T} y_{it}$ and $\bar{\mathbf{x}} = \frac{1}{N}\frac{1}{T}\sum_{i=1}^{N}\sum_{t=1}^{T} \bar{\mathbf{x}}_{it}$, we rewrite the panel data model (1) as follows:

$$y_{it} = f(\mathbf{x}_{it}) + \beta^{\intercal}\mathbf{z}_{it} + e_{it}, \quad 1 \leq i \leq N, \quad 1 \leq t \leq T, \tag{2}$$

where $\mathbf{z}_{it} = (\bar{y}_t - \bar{y}, (\bar{\mathbf{x}}_t - \bar{\mathbf{x}})^{\intercal}, \bar{y}_i - \bar{y}, (\bar{\mathbf{x}}_i - \bar{\mathbf{x}})^{\intercal})^{\intercal}$, and $\beta = (\beta_1, \beta_2, \beta_3, \beta_4)^{\intercal}$ is the corresponding $(2d+2)$-dimensional coefficient vector. We use the demeaned version here for identification purpose. The error term e_{it}, which consists of both the idiosyncratic term ε_{it} and the approximated error from the Taylor expansion, is a mean zero residual term and not correlated with $\mathbf{x}_{it}$ and $\mathbf{z}_{it}$. When there are no unobserved common factors, we alternatively use $\mathbf{z}_{it} = (\bar{y}_i - \bar{y}, (\bar{\mathbf{x}}_i - \bar{\mathbf{x}})^{\intercal})^{\intercal}$.

Model (2) is a semiparametric partial linear model with a parametric component $\beta^{\intercal}\mathbf{z}_{it}$ and a nonparametric component $f(\mathbf{x}_{it})$. We take conditional expectation on both sides of the model and can obtain

$$E(y|\mathbf{x} = \mathbf{x}_{it}) = f(\mathbf{x}_{it}) + \beta^{\intercal}E(\mathbf{z}|\mathbf{x} = \mathbf{x}_{it}).$$

Define $g(\mathbf{x}_{it}) = E(y|\mathbf{x} = \mathbf{x}_{it})$ and $m(\mathbf{x}_{it}) = E(\mathbf{z}|\mathbf{x} = \mathbf{x}_{it})$. Subtracting this equation from (2), we have

$$y_{it} - g(\mathbf{x}_{it}) = \beta^{\intercal}(\mathbf{z}_{it} - m(\mathbf{x}_{it})) + e_{it}.$$

For machine learning-based estimators such as $\widehat{g}(\mathbf{x}_{it})$ and $\widehat{m}(\mathbf{x}_{it})$, regularization techniques like model selection or parameter shrinkage become necessary when the number of parameters is relatively large compared to the available sample size. However, while these regularization methods help reduce the variance of the estimators, they often introduce significant biases. To mitigate the regularization-induced bias in the plug-in nonparametric estimators $\widehat{g}(\mathbf{x}_{it})$ and $\widehat{m}(\mathbf{x}_{it})$, we utilize the debiased machine learning approach introduced by Chernozhukov et al. (2018) to estimate the parameter β. Specifically, we

define a score equation

$$\psi(W_{it};\beta,\eta) = \big(y_{it} - g(\mathbf{x}_{it}) - \beta^{\intercal}(\mathbf{z}_{it} - m(\mathbf{x}_{it}))\big)(\mathbf{z}_{it} - m(\mathbf{x}_{it})), \tag{3}$$

where the data set $W_{it} = (y_{it}, \mathbf{x}_{it})$, the coefficients β, and a vector of the nonparametric nuisance functions $\eta = (g(\cdot), m(\cdot))$. The attractive point of above score equation (3) is that it obeys the Neyman orthogonality condition (Neyman, 1959; Neyman, 1979)

$$\partial_{\eta} E\left[\psi\left(W_{it};\beta,\eta\right)\right]\big|_{\eta=\eta_0} = 0,$$

where the derivative ∂_{η} denotes the pathwise (Gateaux) derivative operator.The aforementioned orthogonality condition indicates that the score equation remains unaffected by minor errors in estimating the nuisance functions $\widehat{\eta}$ around their actual values η_0, provided that these estimators are of high quality in terms of machine learning techniques. By construction, we can obtain the estimator $\widehat{\beta}$ by solving

$$\sum_{t=1}^{T}\sum_{i=1}^{N} \psi\left(W_{it};\widehat{\beta},\widehat{\eta}\right) = 0, \tag{4}$$

where $\widehat{\eta}$ is a plug-in nonparametric estimator and can be estimated by a variety of machine learning methods such as boosting, random forests, ensemble, and hybrid machine learning methods.

Additionally, we implement the sample splitting strategy proposed by Chernozhukov et al. (2018) to address the problem of overfitting that often arises in the estimation of nuisance functions $\widehat{\eta}$ using machine learning approaches. Specifically, we divide the time indices $\{1,\cdots,T\}$ evenly into K-fold partition $\{P_k\}_{k=1}^{K}$, with the size of each fold P_k being $N \times T_0 = N \times [T/K]$ and $[\cdot]$ the integer operator. For each $k \in \{1,\cdots,K\}$, we leave the kth fold out to obtain a high-quality machine learning estimator $\widehat{\eta}^{(-k)} = \widehat{\eta}^{(-k)}(W_{it} : t \notin P_k)$. And we can obtain the estimator $\widehat{\beta}$ as the solution to the equation[1]

$$\sum_{i=1}^{N}\sum_{k=1}^{K}\sum_{t\in P_k} \psi\left(W_{it};\widehat{\beta},\widehat{\eta}^{(-k)}\right) = 0. \tag{5}$$

[1]An alternative estimator is to obtain the estimator $\widehat{\beta}$ via aggregation $\widehat{\beta} = \frac{1}{K}\sum_{k=1}^{K}\tilde{\beta}^{(k)}$ with each estimator $\tilde{\beta}^{(k)}$ being the solution to the equation $\frac{1}{NT_0}\sum_{t\in P_k}\psi\left(W_{it};\tilde{\beta}^{(k)},\widehat{\eta}^{(-k)}\right) = 0$. However, this estimator is not as stable as that obtained by (5). See Bach et al. (2022) .

As the score $\psi\left(W_{it};\widehat{\beta},\widehat{\eta}^{(-k)}\right)$ can be written as a linear form $\psi\left(W_{it};\widehat{\beta},\widehat{\eta}^{(-k)}\right) = -\left(\mathbf{z}_{it}-\widehat{m}^{(-k)}(\mathbf{x}_{it})\right)\left(\mathbf{z}_{it}-\widehat{m}^{(-k)}(\mathbf{x}_{it})\right)^{\intercal}\widehat{\beta}+\left(\mathbf{z}_{it}-\widehat{m}^{(-k)}(\mathbf{x}_{it})\right)\left(y_{it}-\widehat{g}^{(-k)}(\mathbf{x}_{it})\right)$, we can find the estimator as

$$\widehat{\beta} = \left[\sum_{i=1}^{N}\sum_{k=1}^{K}\sum_{t\in P_k}\left(\mathbf{z}_{it}-\widehat{m}^{(-k)}(\mathbf{x}_{it})\right)\left(\mathbf{z}_{it}-\widehat{m}^{(-k)}(\mathbf{x}_{it})\right)^{\intercal}\right]^{-1}\times \sum_{i=1}^{N}\sum_{k=1}^{K}\sum_{t\in P_k}\left(\mathbf{z}_{it}-\widehat{m}^{(-k)}(\mathbf{x}_{it})\right)\left(y_{it}-\widehat{g}^{(-k)}(\mathbf{x}_{it})\right) \quad (6)$$

where $\widehat{g}^{(-k)}(\mathbf{x}_{it})$ is the predicted value at the sample point $\mathbf{x}_{it}$ by performing machine learning-based regression of y_{it} on $\mathbf{x}_{it}$ without the kth fold P_k. And $\widehat{m}^{(-k)}(\mathbf{x}_{it})$ represents the predicted values at the sample point $\mathbf{x}_{it}$ by performing machine learning-based regression of $\mathbf{z}_{it}$ on $\mathbf{x}_{it}$ without the kth fold P_k.

We employ the following regularity conditions to derive the asymptotic limit of the estimators.

(C1) Assume the data $\{y_{it},\mathbf{x}_{it};i=1,\cdots,N,t=1,\cdots,T\}$ is a stationary process. The number of cross-sectional units $N\to\infty$, the time dimension $T\to\infty$, and the numbers of regressors d and unobserved common factors p are fixed.

(C2) The individual-specific (idiosyncratic) error ε_{it} is a sequence of independent and identically distributed (i.i.d.) random variables with zero mean and finite variance, independent of $\mathbf{x}_{it}$, α_i, and λ_t for all i and t. The unobserved p-dimensional time-varying common factors λ_t are covariance stationary and independent of the individual-specific error ε_{it}, but can be correlated with the regressors $\mathbf{x}_{it}$. The unobserved p-dimensional factor loadings γ_i are i.i.d with finite mean and finite variance, and independent of $\mathbf{x}_{it}$, the common factors λ_t, and the individual-specific error ε_{it}. We further assume $E(\alpha_i)=0$ and $E(\lambda_t)=0$ for the identification purpose. And $\bar{\gamma}\bar{\gamma}^{\intercal}$ has the full rank so that it is invertible.

(C3) The nuisance parameters $\hat{\eta}$ are high-quality machine learning estimators such that $\hat{\eta}-\eta=o_p((NT)^{-1/4})$.

(C4) Define the neighborhood sample points set $\mathcal{N}(\mathbf{x},\nu)=\{(i,t)|\|\mathbf{x}_{it}-\mathbf{x}\|_2<\nu\}$ with $\|\cdot\|_2$ being the Euclidean norm and ν some positive constant. As $N\to\infty,T\to\infty,\nu\to 0$ and the number of sample points in the set $\mathcal{N}(\mathbf{x},\nu)$, i.e. $|\mathcal{N}(\mathbf{x},\nu)|\to\infty$. The functions

$f(\cdot)$, $g(\cdot)$ and $m(\cdot)$ are continuous, bounded and has bounded second order derivatives. And assume $E\left\{\mathbf{z}\middle|\mathbf{x}\right\} = O_p(1)$.

Remark 1. *The stationarity condition in (C1) is critical in proof for equation (2). We assume $N \to \infty$ and $T \to \infty$ so that the cross-sectional averages and time averages are valid instruments for unobserved common factors and fixed effects, respectively. Condition (C2) is classical in panel data model with cross-sectional dependence. Conditions (C3) and (C4) are for Theorems 1 and 2. The number of regressors d can be large as long as condition (C3) holds. The condition $|\mathcal{N}(\boldsymbol{x}, \nu)| \to \infty$ in (C4) implies that there are enough sample points in the neighborhood of given point $\boldsymbol{x}$ as $N \to \infty$ and $T \to \infty$.*

Theorem 1. *Suppose that conditions (C1) $\sim$ (C4) hold, this double/debiased machine learning estimator $\hat{\beta}$ is $\sqrt{NT}$ consistent, and the estimation error $\sqrt{NT}\left(\widehat{\beta} - \beta\right)$ is approximately a normal distribution as:*

$$\sqrt{NT}\left(\widehat{\beta} - \beta\right) \sim N\left(0, \Sigma^{-1} V \Sigma^{-1}\right),$$

where $\Sigma = E\left[\left(\boldsymbol{z}_{it} - m(\boldsymbol{x}_{it})\right)\left(\boldsymbol{z}_{it} - m(\boldsymbol{x}_{it})\right)^{\intercal}\right]$ and $V = E\left[\psi\left(W_{it}; \beta, \eta\right)\psi\left(W_{it}; \beta, \eta\right)^{\intercal}\right]$ with $\eta = (g(\cdot), m(\cdot))$.

Remark 2. *Theorem 1 is a special case in Chernozhukov et al. (2018), so we skip the detailed proof here.*

In real application, we choose $K = 5$ as suggested by Bach et al. (2022). By construction, as long as the nuisance parameters $\widehat{\eta}$ are high-quality machine learning estimators in the senses that in the worst cases, they are estimated at the rate $o_p((NT)^{-1/4})$ when the number of regressors d is fixed, this double/debiased machine learning estimator $\widehat{\beta}$ is $\sqrt{NT}$ consistent, and the estimation error $\sqrt{NT}\left(\widehat{\beta} - \beta\right)$ is approximately a normal distribution with zero mean and finite variance. On the contrary, in the absence of both the Neyman orthogonality condition for the score function and the sample-splitting procedure, the estimator is likely to exhibit substantial bias.

Finally, given $\widehat{\beta}$ from the equation (6), we compute the estimators $\widehat{f}(\mathbf{x}_{it})$ by performing machine learning-based regression of $y_{it} - \widehat{\beta}^{\intercal}\mathbf{z}_{it}$ on $\mathbf{x}_{it}$. And the fitted value for y_{it} is $\widehat{y}_{it} = \widehat{f}(\mathbf{x}_{it}) + \widehat{\beta}^{\intercal}\mathbf{z}_{it}$.

Theorem 2. *Suppose that conditions (C1)∼(C4) hold and* $\widehat{\beta} - \beta = O_p((NT)^{-1/2})$. *The machine learning based nonparametric estimator* $\widehat{f}(\boldsymbol{x})$ *is consistent in the sense that* $\widehat{f}(\boldsymbol{x}) \xrightarrow{p} f(\boldsymbol{x})$ *as* $N \to \infty$ *and* $T \to \infty$.

Remark 3. *We don't present the asymptotic distribution of the nonparametric estimator* $\widehat{f}(\boldsymbol{x})$ *as many machine learning estimators such as random forest and neural network neither.*

When implementing out of sample forecasting, the fitted value for $y_{i(t+1)}$ at given point $\mathbf{x}_{i(t+1)}$ is $\widehat{y}_{i(t+1)} = \widehat{f}\left(\mathbf{x}_{i(t+1)}\right) + \widehat{\beta}_1(\bar{y}_{t+1} - \bar{y}) + \widehat{\beta}_2^{\intercal}(\bar{\mathbf{x}}_{t+1} - \bar{\mathbf{x}}) + \widehat{\beta}_3(\bar{y}_i - \bar{y}) + \widehat{\beta}_4^{\intercal}(\bar{\mathbf{x}}_i - \bar{\mathbf{x}})$. As $\bar{y}_{t+1}$ are not observed, we estimate $\bar{y}_{t+1}$ by conducting linear regression or machine learning-based regression $\bar{y}_s$ on $\bar{\mathbf{x}}_s (s \leq t)$, and evaluate at point $\bar{\mathbf{x}}_{t+1}$.

3 Numerical Studies

In this section, we assess the finite sample performance of our method through Monte Carlo simulations. The data generating process follows

$$y_{it} = c_1 \times \alpha_i + c_2 \times \gamma_i \lambda_t + f(\mathbf{x}_{it}) + \varepsilon_{it}, \tag{7}$$

where $i = 1, 2, \cdots, N$ with $N \in \{10, 20\}$, and $t = 1, 2, \cdots, T$ with $T \in \{100, 200, 400\}$. α_i measures the magnitude of the fixed effects of unit i, and is distributed to $N(0, 1)$. λ_t denotes a sequence of time-varying common factors which are also distributed to $N(0, 1)$, and $\gamma_i = 0.5$ denotes the associated factor loading of unit i. We use demeaned version of α_i and λ_t, so that $\sum_{i=1}^{N} \alpha_i = 0$ and $\sum_{t=1}^{T} \lambda_t = 0$ hold. The idiosyncratic error term ε_{it} is assumed to be independent across i and t, and distributed to $N(0, 1)$. $\mathbf{x}_{it}$ denotes a d-dimensional vector which represents the potential predictors with $d \in \{5, 10\}$ throughout this section. For $k = 1, \cdots, d$, to allow $\mathbf{x}_{it,k}$ to be correlated with the i-th entity's fixed effect and the time-varying factors at time t, we define $\mathbf{x}_{it,k} = 0.3\alpha_i + 0.3\lambda_t + \eta_{it,k}$, where $\eta_{it,k} \sim N(0, 1)$.

In model (7), the function $f(\cdot)$ is assumed to have the following two specifications:

TYPE 1: $f(\mathbf{x}_{it}) = 0.2x_{it,1} + 0.2x_{it,2} + 0.2x_{it,3}$;

TYPE 2: $f(\mathbf{x}_{it}) = 0.4x_{it,1} + 0.3x_{it,1}x_{it,2} + 0.12\text{sgn}(x_{it,3})$, where $\text{sgn}(\cdot)$ denotes the sign function.

Here, Type 1 represents a sparse and simple linear model, while Type 2 contains both nonlinear and interacted terms. Additionally, regardless of the dimension of $\mathbf{x}_{it}$ (i.e., $d = 5$ or $d = 10$), only the first three predictors contribute to the outcome variable y_{it}. c_1 and c_2 can be regarded as switches that determine the presence of individual fixed effects and cross-sectional dependence: (i) when $c_1 = c_2 = 0$, neither individual fixed effects nor cross-sectional dependence exist, (ii) when $c_1 = 1$ and $c_2 = 0$, only individual fixed effects exist while cross-sectional dependence does not exist, and (iii) when $c_1 = c_2 = 1$, both individual fixed effects (FE) and cross-sectional dependence exist.

For each Monte Carlo experiment, we divide the generated series into three consecutive subsamples – training ($\mathcal{T}_1$), validation ($\mathcal{T}_2$), and testing ($\mathcal{T}_3$) – with each subsample respectively accounts for the 30%, 20%, and 50% of the whole sample. Specifically, we estimate each of the two aforementioned types in the training set using several widely used machine learning methods, then choose tuning parameters for each method in the validation set, and calculate the prediction errors in the testing set.

In this study, to conserve space, we do not attempt to exhaustively consider all different machine learning techniques but simply adopt three widely used ones: LASSO, random forests (RF), and neural network (NN) with three hidden layers of 32, 16, and 8 neurons, respectively, as Gu et al. (2020) find that this architecture exhibits better performance in their monthly return setting.[2] We first assess the performance of these methods in estimating $f(\mathbf{x}_{it})$. Specifically, the estimation accuracy is proxied by the mean squared errors defined as $MSE(f) = \frac{1}{N}\frac{1}{T_1}\sum_{i=1}^{N}\sum_{t=1}^{T_1}\left(\hat{f}(\mathbf{x}_{it}) - f(\mathbf{x}_{it})\right)^2$, where $(i,t) \in \mathcal{T}_1$. Table 1 presents the results across a variety of combinations of N, T, d and (c_1, c_2) when $f(\cdot)$ adopts the Type 1 specification. In panel A when the true specification contains neither cross-sectional dependence nor individual FE (i.e., $c_1 = c_2 = 0$), the machine learning method by pooling all data (Pooled) yields the lowest MSEs in LASSO, and increasing N or T can effectively lower the MSEs for all three methods. That is, pooling the whole data and applying a single machine learning model will yield satisfactory results when individual FE and cross-sectional dependence do not exist. Such a pattern in more pronounced for LASSO, which also yields the the lowest MSEs compared with RF and NN. This result should not be surprising because $f(\cdot)$ in Type 1 is sparse and linear in the predictors. On the other hand, the more advanced algorithms such as RF and NN tend to

[2]We also examine NNs with 1, 2, 4, and 5 hidden layers, as defined in Gu et al. (2020), and find quite similar outcomes. The results are available upon request.

overfit, yielding poorer in-sample performance. When individual FE and cross-sectional dependence are added to the true specification, the performance of pooled regression becomes remarkably deteriorated, while the machine learning based panel data model with individual FE or both cross-sectional dependence and FE (CR + FE) yields the lowest MSEs, as displayed by panels B and C, respectively. This finding indicates that the pooled regression method can generate large estimation error when individual FE or cross-sectional dependence indeed presents. Additionally, in each panel, increasing the number of predictors from $d = 5$ to $d = 10$ will notably increase the MSEs for all three methods. This should also be expected given the increased complexity caused by the higher dimension. When $f(\dot{c})$ adopts the Type 2 specification which is nonlinear, LASSO becomes dominated by RF and NN regardless of the combinations of c_1 and c_2 and the presence of individual FE and cross-sectional dependence, as displayed by Table 2. When $d = 5$, RF performs slightly better than NN when $N = 10$ and T is small. However, when the dimension increases to 10, NN tends to yield lower estimation errors than RF. Overall, the proposed machine learning based panel data model still exhibits robust performance regardless of the presence of individual FE and cross-sectional dependence.

[Insert Tables 1 and 2 About Here.]

Next, we report the averages of both in-sample (IS) and out-of-sample (OOS) R^2s for each type and each method. Here, we follow Gu et al. (2020) and define $R^2_{IS} = 1 - \frac{\sum_{(i,t)\in\mathcal{T}_1}(\hat{y}_{it}-y_{it})^2}{\sum_{(i,t)\in\mathcal{T}_1} y^2_{it}}$ and $R^2_{OOS} = 1 - \frac{\sum_{(i,t)\in\mathcal{T}_3}(\hat{y}_{it}-y_{it})^2}{\sum_{(i,t)\in\mathcal{T}_3} y^2_{it}}$, where $\hat{y}_{it} = \hat{f}(\mathbf{x}_{it}) + \hat{\beta}_1(\bar{y}_t - \bar{y}) + \hat{\beta}_2^{\intercal}(\bar{\mathbf{x}}_t - \bar{\mathbf{x}}) + \hat{\beta}_3(\bar{y}_i - \bar{y}) + \hat{\beta}_4^{\intercal}(\bar{\mathbf{x}}_i - \bar{\mathbf{x}})$ with $\hat{f}(\cdot)$, $\hat{\beta}_1$, $\hat{\beta}_2$, $\hat{\beta}_3$ and $\hat{\beta}_4$ estimated by the training set. As $\bar{y}_t$ are not observed, we estimate $\bar{y}_t$ by regressing $\bar{y}_s$ on $\bar{\mathbf{x}}_s, (i,s) \in \mathcal{T}_1$, and then evaluate at the point $\bar{\mathbf{x}}_t$, where $(i,t) \in \mathcal{T}_3$.

Tables 3 and 4 document the comparison of IS R^2s under Types 1 and 2, respectively. For all three models, the IS R^2s tend to increase as d increases from 5 to 10, regardless of the combination of N, T and (c_1, c_2). A comparison of the IS R^2s of the three competing models indicate that RF yields the best IS performance, as its IS R^2s are the highest across all different combinations. Particularly, regardless of the presence of individual FE or cross-sectional dependence, the RF-based panel data model with both individual FE and cross-sectional dependence (i.e., FE + CR) provides robust and highest IS R^2s when compared with the other two specifications (i.e., Pooled and FE).

Table 5 further compares the OOS performance of the three methods under Type 1. In panel A, when neither individual FE nor cross-sectional dependence exists, pooled regression yields the highest OOS R^2s. Under panels B and C, when individual FE and cross-section dependence are added, Pooled is outperformed by FE and CR + FE methods. Across the three panels, the LASSO-based model exhibits the best performance, which is not surprising given the linearity of Type 1. We can observe similar patterns in Table 6, although the LASSO-based models become outperformed by the RF- and NN-based models due to the complexity and nonlinearity in Type 2.

[Insert Tables 3 - 6 About Here.]

By tuning the value of c_1 and c_2, we could further visualize the comparison of performance. For example, in the first scenario, we fix $c_2 = 0$ and then allow $c_1 \in \{0, 0.1, 0.2, \cdots, 0.9, 1\}$ so that the cross-section dependence does not exist, while the magnitude of individual FE gradually increases. Results in Figure 1(a) indicate two observations. First, for all three machine learning methods, pooled regression yields reasonably good results when $c_1 = 0$, but its MSEs increase substantially when c_1 deviates from zero. That is, when individual FE indeed presents, the fitting performance of pooled regression becomes severely deteriorated. Second, if individual FE is effectively addressed, the proposed methods yield better fitting performance as the corresponding MSEs are not only smaller but also stable to the magnitude of FE, as their MSE paths are quite flat along the value of c_1. We can observe similar results in Figure 2(a) where $f(\cdot)$ is nonlinear. For the remaining panels in Figures 1 and 2, we consider other three scenarios of c_1 and c_2, and find that the proposed panel data model with both individual FE and cross-sectional dependence always outperforms the other two methods, as its MSEs are lower and robust to c_1 and c_2. Figures 3 - 6 display the comparison of IS and OOS R^2s under Types 1 and 2, and show that the proposed panel data model with both individual FE and cross-sectional dependence exhibits higher and robust R^2 across the four scenarios.

[Insert Figures 1 and 6 About Here.]

4 Estimation of Total Factor Productivity

In this empirical illustration, we apply the proposed machine learning-based panel data model to estimate the total factor productivity (TFP) of public companies in mainland China. The estimation of TFP is a critical aspect of economic growth with significant policy implications. Traditionally, a firm's TFP is estimated through the classical Cobb-Douglas regression (Cobb and Douglas, 1928):

$$y_{it} = \gamma_0 + \gamma_L \ln L_{it} + \gamma_K \ln K_{it} + \omega_{it} + \varepsilon_{it}, \quad 1 \leq i \leq N, \quad 1 \leq t \leq T, \tag{8}$$

where y_{it} represents the logarithm of the value-added of firm i in year t. Two crucial inputs for value-added, labor ($\ln L$) and capital ($\ln K$), serve as regressors. The coefficients γ_L and γ_K denote the elasticities of labor and capital, respectively. The term ω_{it} represents the unobserved productivity or technical efficiency, while ε_{it} is an idiosyncratic output shock following a white noise distribution. The TFP of firm i can then be estimated as $\widehat{TFP}_i = y_{it} - \hat{\gamma}L \ln L_{it} - \hat{\gamma}K \ln K_{it}$.

In practice, two issues arise when estimating TFP. The first is about the functional form of the production function. The Cobb-Douglas specification in (8) could be potentially stringent as it assumes a constant elasticity of the value-added. An alternative specification is the translog production function, which does not require the assumption of smooth substitution between production factors, see Gandhi et al. (2020) and De Loecker and Warzynski (2012). The second issue is the endogeneity of input choice arising from the the positive correlation between the observed input levels and the unobserved productivity shocks which leads to biased estimation for the production function. To fix the second issue, over the past two decades researchers have tried to alleviate the concern of endogeneity by specifying $\omega_{it} = h(\cdot)$ with $h(\cdot)$ being some control functions. Olley and Pakes (1996) were the first to propose a consistent two-step estimation procedure for (8). They exploit firm investment level $\ln I$ as a proxy variable in the control function. As many firms' investment levels are zeros, Levinsohn and Petrin (2003) use intermediate input level $\ln M$ as a proxy variable in the control function. These two studies take $\ln K$ as endogenous variable, while Ackerberg et al. (2015) additionally take $\ln L$ as endogenous and employ the lagged terms $\ln L_{i,t-1}$ and $\ln M_{i,t-1}$ as instrument variables to conduct the GMM estimation.

In this analysis, we specify the production function as a nonparametric function and estimate it via three popular machine learning methods: LASSO, random forests, and neural networks. Meanwhile, we assume the unobserved productivity ω_{it} is separable and can be decomposed as $\omega_{it} = \alpha_i + \gamma_i^\intercal \lambda_t$. By construction, the endogeneity of the unobserved productivity ω_{it} can be characterized by either α_i, or λ_t, or both. Therefore, we can estimate the production function via the proposed machine learning-based panel data model as

$$y_{it} = f(\ln L_{it}, \ln K_{it}) + \alpha_i + \gamma_i^\intercal \lambda_t + \varepsilon_{it}, \quad 1 \leq i \leq N, \quad 1 \leq t \leq T. \tag{9}$$

The extensive simulations in the previous section have already demonstrated the effectiveness of the proposed estimation procedure. A distinctive point of our setting is that we do not utilize any additional proxies or instrument variables. We present the summary of the above model specifications along with both pooled panel and fixed effects panel in Table 7.

We collect all non-financial companies in Chinese mainland that are publicly traded on the A-share market from 2001 to 2020. All data are from the China Stock Market & Accounting Research (CSMAR), developed by GTA Information Technology, one of the providers of Chinese data. After excluding firms with missing data, our final sample consists of 459 firms across 20 years and total of 9180 observations.

To assess the performance of the proposed methods in comparison to five existing models, we employ investment ($\ln I$) and material input ($\ln M$) as two instruments for a firm's unobserved productivity. Subsequently, we compute the correlations between these instruments and the residuals obtained from the three machine learning-based panel data models and five existing methods. Ideally, in a correctly specified model, the correlation between the two instruments (either $\ln I$ or $\ln M$) and the regression residuals ($\hat{\varepsilon}_{it}$) from the eight models should be close to zero.

Table 8 presents the estimated correlation coefficients. For the method following Olley and Pakes (1996), we exclude the correlation between $\ln I$ and $\hat{\varepsilon}_{it}$, as $\ln I$ is already utilized in the control function $h(\cdot)$. Similar adjustments are made when implementing the methods of Levinsohn and Petrin (2003) and Ackerberg et al. (2015) with $\ln L$ as an instrument. The results in Table 8 indicate that the proposed machine learning-based methods outperform the pooled model, fixed effect model, and methods by Olley and

Pakes (1996) and Ackerberg et al. (2015) when $\ln M$ is used. Furthermore, the machine learning-based methods demonstrate performance comparable to Levinsohn and Petrin (2003) and Ackerberg et al. (2015) when $\ln I$ serves as an instrument, as our models do not incorporate any proxies in the model specification.

5 Conclusion

The primary task of this study is to introduce a comprehensive framework for constructing machine learning-based panel data models. Following the methodology outlined in Huang (2013), we utilize cross-sectional averages and time averages as instrumental variables to capture unobserved common factors and fixed effects, respectively. The resultant nonparametric panel data model, accommodating fixed effects and cross-sectional dependence, is formulated as a partial linear semiparametric panel data model. To ensure the robustness of our estimation procedure in the presence of nuisance parameters estimated using diverse machine learning methods, we adopt the double/debiased machine learning technique developed by Chernozhukov et al. (2018). Throughout this paper, we operate under the assumption of a substantial number of cross-sectional units (N), a large time dimension (T), and a fixed number of regressors (d). Consequently, we directly apply the theory outlined in Chernozhukov et al. (2018). Extensive simulations corroborate the estimation accuracy of our method in scenarios where both fixed effects and cross-sectional dependence are present. When applying the method to estimate the total factor productivity (TFP) of public companies in mainland China, we find that the proposed machine learning-based methods are comparable to other prevalent methods in the literature. Additionally, we acknowledge the challenge posed by high-dimensional datasets (i.e., large d), where both the number of parametric coefficients and nuisance parameters becomes substantial. As of our current knowledge, there exists no established theory to comprehensively address this issue, and we recognize it as an area for future research.

Acknowledgments

Yang, B.'s research was partly supported by NSFC (72173140,71991474), Guangdong Basic and Applied Basic Research Foundation (2022A1515011793). Long's research was

partly supported by the Seed Grant of Murphy Institute and SLA Faculty Research Award at Tulane University.

References

Ackerberg, D. A., Caves, K., and Frazer, G. (2015). Identification properties of recent production function estimators. *Econometrica*, 83:2411–2451.

Babii, A., Ghysels, E., and Striaukas, J. (2022). Machine learning time series regressions with an application to nowcasting. *Journal of Business & Economic Statistics*, 40(3):1094–1106.

Bach, P., Chernozhukov, V., Kurz, M. S., and Spindler, M. (2022). Doubleml: an object-oriented implementation of double machine learning in python. *The Journal of Machine Learning Research*, 23(1):2469–2474.

Bajari, P., Nekipelov, D., Ryan, S. P., and Yang., M. (2015). Machine learning methods for demand estimation. *American Economic Review*, 105(5):481–1485.

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. *The Econometrics Journal*, 21(1):C1–C68.

Chronopoulos, I., Chrysikou, K., Kapetanios, G., Mitchell, J., and Raftapostolos, A. (2023). Deep neural network estimation in panel data models. *arXiv preprint arXiv:2305.19921*.

Cobb, C. W. and Douglas, P. H. (1928). A theory of production. *The American Economic Review*, 18(1):139–165.

De Loecker, J. and Warzynski, F. (2012). Markups and firm-level export status. *American Economic Review*, 102(6):2437–71.

Dube, A., Jeff, J., Suresh, N., and Suri, S. (2020). Monopsony in online labor markets. *American Economic Review: Insights*, 2(1):33–46.

Gandhi, A., Navarro, S., and Rivers, D. (2020). On the identification of gross output production functions. *Journal of Political Economy*, 128(8):2973–3016.

Gu, S., Kelly, B., and Xiu, D. (2020). Empirical asset pricing via machine learning. *The Review of Financial Studies*, 33(5):2223–2273.

Huang, X. (2013). Nonparametric estimation in large panels with cross-sectional dependence. *Econometric Reviews*, 32(5-6):754–777.

Leippold, M., Wang, Q., and Zhou, W. (2022). Machine learning in the chinese stock market. *Journal of Financial Economics*, 145(2, Part A):64–82.

Levinsohn, J. and Petrin, A. (2003). Estimating production functions using inputs to control for unobservables. *The Review of Economic Studies*, 70(2):317–341.

Olley, G. S. and Pakes, A. (1996). The dynamics of productivity in the telecommunications equipment industry. *Econometrica*, 64(6):1263–1297.

Pesaran, M. H. (2006). Estimation and inference in large heterogeneous panels with a multifactor error structure. *Econometrica*, 74(4):967–1012.

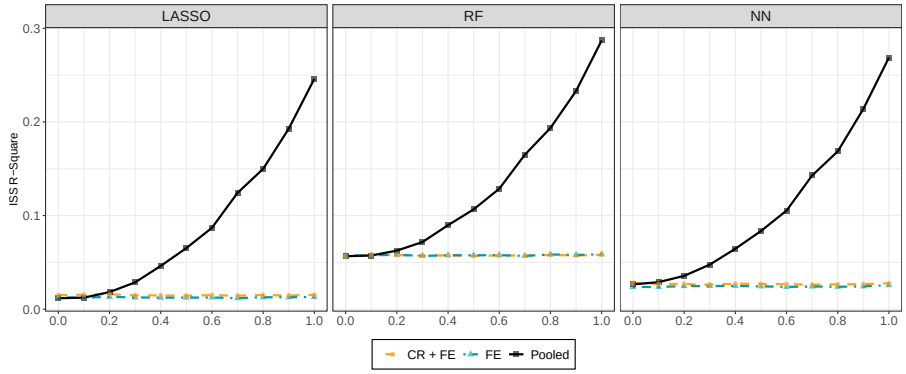
Su, L. and Jin, S. (2012). Sieve estimation of panel data models with cross section dependence. *Journal of Econometrics*, 169(1):34–47.

Sun, Y., Zhang, Y. Y., and Li, Q. (2015). Nonparametric Panel Data Regression Models. In *The Oxford Handbook of Panel Data*. Oxford University Press.

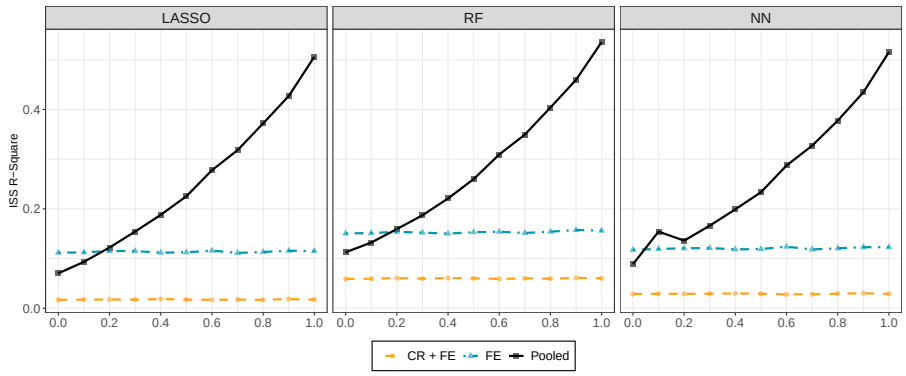
Xu, Q.-H., Cai, Z.-W., and Fang, Y. (2016). Panel data models with cross-sectional dependence: a selective review. *Applied Mathematics-A Journal of Chinese Universities*, 31(2):127–147.

Yang, J.-C., Chuang, H.-C., and Kuan, C.-M. (2020). Double machine learning with gradient boosting and its application to the big n audit quality effect. *Journal of Econometrics*, 216(1):268–283.

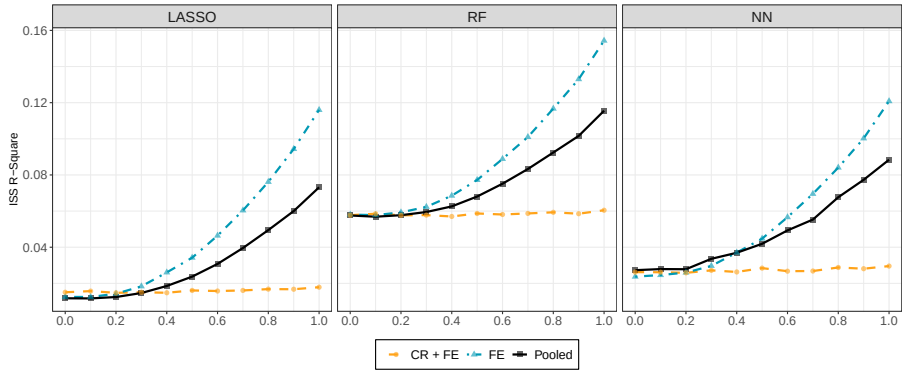
Figure 1: MSE under Type 1



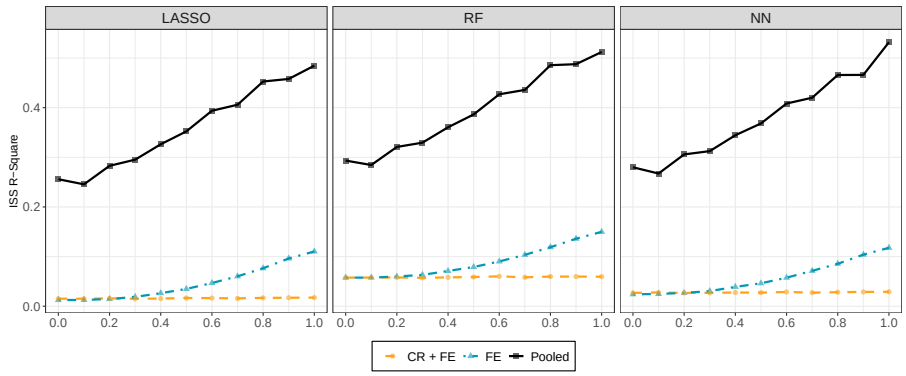
(a) $c_1 \in \{0, 0.1, 0.2, ..., 1\}, c_2 = 0$



(b) $c_1 \in \{0, 0.1, 0.2, ..., 1\}, c_2 = 1$



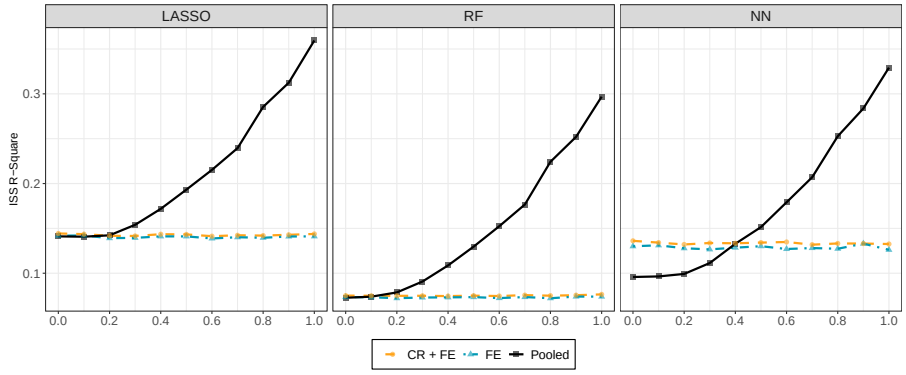
(c) $c_1 = 0, c_2 \in \{0, 0.1, 0.2, ..., 1\}$



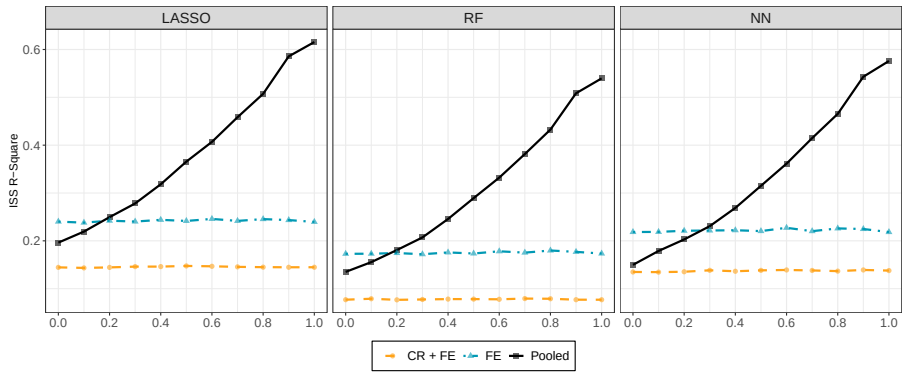
(d) $c_1 = 1, c_2 \in \{0, 0.1, 0.2, ..., 1\}$

Notes. This figure presents the average mean squared errors over 1,000 Monte Carlo repetitions for $f(\mathbf{x}_{it}) = 0.2x_{it,1} + 0.2x_{it,2} + 0.2x_{it,3}$ using LASSO, random forests, and neural networks with three hidden layers of 32, 16, and 8 neurons, respectively. The number of predictors $d = 5$, the number of entities $N = 10$, and the length of sample size $T = 100$. "Pooled" stands for using the machine learning method by pooling all data. "FE" stands for using the proposed machine learning based panel data model with fixed effects. "CR + FE" stands for using the proposed machine learning panel data model with both fixed effects and cross-sectional dependence.

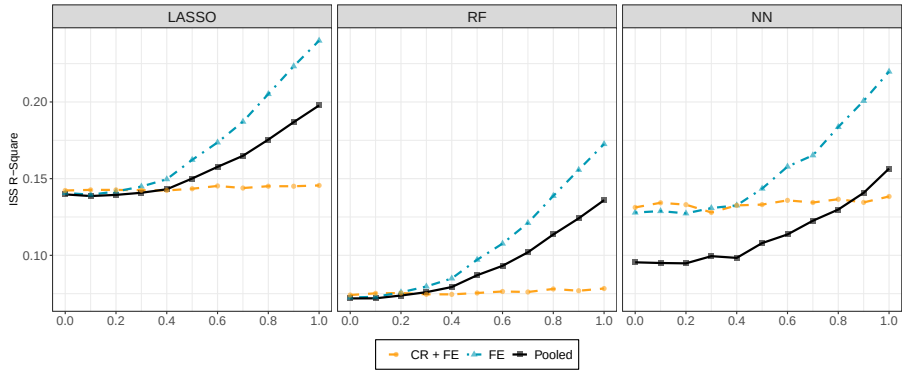
Figure 2: MSE under Type 2



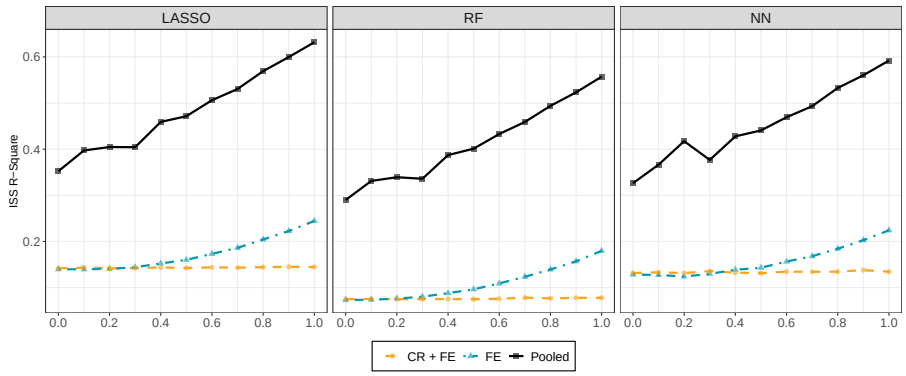
(a) $c_1 \in \{0, 0.1, 0.2, ..., 1\}, c_2 = 0$



(b) $c_1 \in \{0, 0.1, 0.2, ..., 1\}, c_2 = 1$



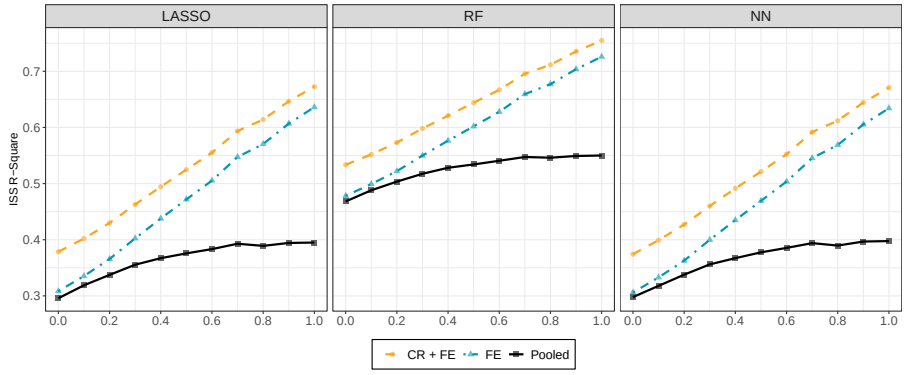
(c) $c_1 = 0, c_2 \in \{0, 0.1, 0.2, ..., 1\}$



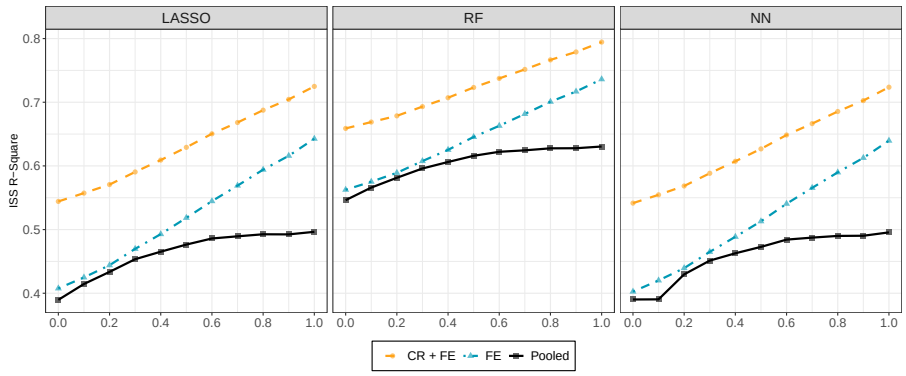
(d) $c_1 = 1, c_2 \in \{0, 0.1, 0.2, ..., 1\}$

Notes. This figure presents the average mean squared errors over 1,000 Monte Carlo repetitions for $f(\mathbf{x}_{it}) = 0.4x_{it,1} + 0.3x_{it,1}x_{it,2} + 0.12\text{sgn}(x_{it,3})$ using LASSO, random forests, and neural networks with three hidden layers of 32, 16, and 8 neurons, respectively. The number of predictors $d = 5$, the number of entities $N = 10$, and the length of sample size $T = 100$. "Pooled" stands for using the machine learning method by pooling all data. "FE" stands for using the proposed machine learning based panel data model with fixed effects. "CR + FE" stands for using the proposed machine learning panel data model with both fixed effects and cross-sectional dependence.

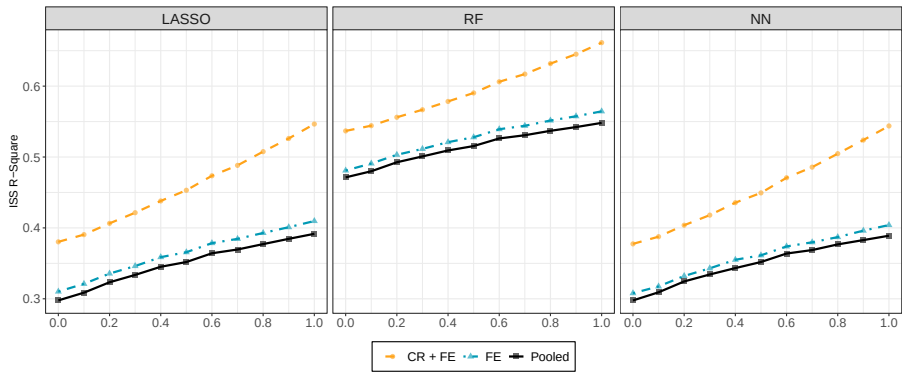
Figure 3: In-sample R^2 under Type 1



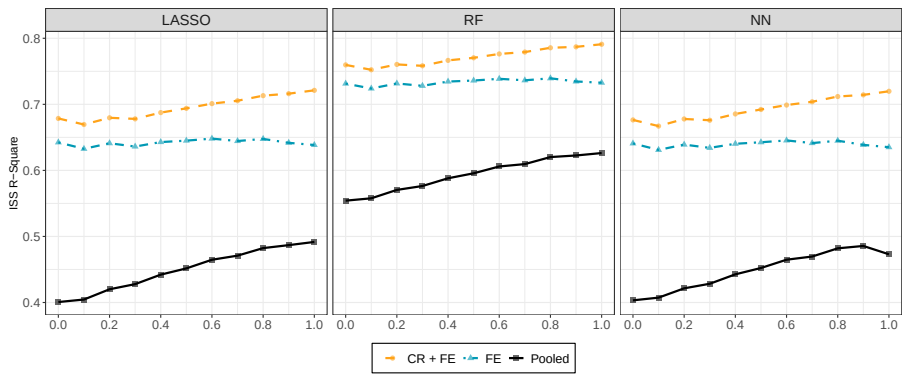
(a) $c_1 \in \{0, 0.1, 0.2, ..., 1\}, c_2 = 0$



(b) $c_1 \in \{0, 0.1, 0.2, ..., 1\}, c_2 = 1$



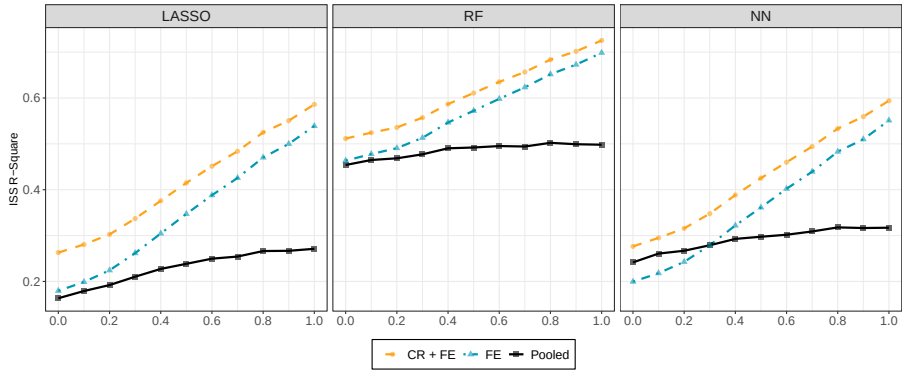
(c) $c_1 = 0, c_2 \in \{0, 0.1, 0.2, ..., 1\}$



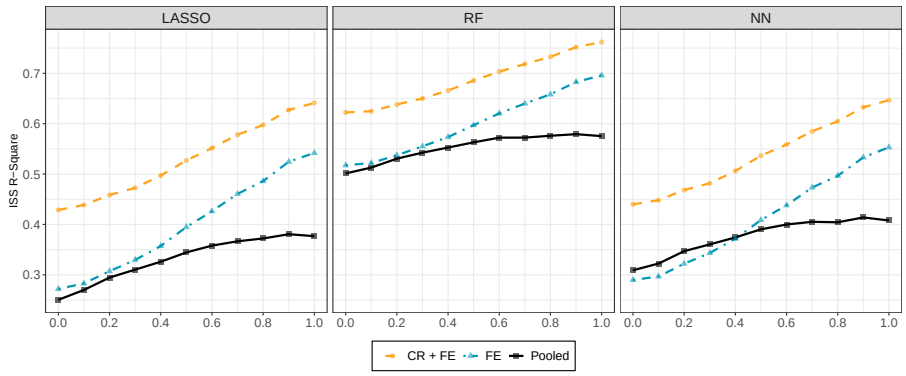
(d) $c_1 = 1, c_2 \in \{0, 0.1, 0.2, ..., 1\}$

Notes. This figure presents the average R^2s over 1,000 Monte Carlo repetitions for $f(\mathbf{x}_{it}) = 0.2x_{it,1} + 0.2x_{it,2} + 0.2x_{it,3}$ using LASSO, random forests, and neural networks with three hidden layers of 32, 16, and 8 neurons, respectively. The number of predictors $d = 5$, the number of entities $N = 10$, and the length of sample size $T = 100$. "Pooled" stands for using the machine learning method by pooling all data. "FE" stands for using the proposed machine learning based panel data model with fixed effects. "CR + FE" stands for using the proposed machine learning panel data model with both fixed effects and cross-sectional dependence.

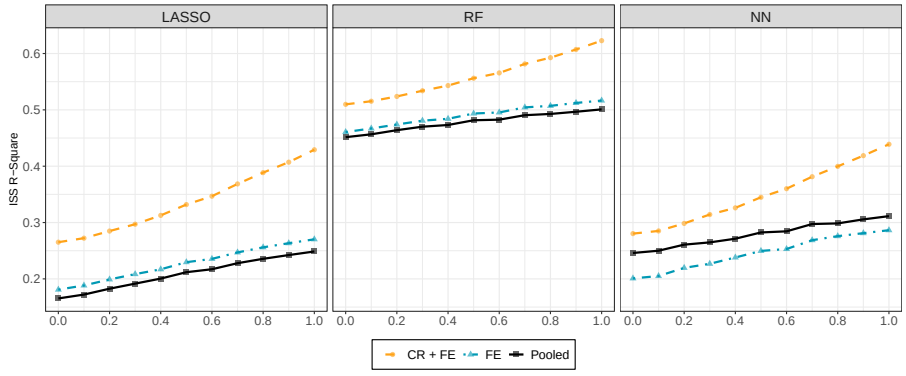
Figure 4: In-sample R^2 under Type 2



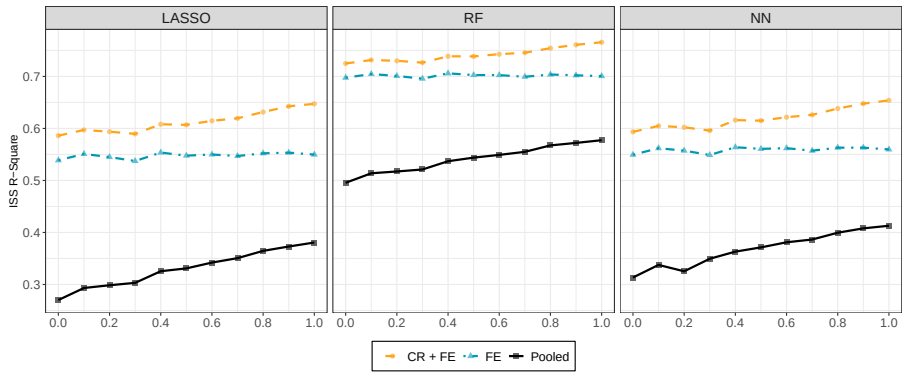
(a) $c_1 \in \{0, 0.1, 0.2, ..., 1\}, c_2 = 0$



(b) $c_1 \in \{0, 0.1, 0.2, ..., 1\}, c_2 = 1$



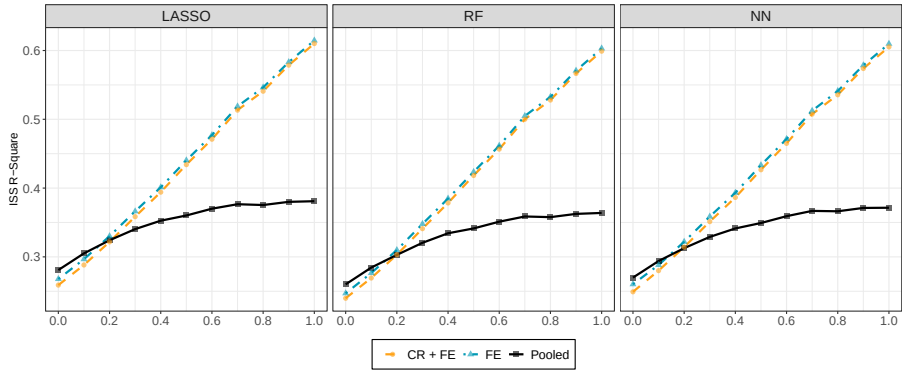
(c) $c_1 = 0, c_2 \in \{0, 0.1, 0.2, ..., 1\}$



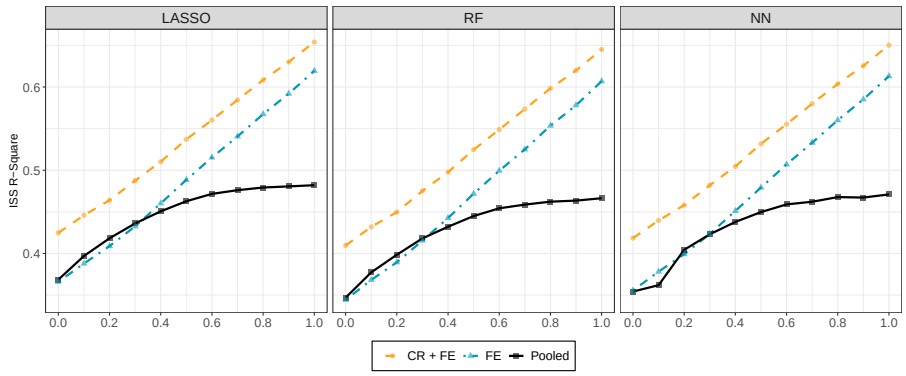
(d) $c_1 = 1, c_2 \in \{0, 0.1, 0.2, ..., 1\}$

Notes. This figure presents the average R^2s over 1,000 Monte Carlo repetitions for $f(\mathbf{x}_{it}) = 0.4x_{it,1} + 0.3x_{it,1}x_{it,2} + 0.12\text{sgn}(x_{it,3})$ using LASSO, random forests, and neural networks with three hidden layers of 32, 16, and 8 neurons, respectively. The number of predictors $d = 5$, the number of entities $N = 10$, and the length of sample size $T = 100$. "Pooled" stands for using the machine learning method by pooling all data. "FE" stands for using the proposed machine learning based panel data model with fixed effects. "CR + FE" stands for using the proposed machine learning panel data model with both fixed effects and cross-sectional dependence.

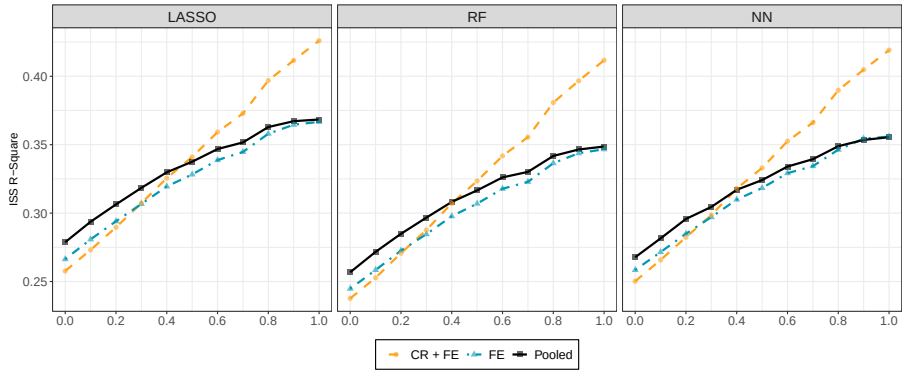
Figure 5: Out-of-sample R^2 under Type 1



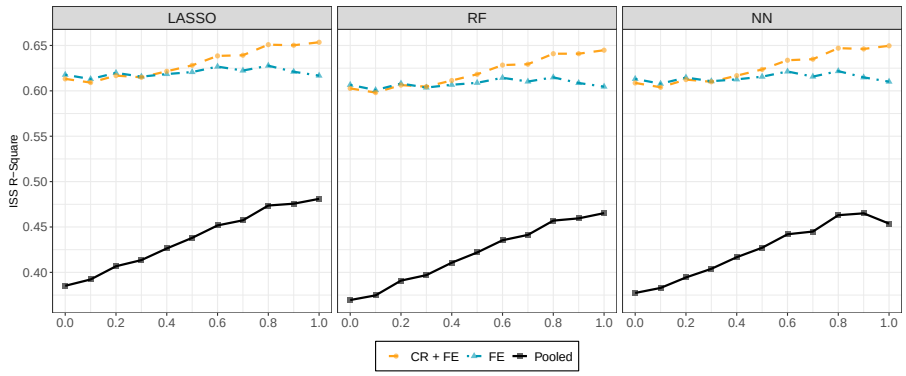
(a) $c_1 \in \{0, 0.1, 0.2, ..., 1\}, c_2 = 0$



(b) $c_1 \in \{0, 0.1, 0.2, ..., 1\}, c_2 = 1$



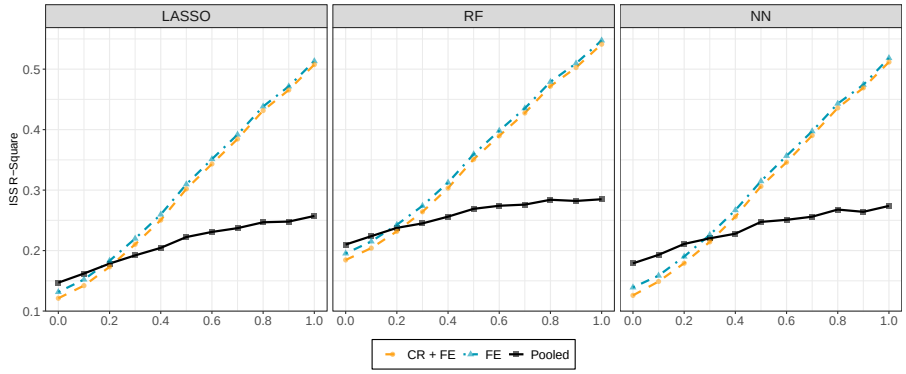
(c) $c_1 = 0, c_2 \in \{0, 0.1, 0.2, ..., 1\}$



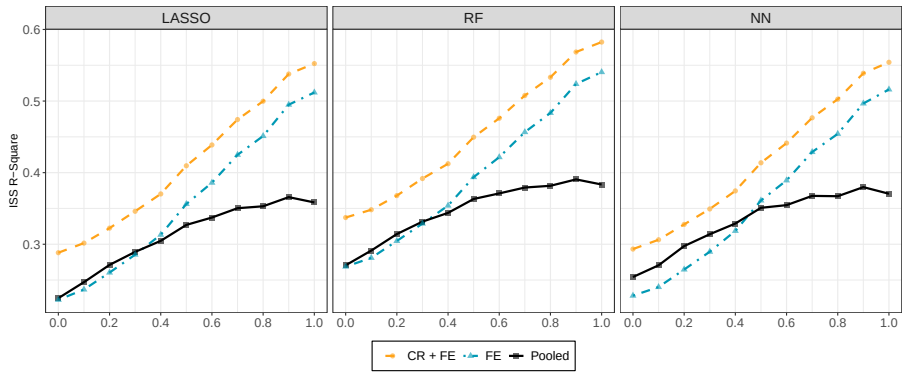
(d) $c_1 = 1, c_2 \in \{0, 0.1, 0.2, ..., 1\}$

Notes. This figure presents the average R^2s over 1,000 Monte Carlo repetitions for $f(\mathbf{x}_{it}) = 0.2x_{it,1} + 0.2x_{it,2} + 0.2x_{it,3}$ using LASSO, random forests, and neural networks with three hidden layers of 32, 16, and 8 neurons, respectively. The number of predictors $d = 5$, the number of entities $N = 100$, and the length of sample size $T = 100$. "Pooled" stands for using the machine learning method by pooling all data. "FE" stands for using the proposed machine learning based panel data model with fixed effects. "CR + FE" stands for using the proposed machine learning panel data model with both fixed effects and cross-sectional dependence.

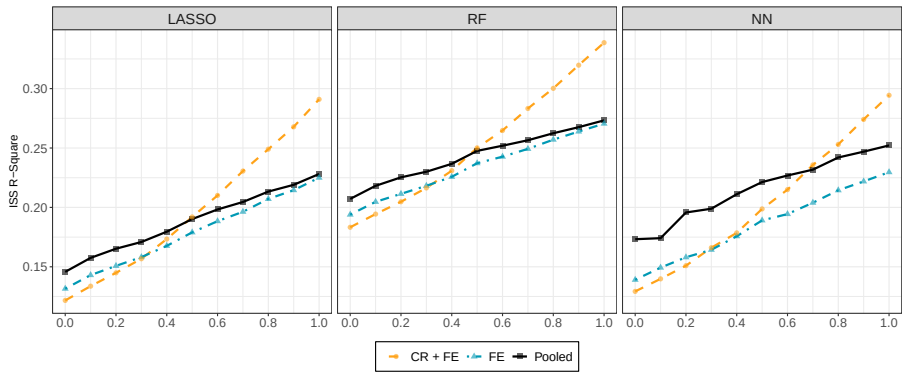
Figure 6: Out-of-sample R^2 under Type 2



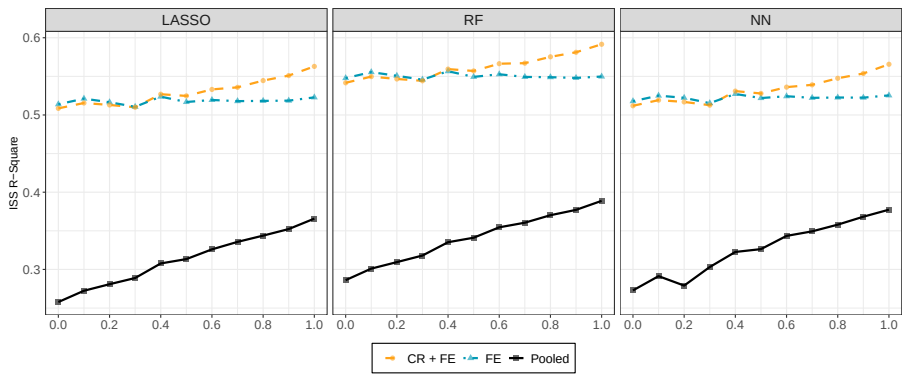
(a) $c_1 \in \{0, 0.1, 0.2, ..., 1\}, c_2 = 0$



(b) $c_1 \in \{0, 0.1, 0.2, ..., 1\}, c_2 = 1$



(c) $c_1 = 0, c_2 \in \{0, 0.1, 0.2, ..., 1\}$



(d) $c_1 = 1, c_2 \in \{0, 0.1, 0.2, ..., 1\}$

Notes. This figure presents the average R^2s over 1,000 Monte Carlo repetitions for $f(\mathbf{x}_{it}) = 0.4x_{it,1} + 0.3x_{it,1}x_{it,2} + 0.12\text{sgn}(x_{it,3})$ using LASSO, random forests, and neural networks with three hidden layers of 32, 16, and 8 neurons, respectively. The number of predictors $d = 5$, the number of entities $N = 10$, and the length of sample size $T = 100$. "Pooled" stands for using the machine learning method by pooling all data. "FE" stands for using the proposed machine learning based panel data model with fixed effects. "CR + FE" stands for using the proposed machine learning panel data model with both fixed effects and cross-sectional dependence.

Table 1: MSE under Type 1: $f(\mathbf{x}_{it}) = 0.2x_{it,1} + 0.2x_{it,2} + 0.2x_{it,3}$

		LASSO			Random Forests			Neural Networks		
N	T	Pooled	FE	CR + FE	Pooled	FE	CR + FE	Pooled	FE	CR + FE
Panel A: $c_1 = c_2 = 0$										
$d = 5$										
10	100	0.0109	0.0116	0.0146	0.0558	0.0558	0.0566	0.0312	0.0233	0.0252
	200	0.0054	0.0055	0.0070	0.0461	0.0457	0.0443	0.0175	0.0133	0.0143
	400	0.0028	0.0030	0.0037	0.0378	0.0377	0.0368	0.0123	0.0074	0.0079
20	100	0.0061	0.0066	0.0075	0.0463	0.0460	0.0461	0.0147	0.0148	0.0154
	200	0.0030	0.0032	0.0037	0.0382	0.0384	0.0374	0.0104	0.0078	0.0084
	400	0.0013	0.0014	0.0016	0.0318	0.0319	0.0311	0.0104	0.0042	0.0044
$d = 10$										
10	100	0.0155	0.0178	0.0227	0.1140	0.1151	0.1192	0.0938	0.0437	0.0496
	200	0.0088	0.0091	0.0112	0.1028	0.1036	0.1016	0.0831	0.0246	0.0268
	400	0.0040	0.0042	0.0052	0.0926	0.0918	0.0892	0.0889	0.0132	0.0180
20	100	0.0085	0.0085	0.0104	0.1012	0.1030	0.1031	0.1299	0.0279	0.0251
	200	0.0040	0.0043	0.0051	0.0916	0.0913	0.0897	0.1158	0.0135	0.0141
	400	0.0019	0.0021	0.0024	0.0803	0.0805	0.0795	0.0829	0.0074	0.0075
Panel B: $c_1 = 1, c_2 = 0$										
$d = 5$										
10	100	0.2269	0.0111	0.0145	0.2627	0.0560	0.0562	0.2430	0.0238	0.0254
	200	0.2300	0.0064	0.0074	0.2705	0.0454	0.0444	0.2519	0.0133	0.0150
	400	0.2251	0.0034	0.0038	0.2667	0.0389	0.0373	0.2528	0.0079	0.0085
20	100	0.2374	0.0060	0.0070	0.2780	0.0455	0.0455	0.2672	0.0141	0.0141
	200	0.2305	0.0029	0.0036	0.2699	0.0383	0.0374	0.2562	0.0076	0.0082
	400	0.2197	0.0016	0.0018	0.2593	0.0322	0.0315	0.2412	0.0044	0.0044
$d = 10$										
10	100	0.3081	0.0175	0.0223	0.4225	0.1184	0.1174	0.4448	0.0480	0.0477
	200	0.2942	0.0091	0.0117	0.3992	0.1010	0.1015	0.4295	0.0235	0.0296
	400	0.2687	0.0043	0.0055	0.3787	0.0926	0.0897	0.4203	0.0133	0.0140
20	100	0.2868	0.0083	0.0097	0.3920	0.0991	0.1003	0.4301	0.0236	0.0250
	200	0.3051	0.0047	0.0055	0.4168	0.0910	0.0915	0.4452	0.0133	0.0139
	400	0.3015	0.0023	0.0027	0.4160	0.0803	0.0801	0.4525	0.0081	0.0084
Panel C: $c_1 = c_2 = 1$										
$d = 5$										
10	100	0.5601	0.1182	0.0164	0.5900	0.1575	0.0591	0.5725	0.1279	0.0270
	200	0.4714	0.1022	0.0086	0.5133	0.1417	0.0463	0.4892	0.1109	0.0169
	400	0.5477	0.1058	0.0047	0.5816	0.1409	0.0373	0.5752	0.1126	0.0092
20	100	0.4964	0.1058	0.0113	0.5319	0.1425	0.0500	0.5053	0.1131	0.0197
	200	0.4905	0.1009	0.0050	0.5251	0.1352	0.0393	0.5158	0.1105	0.0091
	400	0.5267	0.1009	0.0024	0.5637	0.1340	0.0322	0.5452	0.1066	0.0052
$d = 10$										
10	100	0.6585	0.1804	0.0265	0.7669	0.2940	0.1282	0.7302	0.2026	0.0531
	200	0.6864	0.1811	0.0112	0.7968	0.2863	0.1034	0.7545	0.1988	0.0269
	400	0.6847	0.1719	0.0069	0.8025	0.2751	0.0911	0.8099	0.1856	0.0161
20	100	0.7064	0.1826	0.0130	0.8262	0.2866	0.1046	0.7832	0.1983	0.0330
	200	0.6936	0.1732	0.0067	0.8091	0.2671	0.0897	0.7606	0.1831	0.0210
	400	0.6804	0.1737	0.0035	0.7974	0.2752	0.0808	0.7700	0.1826	0.0089

Notes. This table presents the average mean squared errors (MSEs) over 1,000 Monte Carlo repetitions for $f(\mathbf{x}_{it}) = 0.2x_{it,1} + 0.2x_{it,2} + 0.2x_{it,3}$ using LASSO, random forests, and neural networks with three hidden layers of 32, 16, and 8 neurons, respectively. The number of predictors $d \in \{5, 10\}$, the number of entities $N \in \{10, 20\}$, and the length of sample size $T \in \{100, 200, 400\}$. $c_1 = c_2 = 0$ indicates the true model contains neither fixed effectss nor cross-sectional dependence. $c_1 = 1, c_2 = 0$ indicates the true model contains fixed effectss. $c_1 = c_2 = 1$ indicates the true model contains both fixed effects and cross-sectional dependence. "Pooled" stands for using the machine learning method by pooling all data. "FE" stands for using the proposed machine learning based panel data model with fixed effects. "CR + FE" stands for using the proposed machine learning panel data model with both fixed effects and cross-sectional dependence.

Table 2: MSE under Type 2: $f(\mathbf{x}_{it}) = 0.4x_{it,1} + 0.3x_{it,1}x_{it,2} + 0.12\text{sgn}(x_{it,3})$

		LASSO			Random Forests			Neural Networks		
N	T	Pooled	FE	CR + FE	Pooled	FE	CR + FE	Pooled	FE	CR + FE
Panel A: $c_1 = c_2 = 0$										
$d = 5$										
10	100	0.1425	0.1429	0.1448	0.0716	0.0727	0.0740	0.0999	0.1231	0.1320
	200	0.1362	0.1364	0.1379	0.0574	0.0578	0.0586	0.0580	0.1016	0.0941
	400	0.1322	0.1325	0.1332	0.0453	0.0458	0.0459	0.0353	0.0414	0.0517
20	100	0.1372	0.1377	0.1385	0.0573	0.0579	0.0587	0.0483	0.0890	0.0889
	200	0.1341	0.1341	0.1346	0.0472	0.0472	0.0474	0.0349	0.0497	0.0494
	400	0.1329	0.1330	0.1333	0.0375	0.0375	0.0378	0.0317	0.0255	0.0295
$d = 10$										
10	100	0.1378	0.1393	0.1416	0.1330	0.1357	0.1339	0.2005	0.1388	0.1516
	200	0.1339	0.1341	0.1357	0.1151	0.1160	0.1142	0.2024	0.0902	0.1029
	400	0.1337	0.1341	0.1351	0.0986	0.0993	0.0964	0.1396	0.0502	0.0520
20	100	0.1391	0.1394	0.1405	0.1149	0.1177	0.1163	0.2090	0.0916	0.0936
	200	0.1336	0.1336	0.1345	0.0975	0.0973	0.0969	0.1567	0.0519	0.0540
	400	0.1334	0.1335	0.1337	0.0629	0.0635	0.0658	0.1159	0.0318	0.0299
Panel B: $c_1 = 1, c_2 = 0$										
$d = 5$										
10	100	0.3580	0.1428	0.1447	0.3007	0.0730	0.0736	0.3303	0.1342	0.1375
	200	0.3511	0.1369	0.1378	0.2801	0.0590	0.0604	0.3097	0.0908	0.0956
	400	0.4048	0.1369	0.1374	0.3221	0.0464	0.0467	0.3547	0.0427	0.0480
20	100	0.3701	0.1378	0.1389	0.3012	0.0578	0.0582	0.3269	0.0887	0.0937
	200	0.3599	0.1331	0.1334	0.2819	0.0465	0.0466	0.2979	0.0492	0.0453
	400	0.3707	0.1353	0.1355	0.2812	0.0370	0.0373	0.2801	0.0234	0.0216
$d = 10$										
10	100	0.4095	0.1451	0.1495	0.4246	0.1357	0.1386	0.5153	0.1415	0.1623
	200	0.4046	0.1352	0.1371	0.4148	0.1192	0.1166	0.5101	0.0984	0.1005
	400	0.4246	0.1338	0.1348	0.4237	0.0928	0.0930	0.5953	0.0462	0.0511
20	100	0.4526	0.1361	0.1376	0.4543	0.1191	0.1181	0.5945	0.1065	0.1038
	200	0.4631	0.1346	0.1357	0.4704	0.0991	0.0996	0.5834	0.0492	0.0553
	400	0.4252	0.1335	0.1340	0.4029	0.0589	0.0610	0.5189	0.0261	0.0313
Panel C: $c_1 = c_2 = 1$										
$d = 5$										
10	100	0.6462	0.2423	0.1484	0.5676	0.1773	0.0792	0.5994	0.2213	0.1390
	200	0.6376	0.2408	0.1366	0.5599	0.1673	0.0611	0.5773	0.1914	0.0992
	400	0.6484	0.2348	0.1353	0.5609	0.1505	0.0484	0.5775	0.1524	0.0515
20	100	0.6419	0.2389	0.1391	0.5581	0.1597	0.0609	0.5769	0.1940	0.1019
	200	0.6598	0.2377	0.1375	0.5806	0.1520	0.0483	0.5773	0.1520	0.0477
	400	0.6535	0.2362	0.1358	0.5612	0.1430	0.0388	0.5591	0.1315	0.0267
$d = 10$										
10	100	0.7583	0.2808	0.1486	0.7697	0.2994	0.1441	0.8301	0.2894	0.1573
	200	0.8442	0.3110	0.1407	0.8441	0.3085	0.1223	0.8556	0.2735	0.0981
	400	0.7585	0.3001	0.1348	0.7527	0.2808	0.0936	0.7912	0.2280	0.0560
20	100	0.8179	0.3058	0.1425	0.8183	0.2958	0.1203	0.8236	0.2661	0.1062
	200	0.7957	0.3040	0.1365	0.7973	0.2868	0.0987	0.8082	0.2268	0.0637
	400	0.8639	0.3081	0.1368	0.8501	0.2727	0.0638	0.8619	0.2073	0.0292

Notes. This table presents the average mean squared errors (MSEs) over 1,000 Monte Carlo repetitions for $f(\mathbf{x}_{it}) = 0.4x_{it,1} + 0.3x_{it,1}x_{it,2} + 0.12\text{sgn}(x_{it,3})$ using LASSO, random forests, and neural networks with three hidden layers of 32, 16, and 8 neurons, respectively. The number of predictors $d \in \{5, 10\}$, the number of entities $N \in \{10, 20\}$, and the length of sample size $T \in \{100, 200, 400\}$. $c_1 = c_2 = 0$ indicates the true model contains neither fixed effects nor cross-sectional dependence. $c_1 = 1, c_2 = 0$ indicates the true model contains fixed effects. $c_1 = c_2 = 1$ indicates the true model contains both fixed effects and cross-sectional dependence. "Pooled" stands for using the machine learning method by pooling all data. "FE" stands for using the proposed machine learning based panel data model with fixed effects. "CR + FE" stands for using the proposed machine learning panel data model with both fixed effects and cross-sectional dependence.

Table 3: Comparison of in-sample R^2s under Type 1: $f(\mathbf{x}_{it}) = 0.2x_{it,1} + 0.2x_{it,2} + 0.2x_{it,3}$

		LASSO			Random Forests			Neural Networks		
N	T	Pooled	FE	CR + FE	Pooled	FE	CR + FE	Pooled	FE	CR + FE
Panel A: $c_1 = c_2 = 0$										
$d = 5$										
10	100	0.2981	0.3109	0.3797	0.4746	0.4840	0.5398	0.2992	0.3088	0.3772
	200	0.2914	0.2976	0.3688	0.4600	0.4647	0.5221	0.2935	0.2956	0.3673
	400	0.2899	0.2932	0.3644	0.4506	0.4526	0.5102	0.2926	0.2923	0.3634
20	100	0.2908	0.3038	0.3377	0.4574	0.4683	0.4949	0.2952	0.3016	0.3353
	200	0.2907	0.2975	0.3318	0.4486	0.4536	0.4810	0.2934	0.2965	0.3308
	400	0.2939	0.2973	0.3323	0.4453	0.4483	0.4774	0.2992	0.2971	0.3317
$d = 10$										
10	100	0.3011	0.3135	0.3830	0.6097	0.6201	0.6595	0.3509	0.3194	0.3814
	200	0.2941	0.3011	0.3707	0.6040	0.6075	0.6547	0.3512	0.3030	0.3714
	400	0.2927	0.2961	0.3668	0.5876	0.5927	0.6376	0.3521	0.2980	0.3682
20	100	0.2953	0.3084	0.3423	0.5927	0.6057	0.6284	0.3581	0.3073	0.3448
	200	0.2907	0.2975	0.3329	0.5856	0.5918	0.6172	0.3471	0.2996	0.3347
	400	0.2916	0.2947	0.3296	0.5748	0.5774	0.6039	0.3541	0.2955	0.3306
Panel B: $c_1 = 1, c_2 = 0$										
$d = 5$										
10	100	0.4021	0.6390	0.6766	0.5569	0.7275	0.7567	0.4053	0.6373	0.6747
	200	0.3951	0.6373	0.6743	0.5427	0.7212	0.7510	0.4024	0.6362	0.6732
	400	0.3906	0.6311	0.6674	0.5325	0.7138	0.7431	0.3968	0.6307	0.6670
20	100	0.3933	0.6445	0.6626	0.5406	0.7283	0.7423	0.3947	0.6437	0.6621
	200	0.3989	0.6451	0.6632	0.5381	0.7245	0.7396	0.4043	0.6447	0.6627
	400	0.3906	0.6359	0.6541	0.5256	0.7139	0.7287	0.3956	0.6357	0.6538
$d = 10$										
10	100	0.4318	0.6436	0.6803	0.6876	0.7997	0.8239	0.4565	0.6440	0.6789
	200	0.4164	0.6260	0.6640	0.6744	0.7877	0.8140	0.4664	0.6282	0.6647
	400	0.4162	0.6293	0.6667	0.6722	0.7828	0.8089	0.4605	0.6303	0.6671
20	100	0.4253	0.6479	0.6653	0.6838	0.8018	0.8136	0.4827	0.6479	0.6664
	200	0.4285	0.6503	0.6673	0.6786	0.7992	0.8113	0.4861	0.6509	0.6680
	400	0.4226	0.6401	0.6582	0.6683	0.7859	0.8002	0.4741	0.6391	0.6583
Panel C: $c_1 = c_2 = 1$										
$d = 5$										
10	100	0.4828	0.6226	0.7083	0.6182	0.7194	0.7803	0.4813	0.6191	0.7052
	200	0.4920	0.6458	0.7283	0.6186	0.7331	0.7924	0.4939	0.6439	0.7275
	400	0.4879	0.6289	0.7148	0.6102	0.7168	0.7798	0.4936	0.6280	0.7142
20	100	0.5060	0.6624	0.7254	0.6310	0.7469	0.7905	0.5056	0.6607	0.7246
	200	0.4982	0.6492	0.7158	0.6175	0.7320	0.7806	0.5030	0.6483	0.7153
	400	0.5001	0.6496	0.7166	0.6120	0.7281	0.7777	0.5043	0.6491	0.7164
$d = 10$										
10	100	0.5499	0.6592	0.7275	0.7677	0.8204	0.8529	0.5629	0.6554	0.7259
	200	0.5469	0.6545	0.7245	0.7630	0.8157	0.8485	0.5867	0.6532	0.7238
	400	0.5490	0.6537	0.7251	0.7584	0.8111	0.8465	0.5788	0.6529	0.7256
20	100	0.5490	0.6658	0.7186	0.7644	0.8214	0.8433	0.5801	0.6654	0.7181
	200	0.5550	0.6651	0.7186	0.7615	0.8175	0.8390	0.5797	0.6644	0.7178
	400	0.5482	0.6584	0.7129	0.7518	0.8098	0.8285	0.5726	0.6578	0.7131

Notes. This table presents the average in-sample R^2s over 1,000 Monte Carlo repetitions for $f(\mathbf{x}_{it}) = 0.2x_{it,1} + 0.2x_{it,2} + 0.2x_{it,3}$ using LASSO, random forests, and neural networks with three hidden layers of 32, 16, and 8 neurons, respectively. The number of predictors $d \in \{5, 10\}$, the number of entities $N \in \{10, 20\}$, and the length of sample size $T \in \{100, 200, 400\}$. $c_1 = c_2 = 0$ indicates the true model contains neither fixed effects nor cross-sectional dependence. $c_1 = 1, c_2 = 0$ indicates the true model contains fixed effects. $c_1 = c_2 = 1$ indicates the true model contains both fixed effects and cross-sectional dependence. "Pooled" stands for using the machine learning method by pooling all data. "FE" stands for using the proposed machine learning based panel data model with fixed effects. "CR + FE" stands for using the proposed machine learning panel data model with both fixed effects and cross-sectional dependence.

Table 4: Comparison of in-sample R^2s under Type 2: $f(\mathbf{x}_{it}) = 0.4x_{it,1} + 0.3x_{it,1}x_{it,2} + 0.12\text{sgn}(x_{it,3})$

N	T	LASSO			Random Forests			Neural Networks		
		Pooled	FE	CR + FE	Pooled	FE	CR + FE	Pooled	FE	CR + FE
Panel A: $c_1 = c_2 = 0$										
$d = 5$										
10	100	0.1623	0.1783	0.2623	0.4472	0.4568	0.5065	0.2437	0.1989	0.2721
	200	0.1619	0.1701	0.2534	0.4420	0.4475	0.4971	0.2616	0.2231	0.2932
	400	0.1586	0.1632	0.2470	0.4342	0.4368	0.4856	0.2681	0.2414	0.3014
20	100	0.1637	0.1805	0.2223	0.4452	0.4541	0.4778	0.2741	0.2198	0.2598
	200	0.1585	0.1672	0.2097	0.4330	0.4375	0.4625	0.2721	0.2406	0.2727
	400	0.1589	0.1638	0.2066	0.4152	0.4156	0.4403	0.2701	0.2494	0.2829
$d = 10$										
10	100	0.1673	0.1806	0.2677	0.5806	0.5924	0.6306	0.3558	0.2244	0.2846
	200	0.1585	0.1670	0.2516	0.5847	0.5970	0.6323	0.3707	0.2212	0.2925
	400	0.1584	0.1625	0.2480	0.5413	0.5391	0.5879	0.3882	0.2371	0.3131
20	100	0.1660	0.1821	0.2232	0.5834	0.5970	0.6187	0.4134	0.2363	0.2725
	200	0.1595	0.1682	0.2117	0.5481	0.5515	0.5755	0.3443	0.2424	0.2779
	400	0.1604	0.1653	0.2078	0.4609	0.4604	0.4897	0.3465	0.2520	0.2861
Panel B: $c_1 = 1, c_2 = 0$										
$d = 5$										
10	100	0.2623	0.5294	0.5777	0.4893	0.6904	0.7184	0.3064	0.5409	0.5840
	200	0.2658	0.5240	0.5729	0.4866	0.6818	0.7102	0.3236	0.5499	0.5906
	400	0.2622	0.5210	0.5692	0.4757	0.6743	0.7022	0.3357	0.5630	0.5996
20	100	0.2712	0.5549	0.5768	0.4887	0.7043	0.7165	0.3316	0.5805	0.5972
	200	0.2667	0.5412	0.5647	0.4819	0.6910	0.7038	0.3344	0.5821	0.5997
	400	0.2735	0.5521	0.5748	0.4740	0.6840	0.6981	0.3373	0.5973	0.6140
$d = 10$										
10	100	0.3116	0.5415	0.5895	0.6452	0.7740	0.7970	0.4289	0.5585	0.6027
	200	0.2888	0.5250	0.5739	0.6348	0.7620	0.7854	0.4480	0.5491	0.5969
	400	0.2877	0.5225	0.5703	0.6295	0.7368	0.7659	0.4450	0.5643	0.6037
20	100	0.3113	0.5601	0.5831	0.6508	0.7848	0.7936	0.4487	0.5868	0.6064
	200	0.3153	0.5640	0.5864	0.6473	0.7741	0.7907	0.4491	0.6027	0.6215
	400	0.3016	0.5483	0.5714	0.6241	0.6997	0.7194	0.4296	0.5940	0.6129
Panel C: $c_1 = c_2 = 1$										
$d = 5$										
10	100	0.3763	0.5449	0.6446	0.5777	0.6989	0.7638	0.4168	0.5559	0.6517
	200	0.3697	0.5326	0.6338	0.5640	0.6851	0.7527	0.4157	0.5573	0.6522
	400	0.3718	0.5326	0.6353	0.5576	0.6794	0.7490	0.4249	0.5692	0.6623
20	100	0.3836	0.5660	0.6388	0.5731	0.7096	0.7600	0.4266	0.5894	0.6564
	200	0.3826	0.5568	0.6353	0.5629	0.6951	0.7483	0.4294	0.5903	0.6664
	400	0.3899	0.5659	0.6429	0.5620	0.6971	0.7483	0.4394	0.6042	0.6758
$d = 10$										
10	100	0.4520	0.5786	0.6566	0.7308	0.7954	0.8271	0.5013	0.5873	0.6671
	200	0.4432	0.5654	0.6491	0.7317	0.7929	0.8248	0.5437	0.5875	0.6715
	400	0.4385	0.5587	0.6440	0.7199	0.7822	0.7980	0.5351	0.5874	0.6720
20	100	0.4341	0.5670	0.6304	0.7268	0.7929	0.8160	0.5367	0.5872	0.6529
	200	0.4446	0.5724	0.6352	0.7231	0.7894	0.7980	0.5304	0.6057	0.6685
	400	0.4395	0.5635	0.6288	0.7157	0.7807	0.7573	0.5198	0.6026	0.6630

Notes. This table presents the average in-sample R^2s over 1,000 Monte Carlo repetitions for $f(\mathbf{x}_{it}) = 0.4x_{it,1} + 0.3x_{it,1}x_{it,2} + 0.12\text{sgn}(x_{it,3})$ using LASSO, random forests, and neural networks with three hidden layers of 32, 16, and 8 neurons, respectively. The number of predictors $d \in \{5, 10\}$, the number of entities $N \in \{10, 20\}$, and the length of sample size $T \in \{100, 200, 400\}$. $c_1 = c_2 = 0$ indicates the true model contains neither fixed effects nor cross-sectional dependence. $c_1 = 1, c_2 = 0$ indicates the true model contains fixed effects. $c_1 = c_2 = 1$ indicates the true model contains both fixed effects and cross-sectional dependence. "Pooled" stands for using the machine learning method by pooling all data. "FE" stands for using the proposed machine learning based panel data model with fixed effects. "CR + FE" stands for using the proposed machine learning panel data model with both fixed effects and cross-sectional dependence.

Table 5: Comparison of out-of-sample R^2s under Type 1: $f(\mathbf{x}_{it}) = 0.2x_{it,1} + 0.2x_{it,2} + 0.2x_{it,3}$

		LASSO			Random Forests			Neural Networks		
N	T	Pooled	FE	CR + FE	Pooled	FE	CR + FE	Pooled	FE	CR + FE
Panel A: $c_1 = c_2 = 0$										
$d = 5$										
10	100	0.2801	0.2653	0.2564	0.2577	0.2429	0.2351	0.2687	0.2566	0.2472
	200	0.2862	0.2803	0.2763	0.2682	0.2625	0.2594	0.2794	0.2745	0.2705
	400	0.2877	0.2851	0.2834	0.2727	0.2700	0.2694	0.2819	0.2816	0.2797
20	100	0.2852	0.2713	0.2678	0.2670	0.2528	0.2495	0.2744	0.2654	0.2618
	200	0.2892	0.2822	0.2801	0.2736	0.2667	0.2651	0.2838	0.2785	0.2769
	400	0.2910	0.2876	0.2865	0.2783	0.2748	0.2741	0.2837	0.2854	0.2842
$d = 10$										
10	100	0.2736	0.2589	0.2358	0.2410	0.2267	0.2067	0.2107	0.2370	0.2167
	200	0.2825	0.2761	0.2675	0.2571	0.2510	0.2434	0.2154	0.2644	0.2572
	400	0.2873	0.2844	0.2808	0.2665	0.2638	0.2615	0.2186	0.2775	0.2742
20	100	0.2799	0.2652	0.2563	0.2546	0.2400	0.2318	0.2040	0.2492	0.2431
	200	0.2879	0.2816	0.2775	0.2672	0.2611	0.2573	0.2232	0.2756	0.2710
	400	0.2885	0.2855	0.2833	0.2708	0.2678	0.2661	0.2156	0.2818	0.2796
Panel B: $c_1 = 1, c_2 = 0$										
$d = 5$										
10	100	0.3806	0.6144	0.6102	0.3665	0.6048	0.6015	0.3721	0.6102	0.6054
	200	0.3879	0.6260	0.6240	0.3760	0.6174	0.6163	0.3813	0.6231	0.6214
	400	0.3845	0.6236	0.6225	0.3737	0.6156	0.6150	0.3801	0.6217	0.6207
20	100	0.3838	0.6234	0.6211	0.3684	0.6141	0.6120	0.3781	0.6200	0.6181
	200	0.3934	0.6367	0.6356	0.3807	0.6288	0.6278	0.3874	0.6349	0.6339
	400	0.3866	0.6298	0.6294	0.3767	0.6231	0.6229	0.3831	0.6287	0.6283
$d = 10$										
10	100	0.4048	0.6161	0.6046	0.3875	0.6006	0.5899	0.3727	0.6039	0.5932
	200	0.3999	0.6105	0.6059	0.3837	0.5976	0.5936	0.3473	0.6041	0.5993
	400	0.4124	0.6217	0.6196	0.3980	0.6101	0.6087	0.3684	0.6178	0.6163
20	100	0.4103	0.6306	0.6256	0.3941	0.6176	0.6124	0.3427	0.6236	0.6184
	200	0.4154	0.6383	0.6361	0.4010	0.6276	0.6255	0.3528	0.6344	0.6325
	400	0.4188	0.6370	0.6360	0.4068	0.6278	0.6269	0.3634	0.6337	0.6341
Panel C: $c_1 = c_2 = 1$										
$d = 5$										
10	100	0.4817	0.6082	0.6440	0.4657	0.5945	0.6338	0.4704	0.6007	0.6387
	200	0.4968	0.6410	0.6776	0.4836	0.6311	0.6704	0.4892	0.6367	0.6750
	400	0.4872	0.6246	0.6647	0.4765	0.6160	0.6587	0.4805	0.6220	0.6632
20	100	0.4961	0.6414	0.6852	0.4828	0.6321	0.6776	0.4879	0.6373	0.6825
	200	0.4919	0.6378	0.6816	0.4808	0.6298	0.6755	0.4844	0.6352	0.6800
	400	0.4955	0.6437	0.6875	0.4868	0.6370	0.6824	0.4911	0.6421	0.6866
$d = 10$										
10	100	0.5267	0.6282	0.6551	0.5072	0.6117	0.6420	0.5034	0.6129	0.6455
	200	0.5369	0.6386	0.6688	0.5219	0.6252	0.6580	0.4888	0.6313	0.6626
	400	0.5451	0.6484	0.6803	0.5323	0.6370	0.6714	0.5136	0.6443	0.6775
20	100	0.5408	0.6447	0.6788	0.5252	0.6315	0.6675	0.4977	0.6382	0.6716
	200	0.5516	0.6569	0.6912	0.5387	0.6461	0.6822	0.5230	0.6531	0.6873
	400	0.5480	0.6548	0.6905	0.5374	0.6456	0.6832	0.5221	0.6525	0.6889

Notes. This table presents the average out-of-sample R^2s over 1,000 Monte Carlo repetitions for $f(\mathbf{x}_{it}) = 0.2x_{it,1} + 0.2x_{it,2} + 0.2x_{it,3}$ using LASSO, random forests, and neural networks with three hidden layers of 32, 16, and 8 neurons, respectively. The number of predictors $d \in \{5, 10\}$, the number of entities $N \in \{10, 20\}$, and the length of sample size $T \in \{100, 200, 400\}$. $c_1 = c_2 = 0$ indicates the true model contains neither fixed effects nor cross-sectional dependence. $c_1 = 1, c_2 = 0$ indicates the true model contains fixed effects. $c_1 = c_2 = 1$ indicates the true model contains both fixed effects and cross-sectional dependence. "Pooled" stands for using the machine learning method by pooling all data. "FE" stands for using the proposed machine learning based panel data model with fixed effects. "CR + FE" stands for using the proposed machine learning panel data model with both fixed effects and cross-sectional dependence.

Table 6: Comparison of out-of-sample R^2s under Type 2: $f(\mathbf{x}_{it}) = 0.4x_{it,1}+0.3x_{it,1}x_{it,2}+0.12\text{sgn}(x_{it,3})$

N	T	LASSO			Random Forests			Neural Networks		
		Pooled	FE	CR + FE	Pooled	FE	CR + FE	Pooled	FE	CR + FE
Panel A: $c_1 = c_2 = 0$										
$d = 5$										
10	100	0.1487	0.1333	0.1222	0.2121	0.1982	0.1857	0.1810	0.1393	0.1227
	200	0.1554	0.1481	0.1434	0.2275	0.2212	0.2147	0.2137	0.1893	0.1815
	400	0.1592	0.1561	0.1537	0.2358	0.2330	0.2297	0.2318	0.2262	0.2118
20	100	0.1533	0.1387	0.1341	0.2263	0.2142	0.2086	0.2045	0.1691	0.1696
	200	0.1550	0.1476	0.1461	0.2339	0.2274	0.2255	0.2253	0.2148	0.2100
	400	0.1602	0.1568	0.1559	0.2404	0.2371	0.2355	0.2348	0.2374	0.2353
$d = 10$										
10	100	0.1397	0.1241	0.0996	0.1805	0.1650	0.1409	0.0519	0.1222	0.0901
	200	0.1474	0.1400	0.1287	0.2052	0.1981	0.1867	0.0830	0.1657	0.1545
	400	0.1534	0.1505	0.1456	0.2184	0.2151	0.2091	0.0742	0.2054	0.2069
20	100	0.1545	0.1399	0.1288	0.2132	0.1996	0.1880	0.0457	0.1670	0.1580
	200	0.1553	0.1482	0.1429	0.2213	0.2142	0.2088	0.1315	0.2067	0.2033
	400	0.1568	0.1536	0.1512	0.2326	0.2289	0.2254	0.1387	0.2328	0.2311
Panel B: $c_1 = 1, c_2 = 0$										
$d = 5$										
10	100	0.2522	0.5013	0.4950	0.2799	0.5343	0.5284	0.2681	0.5044	0.4974
	200	0.2485	0.5004	0.4975	0.2827	0.5383	0.5357	0.2708	0.5191	0.5124
	400	0.2609	0.5161	0.5146	0.2988	0.5578	0.5557	0.2888	0.5520	0.5463
20	100	0.2655	0.5326	0.5299	0.2979	0.5721	0.5690	0.2871	0.5524	0.5480
	200	0.2625	0.5282	0.5268	0.3004	0.5698	0.5682	0.2918	0.5652	0.5623
	400	0.2737	0.5486	0.5480	0.3128	0.5908	0.5898	0.3084	0.5923	0.5899
$d = 10$										
10	100	0.2751	0.5105	0.4955	0.2964	0.5327	0.5193	0.2040	0.5061	0.4928
	200	0.2796	0.5137	0.5080	0.3076	0.5458	0.5393	0.1891	0.5247	0.5221
	400	0.2810	0.5173	0.5144	0.3160	0.5533	0.5500	0.2157	0.5490	0.5458
20	100	0.2929	0.5357	0.5296	0.3191	0.5660	0.5594	0.2303	0.5465	0.5429
	200	0.3076	0.5535	0.5508	0.3372	0.5878	0.5853	0.2536	0.5831	0.5811
	400	0.2967	0.5425	0.5412	0.3340	0.5825	0.5804	0.2588	0.5844	0.5836
Panel C: $c_1 = c_2 = 1$										
$d = 5$										
10	100	0.3657	0.5201	0.5608	0.3896	0.5497	0.5903	0.3753	0.5225	0.5631
	200	0.3603	0.5184	0.5625	0.3895	0.5500	0.5951	0.3833	0.5362	0.5768
	400	0.3691	0.5249	0.5697	0.4026	0.5601	0.6056	0.4002	0.5584	0.5993
20	100	0.3786	0.5412	0.5915	0.4051	0.5720	0.6237	0.3965	0.5592	0.6057
	200	0.3778	0.5453	0.5943	0.4074	0.5782	0.6286	0.4087	0.5759	0.6263
	400	0.3877	0.5611	0.6097	0.4205	0.5963	0.6456	0.4207	0.5979	0.6447
$d = 10$										
10	100	0.4277	0.5398	0.5694	0.4471	0.5622	0.5901	0.4173	0.5341	0.5657
	200	0.4309	0.5469	0.5799	0.4553	0.5736	0.6071	0.3949	0.5597	0.5943
	400	0.4322	0.5501	0.5870	0.4598	0.5805	0.6171	0.4077	0.5736	0.6142
20	100	0.4278	0.5430	0.5818	0.4489	0.5680	0.6081	0.3923	0.5526	0.5945
	200	0.4393	0.5598	0.6005	0.4656	0.5895	0.6299	0.4285	0.5877	0.6300
	400	0.4335	0.5553	0.5966	0.4638	0.5886	0.6306	0.4349	0.5905	0.6303

Notes. This table presents the average out-of-sample R^2s over 1,000 Monte Carlo repetitions for $f(\mathbf{x}_{it}) = 0.4x_{it,1} + 0.3x_{it,1}x_{it,2} + 0.12\text{sgn}(x_{it,3})$ using LASSO, random forests, and neural networks with three hidden layers of 32, 16, and 8 neurons, respectively. The number of predictors $d \in \{5, 10\}$, the number of entities $N \in \{10, 20\}$, and the length of sample size $T \in \{100, 200, 400\}$. $c_1 = c_2 = 0$ indicates the true model contains neither fixed effects nor cross-sectional dependence. $c_1 = 1, c_2 = 0$ indicates the true model contains fixed effects. $c_1 = c_2 = 1$ indicates the true model contains both fixed effects and cross-sectional dependence. "Pooled" stands for using the machine learning method by pooling all data. "FE" stands for using the proposed machine learning based panel data model with fixed effects. "CR + FE" stands for using the proposed machine learning panel data model with both fixed effects and cross-sectional dependence.

Table 7: Summary of model specifications

Models	Production Function	ω_{it}
LASSO	$f(\ln L_{it}, \ln K_{it})$	$\alpha_i + \gamma_i^\intercal \lambda_t$
Random Forest	$f(\ln L_{it}, \ln K_{it})$	$\alpha_i + \gamma_i^\intercal \lambda_t$
Neural Network	$f(\ln L_{it}, \ln K_{it})$	$\alpha_i + \gamma_i^\intercal \lambda_t$
Pooled	$\gamma_L \ln L_{it} + \gamma_K \ln K_{it}$	0
Fixed Effects	$\gamma_L \ln L_{it} + \gamma_K \ln K_{it}$	$\alpha_i + \zeta_t$
Olley and Pakes (1996)	$\gamma_L \ln L_{it} + \gamma_K \ln K_{it}$	$h(\ln I_{it}, \ln K_{it})$
Levinsohn and Petrin (2003)	$\gamma_L \ln L_{it} + \gamma_K \ln K_{it}$	$h(\ln M_{it}, \ln K_{it})$
Ackerberg et al. (2015)	$\gamma_L \ln L_{it} + \gamma_K \ln K_{it}$	$h(\ln M_{it}, \ln K_{it}, \ln L_{it})$
Ackerberg et al. (2015)	$\gamma_L \ln L_{it} + \gamma_K \ln K_{it}$	$h(\ln I_{it}, \ln K_{it}, \ln L_{it})$

Notes. This table presents the specifications of eight models. The proposed three machine learning-based panel data models assume that the main part of production function is an unknown form and formulated as $f(\ln L_{it}, \ln K_{it})$, while the other five existing methods are in the linear form, i.e., $\gamma_L \ln L_{it} + \gamma_K \ln K_{it}$. The proposed three machine learning-based panel data models assume that the unobserved productivity $\omega_{it} = \alpha_i + \gamma_i^\intercal \lambda_t$, where both fixed effects α_i and common factor λ_t may be endogenous. The pooled panel assumes $\omega_{it} = 0$. The fixed effect panel assumes ω_{it} is the sum of individual fixed effects and time fixed effects. The other three methods assume ω_{it} can be replaced by some control function which depends on a proxy either $\ln I$ or $\ln M$.

Table 8: Comparison of model performance by Cor(proxy, $\hat{\varepsilon}_{it}$)

Models	$\ln M$	$\ln I$
LASSO	0.066	0.073
Random Forests	0.084	0.074
Neural Networks	0.088	0.077
Pooled	0.258	0.095
Fixed Effects	0.244	0.090
Olley and Pakes (1996)	0.218	-
Levinsohn and Petrin (2003)	-	0.065
Ackerberg et al. (2015)	-	0.065
Ackerberg et al. (2015)	0.212	-

Notes. This table presents the correlation coefficient between the instrument variable, i.e., $\ln L$ or $\ln K$, and the estimated residuals, i.e., $\hat{\varepsilon}_{it}$. The correlation between $\ln I$ and $\hat{\varepsilon}_{it}$ is not applicable for the models in Olley and Pakes (1996) and Ackerberg et al. (2015), while the correlation between $\ln M$ and $\hat{\varepsilon}_{it}$ is not applicable for the models in Levinsohn and Petrin (2003) and Ackerberg et al. (2015), as both have already been utilized in their control function $h(\cdot)$.

Technical Appendix

A Proof for equation (2)

Model (1) can be rewritten as

$$y_{it} = f(\mathbf{x}_{it}) + \alpha_i + \gamma_i^\intercal \lambda_t + \varepsilon_{it}, \quad 1 \le i \le N, \quad 1 \le t \le T. \tag{A1}$$

Define the neighborhood sample points set $\mathcal{N}(\mathbf{x}, \nu) = \{(i,t) | \|\mathbf{x}_{it} - \mathbf{x}\|_2 < \nu\}$ with $\|\cdot\|_2$ being the Euclidean norm and ν some positive constant. As $N \to \infty, T \to \infty$, $\nu \to 0$ and the number of sample points in the set $\mathcal{N}(\mathbf{x}, \nu)$, i.e. $|\mathcal{N}(\mathbf{x}, \nu)| \to \infty$. By the first order Taylor expansion at the sample point $\mathbf{x}_{it} = \mathbf{x}$, we have

$$y_{it}^* = f(\mathbf{x}) + (\mathbf{x}_{it}^* - \mathbf{x})^\intercal \theta(\mathbf{x}) + R(\mathbf{x}_{it}^*, \mathbf{x}) + \alpha_i + \gamma_i^\intercal \lambda_t + \varepsilon_{it}^*, \quad (i,t) \in \mathcal{N}(\mathbf{x}, \nu), \tag{A2}$$

where $\theta(\mathbf{x})$ is the first order derivative of y with respect to $\mathbf{x}$, $R(\mathbf{x}_{it}^*, \mathbf{x}) = \frac{1}{2}(\mathbf{x}_{it}^* - \mathbf{x})\Omega(\mathbf{x})(\mathbf{x}_{it}^* - \mathbf{x})^\intercal + r(\mathbf{x}_{it}^*, \mathbf{x})$ with $\Omega(\mathbf{x}) = \partial f(\mathbf{x}) / \partial \mathbf{x} \partial \mathbf{x}^\intercal$, and $r(\mathbf{x}_{it}^*, \mathbf{x})$ being the remainder. Then, averaging (A2) over i at time t yields

$$\bar{y}_t^* = f(\mathbf{x}) + (\bar{\mathbf{x}}_t^* - \mathbf{x})^\intercal \theta(\mathbf{x}) + \bar{R}_t^* + \bar{\gamma}^\intercal \lambda_t + \bar{\varepsilon}_t^*, \quad (\cdot, t) \in \mathcal{N}(\mathbf{x}, \nu), \tag{A3}$$

where $\bar{y}_t^* = \frac{1}{|\mathcal{N}(\mathbf{x},\nu)|} \sum_{\{\cdot,t\} \in \mathcal{N}(\mathbf{x},\nu)} y_{it}^*$, $\bar{\mathbf{x}}_t^* = \frac{1}{|\mathcal{N}(\mathbf{x},\nu)|} \sum_{\{\cdot,t\} \in \mathcal{N}(\mathbf{x},\nu)} \mathbf{x}_{it}^*$, $\bar{R}_t^* = \frac{1}{|\mathcal{N}(\mathbf{x},\nu)|} \sum_{\{\cdot,t\} \in \mathcal{N}(\mathbf{x},\nu)} R(\mathbf{x}_{it}^*, \mathbf{x})$, $\bar{\gamma}^* = \frac{1}{|\mathcal{N}(\mathbf{x},\nu)|} \sum_{\{\cdot,t\} \in \mathcal{N}(\mathbf{x},\nu)} \gamma_i^*$, and $\bar{\varepsilon}_t^* = \frac{1}{|\mathcal{N}(\mathbf{x},\nu)|} \sum_{\{\cdot,t\} \in \mathcal{N}(\mathbf{x},\nu)} \varepsilon_{it}^*$. By condition (C4) in Section B, $f(\mathbf{x})$ has bounded second order derivatives so that $\Omega(\mathbf{x})$ is bounded. Meanwhile, $R(\mathbf{x}_{it}^*, \mathbf{x}) \to 0$ holds as $\{i,t\} \in \mathcal{N}(\mathbf{x}, \nu)$. Therefore, the order of $\bar{R}_t^*$ is $o_p(1)$. Moreover, the data $\{y_{it}, \mathbf{x}_{it}\}$ and factor loadings γ_i are stationary processes by conditions (C1) and (C2), we replace the above sample means of the neighborhood points of $\mathbf{x}$ by more efficient estimators, i.e., the sample means of the full sample $\bar{y}_t = \frac{1}{N} \sum_{i=1}^N y_{it}$, $\bar{\mathbf{x}}_t = \frac{1}{N} \sum_{i=1}^N \mathbf{x}_{it}$, $\bar{\gamma} = \frac{1}{N} \sum_{i=1}^N \gamma_i$, and $\bar{\varepsilon}_t = \frac{1}{N} \sum_{i=1}^N \varepsilon_{it}$, respectively. We observe $\bar{\varepsilon}_i = o_p(1)$ by the law of large number. Therefore, λ_t can be written as a linear combination of 1, $\bar{y}_t$ and $\bar{\mathbf{x}}_t$, i.e.,

$$\lambda_t = (\bar{\gamma}\bar{\gamma}^\intercal)^{-1} \bar{\gamma} \left(\bar{y}_t - f(\mathbf{x}) - (\bar{\mathbf{x}}_t - \mathbf{x})^\intercal \theta(\mathbf{x})\right) + o_p(1).$$

In the same manner, averaging (A2) over t for unit i yields

$$\bar{y}_i^* = f(\mathbf{x}) + (\bar{\mathbf{x}}_i^* - \mathbf{x})^\intercal \theta(\mathbf{x}) + \bar{R}_i + \alpha_i + \bar{\varepsilon}_i^*, \qquad (i,\cdot) \in \mathcal{N}(\mathbf{x},\nu), \tag{A4}$$

where $\bar{y}_i^* = \frac{1}{|\mathcal{N}(\mathbf{x},\nu)|}\sum_{\{i,\cdot\}\in\mathcal{N}(\mathbf{x},\nu)} y_{it}^*$, $\bar{\mathbf{x}}_i^* = \frac{1}{|\mathcal{N}(\mathbf{x},\nu)|}\sum_{\{i,\cdot\}\in\mathcal{N}(\mathbf{x},\nu)} \mathbf{x}_{it}^*$, $\bar{R}_i^* = \frac{1}{|\mathcal{N}(\mathbf{x},\nu)|}\sum_{\{i,\cdot\}\in\mathcal{N}(\mathbf{x},\nu)} R(\mathbf{x}_{it}^*, \mathbf{x})$, and $\bar{\varepsilon}_i^* = \frac{1}{|\mathcal{N}(\mathbf{x},\nu)|}\sum_{\{i,\cdot\}\in\mathcal{N}(\mathbf{x},\nu)} \varepsilon_{it}^*$. Here, we can similarly show $\bar{R}_i^* = o_p(1)$. And we replace the above sample means of the neighborhood points of $\mathbf{x}$ by more efficient estimators, i.e., the sample means of the full sample $\bar{y}_i = \frac{1}{T}\sum_{t=1}^T y_{it}$, $\bar{\mathbf{x}}_i = \frac{1}{T}\sum_{t=1}^T \mathbf{x}_{it}$, and $\bar{\varepsilon}_i = \frac{1}{T}\sum_{t=1}^T \varepsilon_{it}$.

We observe $\bar{\varepsilon}_i = o_p(1)$ by the law of large numbers. Therefore, α_i can be written as a linear combination of 1, $\bar{y}_i$ and $\bar{\mathbf{x}}_i$, i.e.,

$$\alpha_i = \bar{y}_i - f(\mathbf{x}) - (\bar{\mathbf{x}}_i - \mathbf{x})^\intercal \theta(\mathbf{x}) + o_p(1).$$

In sum, we can add cross-sectional averages ($\bar{y}_t$ and $\bar{\mathbf{x}}_t$) and time averages ($\bar{y}_i$ and $\bar{\mathbf{x}}_i$) as regressors to model (1) to account for the unobserved common factors and fixed effects, respectively. That is,

$$y_{it} = f(\mathbf{x}_{it}) + \beta^\intercal \mathbf{z}_{it} + e_{it}, \quad 1 \le i \le N, \quad 1 \le t \le T,$$

where $\mathbf{z}_{it} = (\bar{y}_t - \bar{y}, (\bar{\mathbf{x}}_t - \bar{\mathbf{x}})^\intercal, \bar{y}_i - \bar{y}, (\bar{\mathbf{x}}_i - \bar{\mathbf{x}})^\intercal)^\intercal$, and β is the corresponding $(2d+2)$-dimensional coefficient vector. We use the demeaned version here for identification purpose. The error term e_{it}, which consists of both the idiosyncratic term ε_{it} and the approximated error from the Taylor expansion, is a mean zero residual term and not correlated with $\mathbf{x}_{it}$ and $\mathbf{z}_{it}$.

B Proof of theorem 2:

We rewrite the model (2) as

$$y_{it} - \widehat{\beta}^\intercal \mathbf{z}_{it} = f(\mathbf{x}_{it}) - \left(\widehat{\beta} - \beta\right)^\intercal \mathbf{z}_{it} + e_{it}.$$

It follows $\widehat{\beta} - \beta = O_p((NT)^{-1/2})$ and $\mathbf{z}_{it} = O_p(1)$ that

$$y_{it} - \widehat{\beta}^\intercal \mathbf{z}_{it} = f(\mathbf{x}_{it}) + e_{it} + O_p((NT)^{-1/2}).$$

The conditional expectation estimator on both sides of above equation is

$$\widehat{f}(\mathbf{x}) = E\left(y_{it} - \widehat{\beta}^{\intercal}\mathbf{z}_{it}|\mathbf{x}\right) = f(\mathbf{x}) + O_p((NT)^c) + O_p((NT)^{-1/2}),$$

where $-1/2 \leq c < -1/4$ is the convergence rate for a given machine learning estimator. Therefore, $\widehat{f}(\mathbf{x}) \xrightarrow{p} f(\mathbf{x})$ as $N \to \infty$ and $T \to \infty$.