Deep Autoencoders for Nonlinear Factor Models: Theory and Applications[*]

Zhouyu Shen[†]
University of Chicago

Dacheng Xiu[‡]
University of Chicago and NBER

November 20, 2024

Abstract

Autoencoders are neural networks widely used in unsupervised learning for dimensionality reduction and feature extraction. This paper provides non-asymptotic guarantees for deep autoencoders within a nonlinear factor model, showing they can effectively extract latent components with errors that diminish with increasing dimensionality and sample size. The extracted factors converge to the true latent factors, up to a functional transformation. We extend these results to supervised autoencoders, supporting their use in factor-augmented prediction and structured matrix completion. Finally, we illustrate the practical value of autoencoders in macroeconomic forecasting, asset return prediction, and noise reduction for causal analysis.

Key words: Latent Factors, PCA, Neural Networks, Nonparametrics, Supervised Autoencoders, Factor-Augmented Prediction, Matrix Completion

[*] We benefited tremendously from discussions with conference participants at the Stevanovich Center at the University of Chicago.

[†] Address: 5807 S Woodlawn Avenue, Chicago, IL 60637 USA. Email: zshen10@chicagobooth.edu.

[‡] Address: 5807 S Woodlawn Avenue, Chicago, IL 60637, USA. Email: dacheng.xiu@chicagobooth.edu.

1 Introduction

Autoencoders (AEs) are specialized neural network (NN) models designed to replicate their inputs at their outputs, playing a fundamental role in unsupervised machine learning. These models, which gained traction since the 1980s (e.g., LeCun (1987)), have a canonical architecture with two main components: an encoder, which compresses the inputs into a lower-dimensional representation known as features, codes, embeddings, or factors, and a decoder, which reconstructs the inputs from this compressed form.

We are particularly drawn to AEs due to their close connection with linear factor models and their capability in conducting nonlinear dimensionality reduction. It is well-documented (e.g., Baldi and Hornik (1989)) that a single-layer AE with linear activations is equivalent to Principal Component Analysis (PCA), with the number of neurons in that layer corresponding to the number of components. PCA has been extensively studied, both theoretically and empirically, demonstrating its effectiveness in estimating linear factor models widely used in macroeconomics and finance (Stock and Watson, 1999, Bai and Ng, 2002, Chamberlain and Rothschild, 1983, Connor and Korajczyk, 1986). The success of PCA motivates extending theoretical guarantees to AEs, particularly for data-generating processes (DGPs) with nonlinear low-dimensional structures.

Despite the broad application of AEs in machine learning, there has been scant research providing theoretical justifications. To address this gap, our paper positions AEs as estimators for nonlinear factor models, setting the stage for a comprehensive investigation into their statistical properties. Within this framework, we explore several pivotal questions to enhance our understanding of deep and nonlinear AEs. These questions include whether AEs can effectively identify and extract "commonalities" in the inputs and, if so, what are the associated statistical error bounds. We also examine how architectural parameters of AEs, such as depth, width, and the number of neurons, affect AEs' statistical performance. Furthermore, we investigate whether AEs can recover hidden low-dimensional embeddings inherent in nonlinear factor models. Addressing these inquiries from a theoretical standpoint paves the way for novel applications of this powerful tool in various fields of economics.

Our contributions are threefold. First, on model, we introduce a novel architecture for AEs, termed Disjoint Output AEs. This architecture retains the standard fully connected encoder, with its output forming the bottleneck layer, serving as the intermediary between the encoder and decoder. The bottleneck layer consists of multiple neurons, the number of which corresponds to the dimensionality of the embeddings, akin to the number of factors in a linear factor model. The key innovation lies in the decoder network, which employs separate networks to map all embeddings to each specific output, creating a sparse-link alternative to the conventional fully connected decoder. The imposed sparsity in weight parameters preserves the AE's capacity to approximate a broad range of nonlinear factor model DGPs, yielding desirable approximation errors—specifically, the difference between the nonlinear function and its optimal approximation within the AE class. Additionally, estimating fewer weight parameters accelerates training and reduces the estimation error, defined as the difference between this optimal approximation and the estimated AE.

Second, on theory, we establish non-asymptotic guarantees for AE's convergence rates in approximating the common components of input data. Our analysis accommodates both increasing depth of AE architecture and width of any layer—including the embedding dimension. We demonstrate that AE's convergence rate is driven by two key components: the approximation and estimation errors in recovering latent factors via the encoder, and the corresponding errors in reconstructing the input through the decoder. The encoder's estimation error diminishes with increasing dimensionality, echoing the "blessings of dimensionality" observed in linear factor models. Importantly, as long as the bottleneck layer's width remains bounded, the AE maintains an optimal convergence rate even if the true dimensionality of the factors in the DGP is exceeded. For approximation error, we analyze two scenarios: in the first, the encoder is over-parameterized, achieving zero training error but risking out-of-sample divergence; in the second, the encoder is appropriately parameterized, achieving convergent approximation errors in both in-sample and out-of-sample settings, subject to a pervasiveness condition analogous to that in linear factor models.

The second component driving error concerns the approximation and estimation errors within the decoder, which diminishes as sample size increases. This component resembles the estimation error in factor loadings in linear regression models, though here the "loadings" are high-dimensional nonlinear functions. Importantly, we demonstrate that the decoder's error achieves the optimal nonparametric regression rate, as if the embeddings were directly observable. Our theoretical results further extend to the convergence rate of the factors themselves, which are identifiable up to invertible functional transformations. Consequently, AEs can recover these latent factors up to an unknown nonlinear transformation, with a convergence rate consistent with the result described above.

Third, we extend our results to supervised AEs (SAEs), demonstrating their applicability in factor-augmented regressions and structured matrix completion, thereby broadening the AE framework's relevance in economics. We further illustrate the empirical potential of AEs and SAEs through three distinct applications: forecasting key macroeconomic indicators such as industrial production, inflation, unemployment, and non-farm payrolls; predicting the cross-section of factor returns; and conducting causal analysis with corrupted covariates. In all three cases, AEs significantly outperform PCA due to their superior ability to extract nonlinear factors.

Our work contributes to a growing body of research on nonlinear factor models, which, despite early foundational studies by Etezadi-Amoli and McDonald (1983) and Kenny and Judd (1984), has faced substantial challenges due to the inherent complexity of these models. Building on the insights of Griebel and Harbrecht (2014) and Xu (2017), who show that certain nonlinear factor models could be effectively approximated by low-rank matrices, further research by Agarwal et al. (2021) and Freeman and Weidner (2023) demonstrate the robustness of PCA in extracting the common components of data characterized by such nonlinear structures. More recently, Feng (2023) has extended this line of inquiry to a broader class of factor models initially proposed by Amemiya and Yalcin (2001), introducing a local PCA approach inspired by local regression techniques. This approach achieves a convergence rate consistent with our theoretical results. However, while it is effective at noise reduction, it falls short of constructing the underlying factors as the locally estimated components lack integration into a cohesive time series of factors. By contrast, the encoder of AEs provides a global solution for factor construction, integrating information across the entire dataset to form cohesive factors. Additionally, AEs are versatile, extending to various NN architectures that can capture complex data types, including text and images, making them broadly applicable across diverse domains.

Our paper adds to the growing body of literature on the theoretical properties of deep neural networks (DNNs). This literature builds on the early work of Barron (1993) and Chen and White (1999), which establish and refine nonparametric approximation rates for single-layer sigmoid neural networks. Chen and Shen (1998) provide a general theory on the convergence rate of sieve extremum estimates for time series data, incorporating single-layer sigmoid neural networks as a special case. Mei et al. (2018) and Mei et al. (2019) apply mean field theory to study the behavior of single-layer neural networks with stochastic gradient descent. Yarotsky (2017) explores the optimal approximation errors in deep ReLU networks for a class of smooth functions. Building on this, Schmidt-Hieber (2020) and Farrell et al. (2021) derive the optimal error rate for sparse DNNs, while Kohler and Langer (2021) demonstrate that fully connected DNNs can also achieve this optimal rate. Additionally, works by Bauer and Kohler (2019), Nakada and Imaizumi (2020), and Jiao et al. (2023)

highlight the potential of DNNs to overcome the curse of dimensionality, particularly when the data exhibits intrinsic low-dimensionality or when the target function has a compositional structure. Unlike these studies, which focus on supervised learning problems for feedforward DNNs, our work investigates the theoretical properties of a specific class of DNNs—AEs—in unsupervised learning settings.

This paper is organized as follows: Section 2 presents the nonlinear factor model and introduces Disjoint Output AEs. Section 3 presents the main theoretical results on the statistical properties of these AEs and extends the analysis to SAEs. Section 4 conducts simulations to validate our theoretical predictions. Section 5 discusses empirical results that illustrate the practical applications of AEs. The appendix provides additional comparisons with Kernel PCA and includes detailed mathematical proofs.

Notation: We denote the set of positive integers by $\mathbb{N}_+$. Furthermore, we represent the set that includes both zero and all positive integers, $\{0\} \cup \mathbb{N}_+$, by $\mathbb{N}_0$. For $n \in \mathbb{N}_+$, we use $[n]$ to denote the set $\{1, \ldots, n\}$. For a vector $x = (x_1, \ldots, x_d)^\top$, as usual, we define $\|x\|_p = (\sum_{i=1}^d |x_i|^p)^{\frac{1}{p}}$ and $\|x\|_\infty = \max_i |x_i|$. Additionally, let $\mathcal{P}_n^d$ be the linear span of all monomials of the form $\prod_{k=1}^d x_k^{r_k}$ for some $r_1, \ldots, r_d \in \mathbb{N}_+$, where $r_1 + \cdots + r_d = n$. For a matrix $A \in \mathbb{R}^{n \times m}$, we use $\|A\|$, $\|A\|_\infty$, $\|A\|_F$ and $|A|$ to represent its spectral norm, the maximum absolute value of its entries, its Frobenius norm, and its determinant if it is a square matrix, respectively. Given a function $f : \mathbb{R}^n \to \mathbb{R}^m$, we define $\|f\|_\infty = \sup_{x \in \mathcal{D}} \|f(x)\|_\infty$, where $\mathcal{D}$ represents the domain of f. We say $f \in \mathcal{C}^\beta$ if it has β continuous derivatives on its domain. Furthermore, we define the norm $\|\cdot\|_{\mathcal{C}^\beta}$ of the smooth function space $\mathcal{C}^\beta$ by $\|f\|_{\mathcal{C}^\beta} := \max\left\{\|D^\alpha f\|_\infty : \|\alpha\|_1 \leq \beta, \alpha \in \mathbb{N}_0^d\right\}$. If f is differentiable on its domain, we express its Jacobian matrix at x as $J_f(x)$. For a set $\mathcal{F}$, we use $|\mathcal{F}|$ to indicate the number of elements within the set. We use the notation $x_n \lesssim y_n$ when there exists a constant C such that $x_n \leq C y_n$ holds for sufficiently large n. If $x_n \lesssim y_n$ and $y_n \lesssim x_n$, we write $x_n \asymp y_n$ for short. For two random variables X and Y, we write $X \leq_d Y$ if Y stochastically dominates X and $X \overset{d}{=} Y$ if X has the same distribution as Y. For a sub-Gaussian variable X, its sub-Gaussian norm is defined as $\|X\|_{\psi_2} = \inf\{c > 0 : \mathrm{E}[\exp(X^2/c^2)] \leq 2\}$.

2 Model Setup

We begin by introducing the underlying DGP that will be used to characterize the statistical properties of AEs.

2.1 Nonlinear Factor Model

We model the observed data, X_{it}, for $i = 1, 2, \ldots, N$ and $t = 1, 2, \ldots, T$, based on a framework initially proposed by Amemiya and Yalcin (2001):

$$X_{it} = X_{it}^\star + U_{it} = \varphi_i^\star(F_t^\star) + U_{it}, \tag{1}$$

where the common component, denoted by $X_{it}^\star$, is governed by a potentially nonlinear factor loading function $\varphi_i^\star(\cdot)$ of a K-dimensional vector of unknown factors, $F_t^\star$. The term U_{it} represents the noise component. We also use the notation X_t, $X_t^\star$, $\varphi^\star(F_t^\star)$, U_t to denote $N \times 1$ vectors containing the values for each entry of these variables at time t. Similarly, X, $X^\star$, and U represent their $N \times T$ matrix versions, and $F^\star$ denotes the $K \times T$ matrix of factors.

This model nests the linear factor model as a special case, where $\varphi_i^\star(x) = \Lambda_i^\top x$ is a linear function, and Λ_i is the $N \times 1$ column vector corresponding to the ith row of a $K \times N$ factor exposure matrix, Λ. Other examples include the polynomial factor model, where $\varphi_i^\star(x)$ is a multivariate polynomial function of x, as studied by McDonald (1962) and Kenny and Judd (1984), and the additive nonlinear factor model, where $\varphi_i^\star(x) = \Lambda_i^\top \varphi(x)$ for some nonlinear function $\varphi(\cdot)$, as examined by Zhu and Lee (1999). This framework also encompasses generalized linear latent variable models, where $\varphi_i^\star(x) = \varphi(\Lambda_i^\top x)$, as explored by Moustaki and Knott (2000), Skrondal and Rabe-Hesketh (2004), Huber et al. (2004), Chen et al. (2017), Wei et al. (2021), and Wang (2022). Finally, it includes the generalized factor model, where $\varphi_i^\star(x) = \varphi(\Lambda_i, x)$, as explored by Agarwal et al. (2021) and Freeman and Weidner (2023).

Subsequently, we make assumptions regarding the bounds on various norms of U_t and $F_t^\star$, as well as smoothness conditions on $\varphi_i^\star(\cdot)$.

Assumption 1 (Boundedness)**.** *There exists a constant $B > 0$ such that $\max_{1 \le t \le T} \|F_t^\star\|_\infty \le B$ holds almost surely. Conditional on $F^\star$, $\operatorname{vec}(U) \overset{d}{=} \Sigma^{1/2} \operatorname{vec}(Z)$, where $Z \in \mathbb{R}^{N \times T}$ consists of independent sub-Gaussian random variables with sub-Gaussian norm bounded by σ_z^2. Moreover, the $NT \times NT$ matrix, Σ, is positive semi-definite with bounded spectral norm.*

The almost sure bound on the support of F_t may seem stronger than what is typically assumed in the literature on linear factor models. However, this assumption is common in nonparametric literature (see, e.g., Chen and White (1999)) and is particularly important for developing theoretical results on NNs (e.g., Farrell et al. (2021)). In practice, this assumption is nearly harmless, though it does exclude distributions supported on the entire real line.

The assumption on Σ rules out strong time-series and cross-sectional dependence among entries of U. In Lemma 2 of the appendix, we show that this assumption holds if $U = \Sigma_1^{1/2} Z \Sigma_2^{1/2}$, where Σ_1 and Σ_2 are positive semi-definite matrices with bounded spectral norms. A similar assumption has been adopted by Onatski (2005) and Ahn and Horenstein (2013) in their analysis of linear factor models.

We then impose a smoothness assumption on $\varphi_i^\star(\cdot)$. As is standard in the nonparametric literature (see, e.g., Stone (1982)), the smoothness of the true underlying function influences how accurately it can be approximated by neural nets.

Definition 1 (Hölder Ball). *Let $\beta > 0$ and Ω be a subset of some Euclidean space, the Hölder ball of functions $\mathcal{H}^\beta(\Omega, B)$ is defined as*

$$\mathcal{H}^\beta(\Omega, B) = \left\{ f : \Omega \to \mathbb{R}, \max_{\alpha, |\alpha| \le \lfloor \beta \rfloor} \sup_{x \in \Omega} |D^\alpha f(x)| + \max_{\alpha : \|\alpha\|_1 = \lfloor \beta \rfloor} \sup_{\substack{x, x' \in \Omega \\ x \neq x'}} \frac{|D^\alpha f(x) - D^\alpha f(x')|}{\|x - x'\|^{\beta - \lfloor \beta \rfloor}} \le B \right\},$$

where $\lfloor \beta \rfloor$ represents the largest integer strictly smaller than β.

Assumption 2 (Smoothness). *There exist $\beta \in \mathbb{N}_+$ and an open set Ω containing $[-B, B]^K$, such that $\varphi_i^\star(\cdot)$ lies in the Hölder ball $\mathcal{H}^\beta(\Omega, B)$, where B is given by Assumption 1.*

If the factors $F_t^\star$ were observable, estimating the unknown function $\varphi_i^\star(\cdot)$ would reduce to a standard (supervised) nonparametric regression problem, which could be solved using methods such as sieve estimators. However, our problem is more involved due to its unsupervised and nonparametric nature, requiring the development of a distinct estimator. We now proceed to construct AEs, which will serve as our estimator for nonlinear factor models.

2.2 Autoencoders and Their Architecture

To lay the foundation for constructing AEs, we first formally build the necessary components of NNs. We begin with the rectified linear unit (ReLU) activation function, defined as $\sigma(x) = \max(x, 0)$. Given $v = (v_1, \ldots, v_r) \in \mathbb{R}^r$, we define the shifted activation function $\sigma_v : \mathbb{R}^r \to \mathbb{R}^r$ as follows:[1]

$$\sigma_v \begin{pmatrix} y_1 \\ \vdots \\ y_r \end{pmatrix} = \begin{pmatrix} \sigma(y_1 - v_1) \\ \vdots \\ \sigma(y_r - v_r) \end{pmatrix}.$$

[1] When possible, we omit the brackets in the shifted activation function $\sigma_{v_i}(\cdot)$.

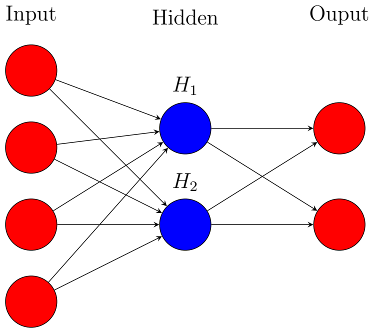

Figure 1: Illustration of Neural Network

Note: This graph illustrates a fully connected neural network with architecture parameters $d = 1$, $w = 2$, and $n = (4, 2, 2)$. The input and output neurons are shown in red, while the neurons in the hidden layer are displayed in blue.

An NN's architecture, denoted as (d, w, n), consists of a positive integer d, which represents the number of hidden layers (depth), and a width vector $n = (n_0, \ldots, n_{d+1}) \in \mathbb{N}^{d+2}$, where $n_i \leq w$ for $i = 1, \ldots, d$. Here n_0 and n_{d+1} correspond to the input and output dimensions, respectively. An NN with architecture (d, w, n) can be expressed as a function of the form:

$$f : \mathbb{R}^{n_0} \to \mathbb{R}^{n_{d+1}}, x \to f(x) = W_d \sigma_{v_d} W_{d-1} \sigma_{v_{d-1}} \cdots W_1 \sigma_{v_1} W_0 x, \tag{2}$$

where W_i denotes a $n_{i+1} \times n_i$ weight matrix, σ_{v_i} is the shifted activation function introduced earlier, with $v_i \in \mathbb{R}^{n_i}$ representing a shift vector. For instance, Figure 1 illustrates an NN with architecture $(1, 2, (4, 2, 2))$. We use this notation to denote a fully connected NN; however, if certain connections are omitted, we simply set the corresponding weights in W_i's to zero. We use the term DNNs to refer to NNs with a potentially diverging depth.

We adopt a setup similar to Schmidt-Hieber (2020), where the parameters of a DNN are bounded to ensure that the network remains well-behaved. Specifically, we define the function space of DNNs as:[2]

$$\mathcal{F}_{n_0}^{n_{d+1}}(d, w, C, B) := \left\{ f \text{ of the form (2)} : \max_{j=0,\ldots,d} \|W_j\|_\infty \vee |v_j|_\infty \leq C, \ \|f\|_\infty \leq B \right\},$$

where B is specified by Assumption 1. The first condition within the brackets ensures that all weight matrices and shift vectors are bounded by C. In Schmidt-Hieber (2020), C is set to a fixed constant (1), whereas in our study, we let $C = T^{5\beta+5}$, allowing it to scale with the

[2] For simplicity, we omit the dependence of $\mathcal{F}_{n_0}^{n_{d+1}}$ on the width vector n here.

sample size. This adjustment permits the DNNs to achieve a smaller approximation error without adversely impacting the estimation error.

The second condition enforces a uniform bound on the NN f, regardless of the sample size or the scale of its architecture parameters. Effectively, this means we consider only DNNs with bounded outputs when used as estimators. Notably, this aligns with the boundedness assumption on the true function $\varphi_i^\star(\cdot)$ in the DGP, as implied from Assumption 2. Practically, this constraint serves not as a stringent restriction but rather as a mild form of regularization.

Having introduced the basic architecture of a DNN, we are now ready to define a special class of DNNs known as AEs. The primary objective of an AE is to reconstruct the input from their compressed form, thereby facilitating dimensionality reduction.

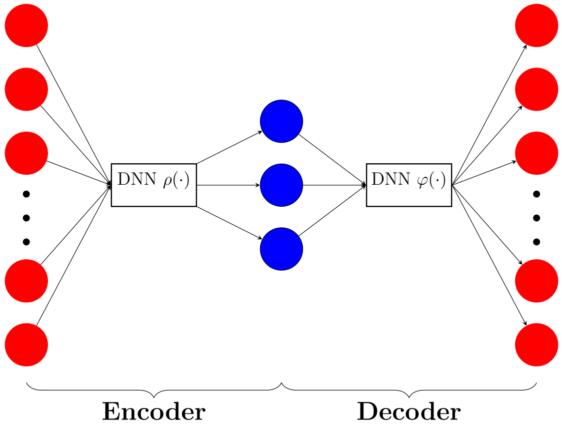

Figure 2: Illustration of Canonical Autoencoder Architecture

Note: This graph illustrates a canonical autoencoder architecture, where the input layer first processes the data, passing it through a sequence of hidden layers structured as a DNN that compresses the information down to a central bottleneck layer—the narrowest point in the architecture. This bottleneck layer, containing three neurons shown in blue, captures the core features in a reduced-dimensional form. The network then reconstructs the input as closely as possible through additional DNN layers, producing an output layer with the same dimension as the input.

Specifically, given an input x, an AE first applies an “encoder,” a DNN denoted by $\rho(\cdot)$, which maps the input into a low-dimensional “bottleneck,” represented by a small number of neurons in a hidden layer—this being the narrowest layer of the AE. The encoded data at the bottleneck layer is then passed through a second DNN, called the “decoder,” denoted by $\varphi(\cdot)$, which reconstructs an approximation of the original input x at the output. Figure 2 illustrates the canonical architecture of a vanilla AE.

Unlike a DNN, which is often used for supervised learning to predict another target variable, AEs have output neurons of the same size as their input, aiming to reconstruct the input itself. This makes AEs an unsupervised learning tool, as no other variables besides the input are involved.

In this paper, we shift our focus from the canonical form of AE to what we refer to as a disjoint output AE, with its architecture depicted in Figure 3. This specialized class of AEs retains the same encoder design as the fully connected AE but features a unique decoder structure: a separate DNN for each output neuron, mapping from the bottleneck to reconstruct the corresponding output.

More specifically, the architecture of the disjoint output AE consists of a single DNN as the encoder and multiple separate DNNs collectively as the decoder. The encoder, $\rho(\cdot) \in \mathcal{F}_N^{K_1}(d_1, w_1, T^{5\beta+5}, B)$, produces a K_1-dimensional bottleneck layer. For each of the N outputs, a separate DNN is employed, where each $\varphi_i(\cdot) \in \mathcal{F}_{K_1}^1(d_2, w_2, T^{5\beta+5}, B)$ is specific to the ith output neuron for $i \in [N]$. As a result, for an input x, the i^{th}-output of the disjoint output AE is given by $\varphi_i \circ \rho(x)$. Formally, we define the disjoint output AE function class as follows:

$$\mathcal{F}_{AE}^{K_1} := \Big\{(\varphi_1, \ldots, \varphi_N) \circ \rho : \rho \in \mathcal{F}_N^{K_1}(d_1, w_1, T^{5\beta+5}, B), \varphi_i \in \mathcal{F}_{K_1}^1(d_2, w_2, T^{5\beta+5}, B), i \in [N]\Big\}. \tag{3}$$

Consider the canonical form of an AE, illustrated in Figure 2, $\varphi \circ \rho(x)$, where the encoder is $\rho(\cdot) \in \mathcal{F}_N^{K_1}(d_1, w_1, T^{5\beta+5}, B)$, and the decoder $\varphi(\cdot) \in \mathcal{F}_{K_1}^N(d_2, Nw_2, T^{5\beta+5}, B)$ is fully connected. The total number of weight parameters in the decoder is of order $O((Nw_2)^2 d_2)$. In contrast, the decoder $(\varphi_1, \ldots, \varphi_N)$ of the disjoint output AE in $\mathcal{F}_{AE}^{K_1}$ contains at most $N \times O((w_2)^2 d_2)$ parameters. This disjoint output architecture introduces sparsity in the decoder's weight parameters, reducing the number of weights by a factor of N, compared to the canonical AE. While it is possible to encourage sparsity on a fully connected decoder through ℓ_1-regularization during training, the disjoint AE naturally achieves sparsity without the need for such regularization.

As we will explain later, this disjoint output architecture delivers strong approximation performance, being versatile enough to capture the nonlinear, low-dimensional structure in the inputs, even with a substantial reduction in the number of weights parameters. At the same time, it achieves desirable statistical estimation properties by reducing the number of neurons, thereby effectively controlling model complexity.

The final step in constructing an AE is specifying its loss function. The AE's ability to perform dimensionality reduction comes from training it to reconstruct the input data

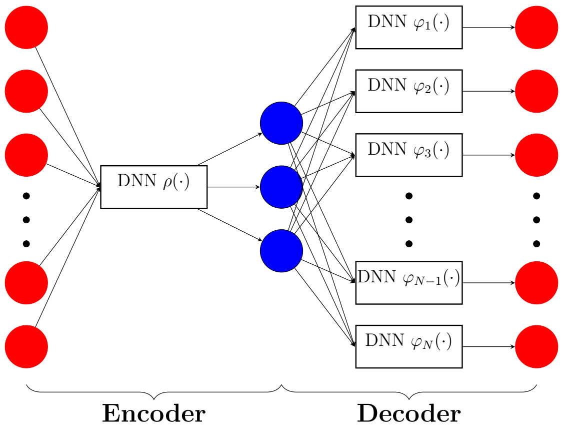

Figure 3: Illustration of Disjoint Output Autoencoder

Note: This graph illustrates a disjoint output autoencoder architecture. The encoder retains the same structure as the canonical autoencoder's encoder shown in Figure 2. The decoder, however, consists of multiple separate DNNs, each mapping the bottleneck layer to a single output.

accurately. This is achieved by minimizing the following loss function:[3]

$$(\widehat{\varphi}_1, \ldots, \widehat{\varphi}_N) \circ \widehat{\rho} = \underset{(\varphi_1,\ldots,\varphi_N)\circ\rho \in \mathcal{F}_{AE}^{K_1}}{\arg\min} \sum_{t=1}^{T} \sum_{i=1}^{N} \left(\varphi_i(\rho(X_t)) - X_{it}\right)^2 . \tag{4}$$

This optimization problem is highly non-convex, necessitating the use of more sophisticated algorithms for solving it. We will explore the optimization perspective in greater detail in the simulation and empirical analysis sections.

Before training an AE, it is necessary to select a key hyperparameter: the number of neurons in the bottleneck layer, K_1. The overall network architecture must also be designed, including decisions about its depth and width. These crucial considerations will be discussed after we present the theoretical analysis, which we turn to now.

3 Main Theoretical Results

In this section, we present a non-asymptotic analysis of the statistical properties of the disjoint output AEs. Our analysis begins with the reconstructed data, $\widehat{X}_{it} := \widehat{\varphi}_i(\widehat{\rho}(X_t))$.

[3] In a recent study, Liu et al. (2024) study the convergence rate of AEs when the common components reside on a certain manifold. Their DGP differs from ours. Moreover, they assume the common components are observable and use them to train an AE, which is not feasible in practical applications. Additionally, they study AEs with fully-connected decoder and achieve a rate of the order $N^2T^{-\frac{2}{K+2}}$, which is slower than ours.

where $\widehat{\rho}(\cdot)$ and $\widehat{\varphi}_i(\cdot)$ for $1 \leq i \leq N$ are defined in equation (4).

3.1 Recovery of the Common Components

Naturally, the population counterpart of $\widehat{X}_{it}$ is $X^\star_{it} = \varphi^\star_i(F^\star_t)$ in the DGP, as given in (1). Consequently, we analyze the difference between these quantities, aggregated across the entire panel: $\frac{1}{NT}\sum_{t=1}^{T}\sum_{i=1}^{N}\left(\widehat{X}_{it} - X^\star_{it}\right)^2$.

Theorem 1. *Consider the AE class $\mathcal{F}^{K_1}_{AE}$ with $K \leq K_1 \leq \min(w_1, w_2)$, $d_2 \asymp \log(T)$, and $w_2 \asymp T^{\frac{K}{2(2\beta+K)}}$. Suppose Assumptions 1-2 hold. Then, with probability at least $1 - C\exp(-cT)$, for $\min(N,T)$ large enough, we have:*

$$\frac{1}{NT}\sum_{t=1}^{T}\sum_{i=1}^{N}\left(\widehat{X}_{it} - X^\star_{it}\right)^2 \lesssim \left(T^{-\frac{2\beta}{2\beta+K}} + N^{-1}K_1 + T^{-1}\inf_{\rho\in\mathcal{F}^K_N}\sum_{t=1}^{T}\|\rho(X_t) - F^\star_t\|^2\right)\log^4(T),$$

where c and C are constants independent of N, T, and $\mathcal{F}^K_N := \mathcal{F}^K_N(d_1, w_1, T^{5\beta+5}, B)$.

We first discuss the assumptions. The conditions require that $K_1 \geq K$, meaning the number of neurons in the bottleneck layer must be at least as large as the number of factors specified in the DGP. However, there is some ambiguity regarding the exact number of factors in a nonlinear factor model, which we will address in greater detail in Section **??**. The requirement $K_1 \leq \min(w_1, w_2)$ ensures that the bottleneck layer is among the narrowest in the network, allowing for ties. On the decoder side, both d_2 and w_2 must increase with the sample size, ensuring that the network remains sufficiently deep and wide. Moreover, the conditions only necessitate that N and T be greater than a certain constant threshold. Given this, the non-asymptotic results naturally extend to asymptotic results as N and T diverge.

As is standard in nonparametric analysis, this result hinges on appropriately selecting the dimension of the bottleneck layer, K_1, and the width of the decoder, w_2, based on the smoothness parameter β and the number of factors K in the true DGP. In practice, these values are typically unknown, so conservative choices are often made. This approach assumes that the true functions are smoother than presumed and intentionally uses a relatively large number of factors—more than what economic theory or economists typically suggest or believe.

Next, we make a few observations regarding the result on the error bound. There are three terms on the right-hand side of the inequality. The first term in the error, $T^{-\frac{2\beta}{2\beta+K}}$, matches the minimax rate in nonparametric regression of X_t on $F^\star_t$, as if these factors were known. For a nonparametric supervised learning task, this rate can be achieved by estimators other than NNs; examples include spline methods by Speckman (1985) or sieve estimators by Newey (1997) and Chen and Shen (1998), among others.

The second term, $N^{-1}K_1$, arises from the estimation error associated with the unknown factors. Intuitively, since there are TK_1 unknown values to estimate and NT observed values in total, the ratio of unknowns per observation results in this rate. Our assumptions permit the number of neurons in the bottleneck layer to increase, and the result shows that this error term scales linearly with the number of neurons, impacting the overall convergence rate at the order of N^{-1}. This implies that selecting more factors than necessary does not negatively impact the convergence rate, highlighting the model's robustness to overestimating the number of factors.

The third term is associated with the error from approximating the unknown factors using the encoder. An important observation is that the estimation error for the encoder does not impact the overall convergence rate—only its approximation error is relevant. In light of this, one can use an over-parameterized encoder such that $\inf_{\rho \in \mathcal{F}_N^K} \sum_{t=1}^T \|\rho(X_t) - F_t^\star\|^2 = 0$, eliminating the approximation error and thereby improving the overall convergence rate based on in-sample criteria.

However, an over-parameterized encoder may lead to poor expected out-of-sample performance, where both estimation and approximation errors can become substantial.[4] A meaningful approach is to carefully balance these two error sources out-of-sample while designing an effective encoder, even if it results in larger approximation error in-sample. To achieve this, we need a more explicit characterization of the approximation error, which necessitates imposing restrictions on the DGPs we consider. Furthermore, we propose a potentially sparse encoder component to help balance the impact of estimation error and approximation error on out-of-sample performance.

Assumption 3 (Pervasiveness)**.** *There exist a matrix $W^\star \in \mathbb{R}^{K \times N}$ and a function $\rho^\star$ whose domain is the image of the mapping $W^\star\varphi^\star$, denoted as $W^\star\varphi^\star([-B, B]^K)$. These satisfy $\rho^\star \in \mathcal{H}^\beta(W^\star\varphi^\star([-B, B]^K), B)$, $\|W^\star\|_\infty \lesssim L^{-1}$, $\|W^\star\|_0 \asymp L$ for some diverging positive integer L, and*

[4] Recent literature indicates that under certain conditions on the data, overfitting can be benign; see, for example, Bartlett et al. (2019), Tsigler and Bartlett (2024), and Hastie et al. (2022). However, these analyses are mainly focused on linear settings for supervised learning problems, and a comprehensive theoretical framework for NNs remains unavailable. We leave the investigation of benign overfitting in unsupervised learning problems for future research.

$$\sup_{x\in[-B,B]^K} \|\rho^\star(W^\star\varphi^\star(x)) - x\|^2 \lesssim L^{-1}. \tag{5}$$

This assumption generalizes the pervasiveness assumption commonly used in linear factor models, as seen in Bai (2003). It allows for an approximate representation of the underlying factors using the input data. The parameter L quantifies the pervasiveness of the factors, representing the number of individuals in X who have a non-trivial exposure to these factors.

Specifically, consider the special case of a linear factor model, where $\varphi^\star(x) = \Lambda x$ with bounded loadings, $\|\Lambda\|_\infty \lesssim 1$. If $\|\Lambda\|_0 \asymp L$ —indicating that each factor influences approximately L variables—we can set $W^\star = L^{-1}\Lambda^\top$ and define $\rho^\star(x) = L(\Lambda^\top\Lambda)^{-1}x$. With these choices, Assumption 3 is satisfied.

Intuitively, although the sparsity assumption weakens the strength of the factors, leading to a greater approximation error, it can also reduce the estimation error. This trade-off has the potential to enhance out-of-sample performance, as we will illustrate in the next theorem.

Theorem 2. *Suppose that Assumptions 1-3 hold, with $K \le K_1 \le \min(w_1, w_2)$, $d_1 \asymp d_2 \asymp \log(T)$, and $w_1 \asymp w_2 \asymp T^{\frac{K}{2(2\beta+K)}}$. Additionally, we assume total number of weight parameters in the encoder is asymptotically bounded by $L + T^{\frac{K}{2\beta+K}}\log T$, and that $\log\max(N,T) = o(L)$. Then, with probability at least $1 - C\exp(-cT) - C\exp(-cL)$, as $\min(N,T)$ becomes sufficiently large, we have:*

$$\frac{1}{NT}\sum_{t=1}^{T}\sum_{i=1}^{N}\left(\widehat{X}_{it} - \varphi_i^\star(F_t^\star)\right)^2 \lesssim (N^{-1}K_1 + T^{-\frac{2\beta}{2\beta+K}} + L^{-1})\log^4(T),$$

where c and C are constants independent of N and T. Moreover, when the data is i.i.d., for a new data point X_{T+1}, we have

$$\frac{1}{N}\sum_{i=1}^{N}\mathbb{E}\left(\widehat{\varphi}_i \circ \widehat{\rho}(X_{T+1}) - X_{T+1}^\star\right)^2 \lesssim (N^{-1}K_1 + T^{-\frac{2\beta}{2\beta+K}} + L^{-1} + T^{-1}L)\log^4(NT).$$

The assumptions in Theorem 2 impose additional restrictions on the encoder compared to those in Theorem 1. Specifically, the depth and width of the encoder are of the same order as those of the decoder, both increasing with the sample size and influenced by the smoothness and number of factors in the DGP. Furthermore, the encoder network cannot contain an excessive number of weight parameters. These constraints simplify the conclusion of Theorem 1, where the approximation error of the decoder is of order L^{-1}, and the estimation error remains negligible for in-sample loss.

Consider the special case of a linear factor model: when $\varphi^\star(\cdot)$ is a linear function, $\beta = \infty$, K is finite, and the pervasiveness assumption holds with $L = N$. We analyze a single-layer AE with a linear activation function. In this case, the total number of parameters in the encoder, $K_1 N$, satisfies the imposed assumptions. Consequently, the overall convergence rate simplifies to $N^{-1} + T^{-1}$ (up to logarithmic terms), matching the rate established by Bai (2003).

The first result indicates that increased sparsity always hurts in-sample performance by raising the approximation error, which is on the order of L^{-1}. However, it can improve out-of-sample performance by reducing the estimation error. The second result of Theorem 2 shows that, for i.i.d. data, the expected out-of-sample loss includes both the aforementioned errors in our in-sample result and an additional term, $T^{-1}L$, arising from the estimation error in the encoder. The optimal rate is achieved when $L \asymp T^{1/2}$, resulting in an out-of-sample error that converges at the rate $(N^{-1}K_1 + T^{-\frac{2\beta}{2\beta+K}} + T^{-1/2}) \log^4(NT)$.

That said, implementing ℓ_0-regularization to achieve sparsity is challenging for NNs, as noted in previous work (e.g., Schmidt-Hieber (2020) and Farrell et al. (2021)). Consequently, while achieving the desirable rate of $T^{-1/2}$ is theoretically possible, it may not be feasible in practical applications.

3.2 Recovery of the Factors

In this section, we provide the theoretical justification for the potential of AEs to recover underlying factors in a nonlinear DGP. An important motivation for using AEs stems from dimensionality reduction, which aims to extract a lower-dimensional set of features that effectively capture the underlying structure of the data. By reducing the dimensionality, we can highlight the most informative aspects of the data, making patterns more interpretable and facilitating downstream tasks such as clustering, visualization, and forecasting.

We begin by noting that for any injective mapping $\mu : \mathbb{R}^K \to \mathbb{R}^K$, the DGP in equation (1) can be equivalently expressed as

$$X_{it} = \varphi_i^\star \circ \mu^{-1} \circ \mu(F_t^\star) + U_{it}.$$

This formulation implies that $\mu(F_t^\star)$ can also serve as valid factors, meaning that the original factors $F_t^\star$ are only identified up to a nonlinear invertible transformation. This ambiguity is similar to the linear case, where factors can only be identified up to an invertible matrix transformation.

The next theorem presents the convergence rate on factor recovery:

Theorem 3. *Under the same conditions as in Theorem 2, there exists a function $\mu : \mathbb{R}^{K_1} \to \mathbb{R}^K$ such that, with probability at least $1 - C\exp(-cT) - C\exp(-cL)$, for N and T sufficiently large,*

$$\frac{1}{T}\sum_{t=1}^{T} \|\mu(\widehat{F}_t) - F_t^\star\|^2 \lesssim (L^{-1}K_1 + NL^{-1}T^{-\frac{2\beta}{2\beta+K}} + NL^{-2}) \log^4(T).$$

The result indicates that when the factors are strong, meaning $L = N$, the convergence rate aligns with that in Theorem 2, and the estimator achieves consistency. However, if the factors are extremely weak—affecting only a small number of variables in X, such as when $L \asymp 1$—the right-hand side fails to converge to zero, and consistent estimation of the factors may not even be possible.

It is insightful to compare our result with that of Bai and Ng (2023), which examines the convergence of factor estimates using PCA when the true factors are weak. Using our notation, their result suggests an error rate of $L^{-1} + (NL^{-1}T^{-1})^2$ in the linear case. In contrast, our rate, $L^{-1} + (NL^{-1}T^{-1}) + NL^{-2}$, is not sharp due to the challenges associated with analyzing the outputs of a specific hidden layer in a DNN.

Importantly, our result does not require the number of neurons in the bottleneck layer to match the number of factors in the DGP. Instead, it only assumes the existence of a DGP specified in (1), where the number of factors K is less than or equal to our chosen K_1.

Identifying the number of factors is a critical problem in (linear) factor analysis. However, determining the true number of factors in nonlinear models is significantly more challenging because of the inherent ambiguity, even at the population level.

For context, consider a linear factor model given by $X_t = \Lambda F_t^\star + U_t$, where $F_t^\star \in \mathbb{R}^K$. Under the conditions that the minimal eigenvalues of $N^{-1}\Lambda^\top\Lambda$ and $\mathrm{Cov}(F_t^\star)$ are asymptotically bounded from below, and that the eigenvalues of $\mathrm{Cov}(U_t)$ are bounded from above, the number of factors K can be readily identified, as demonstrated by Chamberlain and Rothschild (1983). Bai and Ng (2002) also provide a consistent procedure for recovering K.

In nonlinear factor models, however, the notion of the number of factors becomes ambiguous. According to Theorem 2 from Schmidt-Hieber (2021), which extends the Kolmogorov–Arnold representation theorem, any nonlinear factor model can be reduced to a single-factor form. Specifically, there exists a function $\psi : [-B, B]^K \to \mathbb{R}$, such that for any function $\varphi_i^\star : [-B, B]^K \to \mathbb{R}$, a corresponding function $\widetilde{\varphi}_i^\star : \mathbb{R} \to \mathbb{R}$ exists, satisfying

$$\varphi_i^\star(x_1, \ldots, x_K) = \widetilde{\varphi}_i^\star \left(3 \sum_{j=1}^{K} 3^{-j} \psi(x_j) \right).$$

This implies that the original multi-factor model can be reconstructed using a single factor, denoted as $\widetilde{F}_t \in \mathbb{R}$, where $\widetilde{F}_t$ is defined by $3 \sum_{j=1}^{K} 3^{-j} \psi(F_{jt}^\star)$. Consequently, the function $\varphi_i^\star(F_t^\star)$ can be rewritten as $\widetilde{\varphi}_i^\star(\widetilde{F}_t)$. Although this representation reduces the number of factors, there are, to the best of our knowledge, no existing results concerning the smoothness properties of $\varphi^\star(\cdot)$ following this transformation. It is likely that the function becomes less smooth, as otherwise, this would provide a method to break the minimax rate. Consequently, we cannot use this result to justify always adopting a single neuron in the bottleneck layer, since our approach relies on the smoothness of the nonlinear functions.

On the other hand, it may be possible in certain cases to benefit from expressing the original K-factor model in a way that introduces more factors but results in a smoother $\varphi^\star(\cdot)$. For instance, consider a single factor model defined as $\varphi_i^\star(F_t^\star) = \Lambda_i F_t^\star + \psi(F_t^\star)$, where $\psi(\cdot)$ is a β-smooth function. By treating $\psi(F_t^\star)$ as an additional factor, the model becomes a two-factor linear model with an effectively infinite smoothness. This modification enables a convergence rate of $N^{-1} + T^{-1}$, which is faster than the rate $N^{-1} + T^{-\frac{2\beta}{2\beta+1}}$ achievable by fitting a single-factor nonlinear model.

Given these considerations, the conventional notion of the number of factors in a nonlinear factor model becomes ambiguous and may need to be defined alongside the smoothness of the factor loading functions. In practice, we treat it as one of the hyperparameters in the architecture of AEs, determining it through a model selection procedure. This approach makes intuitive sense and aligns with common practice, though providing a formal justification is left for future work.

3.3 Extensions and Applications

In this section, we delve into two extensions of AEs that pave the way for a wider range of applications, enhancing their flexibility and utility.

3.3.1 Factor-Augmented Prediction

When dealing with large datasets, we often encounter scenarios where we not only wish to reconstruct or compress the original data but also make accurate predictions for related outcomes. Standard AEs excel at capturing latent structures in data through unsupervised learning, but they do not directly incorporate information that could enhance predictive performance for external targets. This is where supervised extensions of AEs come into play. By integrating predictive tasks into the AE framework, we can leverage shared latent structures to improve both reconstruction and prediction, making the model more versatile and powerful for applications where joint modeling of input and output data is beneficial.

Consider the scenario where we also observe a large panel $Y_t \in \mathbb{R}^M$ that we wish to predict, in addition to reconstructing X_t. We assume that Y_{t+1} depends on the same latent factors, $F_t^\star$, modeled as:

$$Y_{i,t+1} = \phi_i^\star(F_t^\star) + V_{i,t+1}, \quad \text{for} \quad t = 1, 2, \ldots, T. \tag{6}$$

To train this model, one approach is to perform a nonparametric regression of $Y_{i,t+1}$ on the factors extracted from AEs pre-trained solely with X_t. Alternatively, a more cohesive method involves jointly fitting X_t and Y_{t+1} while training the AE. This strategy originates from Le et al. (2018), who introduced a supervised autoencoder (SAE) architecture. In this setting, X_t remains the input, and the model is trained to simultaneously construct X_t and predict Y_{t+1}, leveraging the shared latent structure. Figure 4 presents the architecture of the SAE.

This SAE approach differs from factor-augmented regressions (as in Bernanke and Boivin (2003)) in two key aspects. First, the SAE framework is nonparametric, enabling flexible and complex nonlinear relationships between the target and the factors. Second, the high dimensionality of the target in SAEs imposes meaningful supervision on the factor construction process, enhancing the model's predictive power. In contrast, traditional factor-augmented regressions typically use low-dimensional targets, rely on linear DGPs, and construct factors without incorporating information from the target.

Suppose we observe the sequences X_t and Y_{t+1}, for $t = 1, 2, \ldots, T-1$, and our goal is to predict Y_{T+1}. To achieve this, we first train an SAE using the following loss function:

$$\begin{aligned}&(\widehat{\varphi}_1, \ldots, \widehat{\varphi}_N, \widehat{\phi}_1, \ldots, \widehat{\phi}_M) \circ \widehat{\rho} \\ =& \underset{(\varphi_1,\ldots,\varphi_N,\phi_1,\ldots,\phi_M)\circ\rho \in \mathcal{F}_{SAE}^{K_1}}{\arg\min} \sum_{t=1}^{T-1} \left(\sum_{i=1}^{N} \left(\varphi_i(\rho(X_t)) - X_{it}\right)^2 + \sum_{j=1}^{M} \left(\phi_j(\rho(X_t)) - Y_{j,t+1}\right)^2 \right).\end{aligned} \tag{7}$$

The supervised AE function class $\mathcal{F}_{SAE}^{K_1}$ is defined as:

$$\mathcal{F}_{SAE}^{K_1} := \Big\{ (\varphi_1, \ldots, \varphi_N, \phi_1, \ldots, \phi_M) \circ \rho : \rho \in \mathcal{F}_N^{K_1}(d_1, w_1, T^{5\beta+5}, B),$$

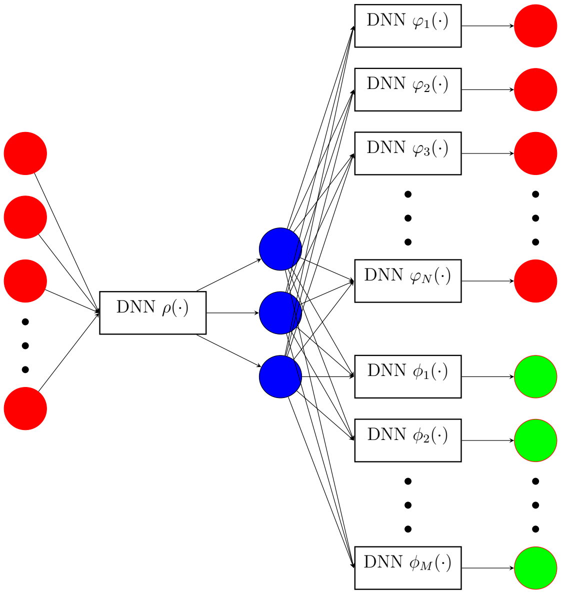

Figure 4: Illustration of Disjoint Output Supervised Autoencoder

Note: This graph illustrates a disjoint output supervised autoencoder architecture. The encoder retains the same structure as the canonical autoencoder's encoder shown in Figure 2. The decoder, however, consists of multiple separate DNNs, each responsible for mapping the bottleneck layer to a single output. The portion of the output highlighted in red is aimed at reconstructing the input, using the same architecture as shown in Figure 3. The other portion, marked in green, is designed to predict additional target variables, while supervising the factor construction process.

$$\varphi_i \in \mathcal{F}^1_{K_1}(d_2, w_2, T^{5\beta+5}, B), i \in [N], \phi_j \in \mathcal{F}^1_{K_1}(d_2, w_2, T^{5\beta+5}, B), j \in [M]\Big\}.$$

The predictor for $Y_{j,T+1}$, denoted by $\widehat{Y}_{j,T+1}$, is given by $\widehat{\phi}_j(\widehat{\rho}(X_T))$. We then establish the following corollary.

Corollary 1. *Assume $\phi^\star_i \in \mathcal{H}^\beta(\Omega, B)$ for $i = 1, \dots, M$. In addition, we assume that conditional on $F^\star$, $(U_t^\top, V_{t+1}^\top)^\top \in \mathbb{R}^{N+M}$ is i.i.d. and takes the form $\Sigma^{1/2}Z_t$, where $Z_t \in \mathbb{R}^{N+M}$ consists of independent sub-Gaussian random variables with sub-Gaussian norm bounded by σ_z^2 and $\Sigma \in \mathbb{R}^{N+M}$ is positive semi-definite with bounded spectral norm. Under the same conditions as in Theorem 2, as* $\min(N + M, T)$ *becomes sufficiently large, we have:*

$$\frac{1}{N+M}\left(\sum_{i=1}^{N}\mathbb{E}\big(\widehat{X}_{i,T} - X^\star_{i,T}\big)^2 + \sum_{j=1}^{M}\mathbb{E}\big(\widehat{Y}_{j,T+1} - \phi^\star_j(F^\star_T)\big)^2\right)$$
$$\lesssim ((N+M)^{-1}K_1 + T^{-\frac{2\beta}{2\beta+K}} + L^{-1} + T^{-1}L)\log^4((N+M)T). \tag{8}$$

Consequently, when $N \lesssim M$, we obtain:

$$\frac{1}{M}\sum_{j=1}^{M}\mathbb{E}\big(\widehat{Y}_{j,T+1} - \phi^\star_j(F^\star_T))\big)^2 \lesssim (M^{-1}K_1 + T^{-\frac{2\beta}{2\beta+K}} + L^{-1} + T^{-1}L)\log^4(NT). \tag{9}$$

This corollary suggests that when the dimension of X has a small or equal order compared to the dimension of Y, the prediction error for $\widehat{Y}_{T+1}$ vanishes as the sample size increases, ensuring that the model effectively leverages the information from Y for accurate predictions. However, if M is much smaller than N, the influence of Y_t in the training process diminishes, and the prediction performance for $\widehat{Y}_{T+1}$ may not be guaranteed.

3.3.2 Matrix Completion

AEs provide a powerful framework for tackling problems involving nonlinear matrix completion and missing data. In many real-world applications, such as recommendation systems, financial time series, or large-scale survey data, we encounter datasets that are both high-dimensional and partially observed, sometimes with missing values scattered throughout. Traditional matrix completion methods, which rely on linear assumptions, may struggle to capture the complex, nonlinear relationships inherent in the data.

AEs offer a natural solution to this challenge by learning a compact, latent representation of the observed data that can encode nonlinear structures effectively. By training the AE to reconstruct the original matrix from its latent representation, the model can infer and fill in missing entries in a way that captures intricate patterns. This capability is crucial not only for matrix completion but also for broader applications, such as causal inference.

In the estimation of causal effects within a panel setting, counterfactual outcomes—representing what would have occurred under alternative treatment conditions—can be regarded as missing data. AEs can assist in this context by imputing these unobserved counterfactuals, offering a novel and effective approach to estimating causal relationships (see, e.g., Athey et al. (2017), Bai and Ng (2021)). Importantly, missing data in these scenarios are often not random. In many instances, the data can be structured so that all missing entries are concentrated within a sub-block of a matrix. Therefore, we also explore approaches for addressing and imputing this type of structured missing data.

Specifically, we first consider a scenario with random missing data, where we observe an incomplete version of the matrix X, denoted by $\tilde{X}$, which we assume follows the DGP:

$$\tilde{X}_{it} = \begin{cases} 0, & \text{with probability } 1 - \pi_i, \\ X_{it}, & \text{with probability } \pi_i, \end{cases} \tag{10}$$

where X follows the nonlinear factor model described in (1), and $\pi_i \in (0, 1)$ represents the probability of observing an entry for the ith variable. This probabilistic framework captures scenarios where the data is missing at random and the degree of missingness is heterogeneous.

Using $\tilde{X}_t$ to train an AE, we derive the following result regarding the output $\widehat{X}_t$:

Corollary 2. *Assume $\min_{1 \le i \le N} \pi_i$ is lower bounded by some positive constant ε, U_{it} are independent of each other, and $\tilde{X}_{it}$ is missing at random, satisfying (10). Under the conditions specified in Theorem 1, with probability at least $1 - C\exp(-cT^{K/(2\beta+K)})$, for sufficiently large $\min(N, T)$, we have:*

$$\frac{1}{NT}\sum_{t=1}^{T}\sum_{i=1}^{N}\left(\widehat{X}_{it}/\widehat{\pi}_i - X^{\star}_{it}\right)^2 \lesssim \left(T^{-\frac{2\beta}{2\beta+K}} + N^{-1}K_1 + T^{-1}\inf_{\rho\in\mathcal{F}_N^K}\sum_{t=1}^{T}\|\rho(\tilde{X}_t) - F^{\star}_t\|^2\right)\log^4(T), \tag{11}$$

where c and C are constants independent of N and T and $\widehat{\pi}_i = \max(\varepsilon, T^{-1}\sum_{t=1}^{T} 1_{\{\tilde{X}_{i,t} \neq 0\}})$.

The theorem indicates that consistent recovery of the expected values of the missing entries is achievable under the Frobenius norm. Given that matrix completion is primarily an in-sample exercise, utilizing an over-parameterized encoder can be beneficial for attaining a desirable error rate, as discussed previously.

Next, we consider the scenario of structured missing data, for which we employ SAEs. We assume that $X_t = (X_t^{(1)}, X_t^{(2)})$ follows the nonlinear factor model described in (1), where $X_t^{(1)} \in \mathbb{R}^{N_1}$, $X_t^{(2)} \in \mathbb{R}^{N_2}$, and $N_1 + N_2 = N$. We assume that $X_t^{(1)}$ is fully observed across all time periods, while $X_t^{(2)}$ follows a structured missing pattern such that there exists a non-random $T_0 > 0$ where $X_t^{(2)}$ is missing for $t > T_0$.

The matrix X illustrates the separation between observed and missing data segments as follows:

$$X = \left(\begin{array}{cccc|cccc} X_{1,1}^{(1)} & \cdots & \cdots & X_{1,T_0}^{(1)} & X_{1,T_0+1}^{(1)} & \cdots & \cdots & X_{1,T}^{(1)} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ X_{N_1,1}^{(1)} & \cdots & \cdots & X_{N_1,T_0}^{(1)} & X_{N_1,T_0+1}^{(1)} & \cdots & \cdots & X_{N_1,T}^{(1)} \\ \hline X_{1,1}^{(2)} & \cdots & \cdots & X_{1,T_0}^{(2)} & * & \cdots & \cdots & * \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ X_{N_2,1}^{(2)} & \cdots & \cdots & X_{N_2,T_0}^{(2)} & * & \cdots & \cdots & * \end{array} \right). \tag{12}$$

The economic context behind this missing data problem is to estimate the average treatment effect for the treated group. In this setup, $X^{(1)}$ represents the control group outcomes, while $X^{(2)}$ corresponds to the treated group that undergoes an irreversible treatment starting at time $T_0 + 1$. The objective is to assess how the treatment impacts the treated units and periods by imputing the potential outcomes for $X^{(2)}$ from $T_0 + 1$ to T, and comparing them with the actual observed outcomes. This analysis assumes that potential outcomes at each point depend only on the contemporaneous treatment status of each unit and do not rely on past treatments or the treatments of other units.

As discussed by Athey et al. (2017), this missing data pattern encompasses two significant special cases in the causal inference literature. The first is the unconfoundedness literature, such as Imbens and Rubin (2015), which typically focuses on scenarios involving a single treated period. The second is the synthetic control literature, including Abadie et al. (2010) and Abadie (2021), which centers on settings with a single treated unit.

To impute these missing values, we train an SAE using $X_t^{(1)}$ as the input and X_t as the output, with the portion corresponding to $X_t^{(2)}$ serving as the supervised target. This approach effectively transforms the problem into an SAE task, where the model leverages the "in-sample" data available up to T_0 to learn a low-dimensional structure that helps predict $X_t^{(2)}$ for $t > T_0$.

The following corollary becomes a straightforward application of Corollary 1.

Corollary 3. *Under the same conditions as in Theorem 2, as* $\min(N, T_0)$ *becomes sufficiently large, and* $N_1 \lesssim N_2$*, we have:*

$$\begin{aligned}&\frac{1}{NT}\sum_{t=1}^{T}\left(\sum_{i=1}^{N_1}\mathbb{E}\big(\widehat{X}_{i,t}^{(1)} - X_{i,t}^{(1),\star}\big)^2 + \sum_{i=1}^{N_2}\mathbb{E}\big(\widehat{X}_{i,t}^{(2)} - X_{i,t}^{(2),\star}\big)^2\right)\\ \lesssim&(N_1^{-1}K_1 + T_0^{-\frac{2\beta}{2\beta+K}} + L^{-1} + T_0^{-1}L)\log^4(NT).\end{aligned}$$

and

$$\frac{1}{N_2(T-T_0)}\sum_{t=T_0+1}^{T}\sum_{i=1}^{N_2}\mathbb{E}\big(\widehat{X}_{i,t}^{(2)} - X_{i,t}^{(2),\star}\big)^2 \lesssim (N_1^{-1}K_1 + T_0^{-\frac{2\beta}{2\beta+K}} + L^{-1} + T_0^{-1}L)\log^4(NT).$$

In the special linear case, the best achievable rate given by Corollary 1 is $N_1^{-1} + T_0^{-1/2}$ when $L = T_0^{1/2}$. Bai and Ng (2021) obtain a stronger result of $N_1^{-1} + T_0^{-1}$, assuming that the factors are pervasive (i.e., $L = N$). Athey et al. (2017)'s result suggests a convergence rate $T^{-1} + N^{-1/2}$, provided that at least a constant portion of periods for each unit is observed and the matrix X has a low rank. Their rate applies to the entire matrix, not just the missing portion, but is comparable to our rate when the missing portion is of the same order as the complete data.

4 Monte Carlo Simulations

In this section, we conduct simulation experiments to validate our theoretical predictions and examine practical considerations in the design and training of AEs and SAEs. This analysis bridges theoretical insights with empirical applications, exploring the strengths and limitations of these methods in finite sample settings.

4.1 Simulation Setup

We begin by introducing the DGPs used in our simulations. Specifically, we consider four distinct DGPs by (1), comprising one linear factor model and three nonlinear factor models.

The baseline model is defined by $\varphi_i^\star(x) = C\Lambda_i^\top x$, representing a traditional linear factor structure. The first nonlinear model we consider is an example of the generalized linear latent factor model, specified as $\varphi_i^\star(x) = C\exp(\Lambda_i^\top x)$. This model introduces exponential transformations to the baseline, adding nonlinearity and challenging methods that rely on linearity. The next model completely breaks the linear structure and serves as an example of a generalized nonlinear factor model, expressed as $\varphi_i^\star(x) = C\exp(-\|\Lambda_i - x\|^2)$. This model does not have an inherent linear low-dimensional structure, posing challenges for methods that might benefit from it. The final model is a polynomial factor model, given by $\varphi_i^\star(x) = C_1\Lambda_{1i}^\top x + C_2 x^\top \Lambda_{2i} x$, blending linear interactions with higher-order dependencies.

For each of the 100 Monte Carlo repetitions, we use a five-factor model by setting $K = 5$ and defining $F_t^\star \in \mathbb{R}^5$, where each element of $F_t^\star$ follows a uniform distribution $\mathbb{U}(-2, 2)$. The elements of $\Lambda_i, \Lambda_{1i}, \Lambda_{2i}$ are i.i.d. following a uniform distribution $\mathbb{U}(-1, 1)$. The noise U_t is normally distributed, $\mathcal{N}(0, \mathbb{I}_N)$. Calibration of C, C_1, and C_2 ensures that the unconditional variance of $\varphi_i^\star(F_t^\star)$ for each model is normalized to 1, and that each term contributes equally to the variance for the polynomial factor model.

Throughout the simulations, we benchmark performance against PCA. While PCA is traditionally associated with linear factor analysis, recent research has demonstrated its effectiveness in approximating certain nonlinear models.[5] For PCA, we report results using the number of factors ranging from 1 to 19 in increments of two to illustrate its impact on PCA performance.

Turning to the design of AEs, we first address the bottleneck layer. We vary the number of neurons in this layer, K_1, in the same way as with PCA, ranging from 1 to 19. For the encoder, we use a single hidden layer with 20 neurons to ensure it is wider than the bottleneck. We experiment with four decoder architectures of increasing complexity. The first architecture (AE1) has a single hidden layer in the decoder with two neurons, followed by AE2 with four neurons, and AE3 with eight neurons in their respective hidden layers. All these AEs have disjoint outputs. Additionally, we include a fourth architecture (AE4), which has a fully connected decoder on the basis of AE3's. We illustrate AE3's architecture in Figure 5.

When designing these architectures, we adhere to the principle of parsimony, experimenting with simple yet non-trivial structures to validate our theoretical results. Model

[5] According to Udell and Townsend (2019), a matrix with a small spectral norm can be effectively approximated by a low-rank matrix. For a matrix $X \in \mathbb{R}^{m\times n}$ where $m \geq n$ and $0 < \epsilon < 1$, the approximation error is bounded as:

$$\inf_{\operatorname{rank}(Y)\leq r} \|X - Y\|_\infty \leq \epsilon\|X\|_2, \text{ with } r = \lceil 72\log(2n+1)/\epsilon^2 \rceil.$$

Additionally, Griebel and Harbrecht (2014) and Xu (2017) demonstrate that the matrix $X_{it} := \varphi(\Lambda_i, F_t^\star)$ in nonlinear models admits a low-rank approximation, where for any $\delta > 0$,

$$\inf_{\operatorname{rank}(Y)\leq r} \|X - Y\|_\infty \lesssim \delta^\beta, \text{ with } r \asymp \delta^{-K},$$

where K is the dimension of $F_t^\star$, and β is given by Assumption 2.

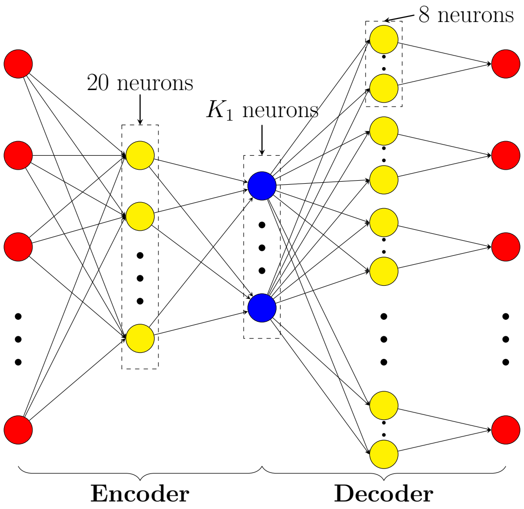

Figure 5: Illustration of AE3's Architecture

Note: This graph illustrates the architecture of AE3 used in our numerical analysis. The encoder has one hidden layer with 20 neurons, while the decoder consists of multiple separate neural nets, each containing a hidden layer with 8 neurons. We vary the number of neurons, K_1, in the bottleneck layer.

complexity can be increased if needed—such as when training losses do not reach a sufficiently low level compared to a benchmark method or architecture—to avoid potential underfitting. In most cases we consider, the training loss does not reach zero, as our disjoint output AEs have a limited number of parameters relative to the sample size, except for the fully connected AE4 model and when T is small. Thus, most of our results do not fall within the over-parametrized regime. We address the training details separately based on different AE applications, starting with using AEs for extracting common components.

4.2 Finite-Sample Recovery of Common Components with Autoencoders

To train the aforementioned AEs in this scenario, we use stochastic gradient descent with the Adam optimizer (Kingma and Ba, 2014) to minimize the loss defined in (4). The batch size is fixed at 5 for $T = 50$ and 50 for $T = 500$, ensuring each epoch consists of ten gradient descent updates. In simulations for AEs, the only parameter we tune is the learning rate in the optimization algorithm, selected from $\{0.005, 0.01, 0.05, 0.1\}$ using a training-validation scheme, with results reported for each K_1. To mitigate the effects of random initialization during training, we report the average performance of an ensemble of 10 independently trained models, following standard practices in the literature.

For each tuning parameter (e.g., learning rate), we train the AEs using the first 4/5 of the sample (training sample) and apply the inferred encoders and decoders to construct an estimate of $X^\star$ in the remaining sample (validation sample). This approach allows us to calculate the validation loss and determine the optimal tuning. When training, we apply early stopping if the validation loss does not decrease for 50 consecutive epochs. Finally, we refit the model using the entire sample with the selected tuning parameter.

We report the mean squared error (MSE) given by:

$$N^{-1}T^{-1}\sum_{t=1}^{T}\sum_{i=1}^{N}(\widehat{X}_{it}-\varphi_i^\star(F_t^\star))^2,$$

with a variety of choices $(N,T)=(50,500),(200,500),(200,50)$. The comparative performance shown in Figure 6 reveals differences in how PCA and AEs respond to changes in sample size and model complexity.

In the baseline linear case, PCA achieves optimal MSE when the number of factors is set to five (the true value) and is clearly the best performer when the sample size N is larger than T. In this linear setting, AE1 through AE3 exhibit similar performance. These AEs tend to achieve lower MSEs than PCA when the number of factors is below five; however, their performance, like that of PCA, begins to deteriorate as the number of factors increases beyond five. This decline is due to the addition of extra factors, which introduces noise into the estimation rather than improving accuracy.

AE4 follows a U-shaped pattern similar to the other AEs but demonstrates significantly worse performance. Its greater complexity makes it more prone to overfitting, resulting in higher MSEs compared to the simpler models. This result highlights the benefit of disjoint output architecture, which achieves a balanced trade-off between model flexibility and estimation performance.

The results for all nonlinear DGPs reveal a similar pattern. As the number of factors increases, PCA's performance continues to improve, suggesting that nonlinearity compels PCA to extract more linear factors to effectively approximate the model. In all cases, PCA struggles to match the performance of the nonlinear methods, even with up to 20 factors in the case where $N=200$ and $T=500$, while these nonlinear methods deliver superior results with just five factors. When N is small relative to T, AE1 through AE3 perform comparably and significantly outperform AE4, which quickly begins to overfit. Conversely, when N is large relative to T, AE1 tends to dominate in terms of performance, while AE2 and AE3 become comparable to AE4, indicating that their relative advantage diminishes as the ratio between sample size and dimensionality shifts.

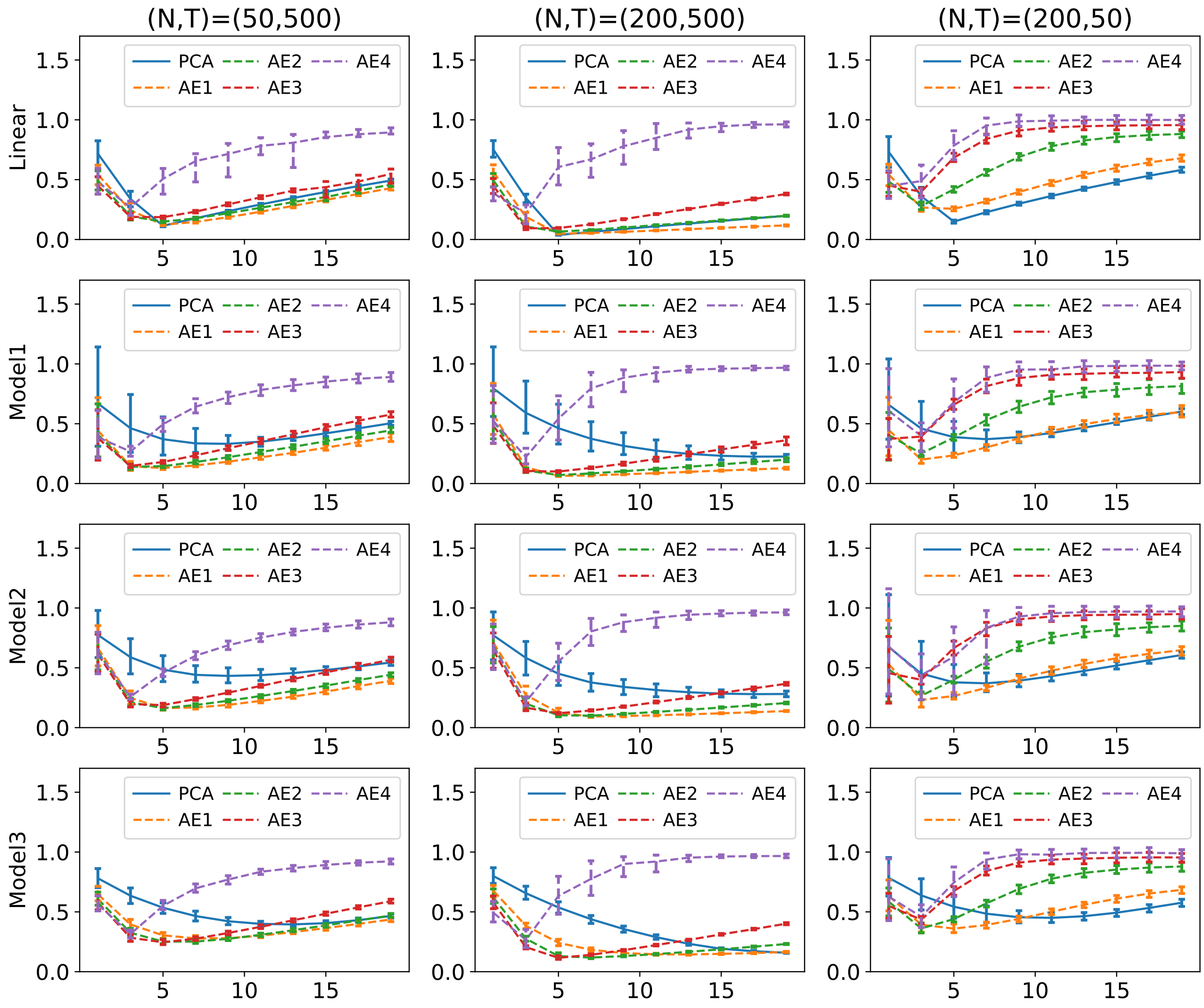

Figure 6: Mean Squared Error Comparison in Simulations

Note: This figure shows boxplots of the mean squared errors for each specified number of factors, comparing five different methods applied to four models across various choices of N and T, based on 100 Monte Carlo repetitions.

In the supplementary appendix, we also compare AEs with Kernel PCA. The findings reveal that while Kernel PCA introduces nonlinearity to its feature extraction process, it still falls short of matching the performance of AEs. This performance gap is largely due to the inherent limitations of Kernel PCA, which relies on pre-specified kernel functions and lacks the adaptive, data-driven architecture of AEs. In contrast, AEs, with their deep learning structures, dynamically learn representations that capture intricate relationships within the data, enabling them to outperform Kernel PCA across various sample sizes and model complexities.

4.3 Finite-Sample Matrix Completion with Autoencoders

Next, we examine the performance of missing data imputation using AEs, employing the polynomial factor model as the DGP to generate data, with $N = 50$ and $T = 500$, as motivated by our empirical application below.

To simulate the missing-at-random scenario, we generate a probability $\pi_i \sim \mathbb{U}(\pi_{min}, 1)$ for each variable X_i, representing the likelihood of observing a value. We vary π_{min} with values of 0.2, 0.4, 0.6, and 0.8. We train AE1 through AE3 using the observed data matrix $\tilde{X}$, with K_1 fixed at $1, 3, 5, 7$, and 9, respectively. The training, validation, and refitting steps follow the same procedure as described above.

Table 1 reports the MSEs for the imputed entries:

$$\frac{1}{N}\sum_{i=1}^{N}\frac{1}{\#\{t : \tilde{X}_{i,t} = 0\}}\sum_{t\in\{t:\tilde{X}_{i,t}=0\}}\left(\widehat{X}_{it}/\widehat{\pi}_i - X^{\star}_{it}\right)^2.$$

A few notable results emerge. As data availability increases ($\pi_{\min}$ becomes larger), the performance of all methods improves. Overall, AE1–AE3 consistently outperform PCA across all scenarios. While PCA benefits from additional factors to better approximate the nonlinear structure, AE4's performance deteriorates with more factors due to its excessive complexity, leading to overfitting. In contrast, AE1–AE3 demonstrate robustness with respect to the number of factors, often reaching optimal performance with the actual five factors.

4.4 Finite-Sample Predictive Performance with Supervised Autoencoders

We conclude by evaluating the performance of SAEs, focusing on a prediction setting. The simulation framework for structured matrix completion closely mirrors this scenario and is

	$\pi_i \sim \mathbb{U}(0.2, 1)$						$\pi_i \sim \mathbb{U}(0.4, 1)$				
K_1	1	3	5	7	9	K_1	1	3	5	7	9
PCA	0.865	0.827	0.807	0.796	0.802	PCA	0.852	0.794	0.756	0.726	0.712
AE1	0.793	0.684	0.661	0.661	0.672	AE1	0.763	0.625	0.585	0.579	0.585
AE2	0.771	0.645	0.635	0.651	0.676	AE2	0.738	0.574	0.544	0.556	0.577
AE3	0.755	0.642	0.649	0.683	0.719	AE3	0.721	0.558	0.548	0.576	0.614
AE4	0.731	0.743	0.864	0.914	0.934	AE4	0.696	0.651	0.795	0.863	0.897
	$\pi_i \sim \mathbb{U}(0.6, 1)$						$\pi_i \sim \mathbb{U}(0.8, 1)$				
PCA	0.842	0.769	0.712	0.666	0.635	PCA	0.831	0.743	0.674	0.616	0.574
AE1	0.743	0.58	0.523	0.507	0.509	AE1	0.725	0.545	0.474	0.45	0.449
AE2	0.715	0.521	0.472	0.479	0.498	AE2	0.693	0.478	0.414	0.416	0.434
AE3	0.693	0.496	0.473	0.5	0.538	AE3	0.669	0.447	0.413	0.437	0.476
AE4	0.675	0.594	0.758	0.839	0.877	AE4	0.648	0.537	0.722	0.816	0.865

Table 1: Simulation Results for Matrix Completion

Note: This table reports the MSEs of imputed entries across various matrix completion algorithms, including the benchmark PCA method and multiple AE architectures of increasing complexity. K_1 corresponds to the number of factors in PCA and the number of neurons in the bottleneck layers of the AEs. π_i represents the heterogeneous probabilities of non-missing data for each row of X, drawn from different uniform distributions.

thus omitted.

In this supervised setting, we adopt a polynomial factor model as the DGP for X_{it}. Additionally, we simulate target variables Y_{it} based on (6), where $\phi_i^{\star}(x)$ also follows a polynomial factor model but with a distinct set of randomly generated parameters (e.g., Λ_{1i} and Λ_{2i}). We adjust C_1 and C_2 in the DGP to calibrate the signal-to-noise ratio for X and Y based on empirical observations. The signal to noise ratio of X is set to 1, reflecting the fact that the top five factors of book-to-market ratios explain approximately 50% of the variance. For Y, the signal-to-noise is set to 1%, aligning with the empirical observation that the out-of-sample R^2s for predicting factors are lower than 1%.

We fix $N = 50$, while varying the dimension of Y_{it}, M. The in-sample size is set at $T = 200$. As discussed earlier, a sparse encoder can enhance model's out-of-sample performance. Sparsity is introduced here via a pruning procedure. Specifically, we first train SAEs with a fully connected encoder and then apply post-training pruning by setting weights below a specified threshold to zero. The pruning threshold is determined by sorting all weights and truncating those below a specified percentile, referred to as the pruning ratio. We vary the pruning ratio from 0% to 95% in increments of 5%. Since pruning is applied after training, adjusting the pruning ratio is computationally efficient.[6] Consequently, the pruning ratio and learning rate are selected jointly during the validation step.

After training and validation, we evaluate the selected model's out-of-sample performance using a separate testing sample of 100 observations. For each target variable, we can calculate its out-of-sample R^2 against its in-sample mean, and then report the average over all such R^2s in Table 2. For comparison, we include principal component regression (PCR) as a linear benchmark. In PCR, each variable Y_i is regressed on the principal components of X_{it} using the entire in-sample dataset. Predictions are then made out-of-sample using the in-sample estimates of principal component weights and regression coefficients.

The results highlight the advantages of our SAE models compared to PCR, which consistently yield negative out-of-sample R^2s values regardless of the number of factors. This underperformance can be attributed to PCR's lack of supervision when recovering factors and its reliance on linear approximations, which prove inadequate for capturing the nonlinear relationships needed to predict the target variables.

SAE4 also struggles to achieve positive R^2 values due to its tendency to overfit in out-of-sample settings. In contrast, SAE1 through SAE3 deliver positive R^2 values, demonstrating their ability to extract nonlinear factors effectively while maintaining sufficient complexity control to avoid overfitting.

As M (the dimension of Y_{it}) increases, prediction becomes increasingly important, leading to improved predictive performance. While the number of neurons in the decoder layer grows as M increases, the total number of weights per target remains fixed for SAE1-SAE3. However, for SAE4, the total number of weights per target increases, contributing to its progressively worse performance.

Developing better architectures, optimization algorithms, or efficient parameter-tuning schemes are important and ongoing research areas in machine learning. Nevertheless, our primary focus here is not on finding the optimal model, but on validating our theoretical insights and identifying a useful model for empirical analysis.

5 Empirical Applications in Economics

In this section, we discuss three separate applications that illustrate the use of AEs and SAEs with economic datasets.

[6] For alternative approaches, see LeCun et al. (1989), Hassibi and Stork (1992), and Frankle and Carbin (2018).

	$M = 25$						$M = 50$				
K_1	1	3	5	7	9	K_1	1	3	5	7	9
PCR	-0.323	-0.922	-1.731	-2.45	-3.38	PCR	-0.297	-1.049	-1.869	-2.692	-3.585
SAE1	0.248	0.267	0.28	0.266	0.277	SAE1	0.309	0.35	0.341	0.348	0.347
SAE2	0.245	0.258	0.257	0.261	0.271	SAE2	0.302	0.328	0.341	0.342	0.354
SAE3	0.176	0.232	0.26	0.246	0.249	SAE3	0.264	0.316	0.337	0.347	0.345
SAE4	-0.146	-0.039	-0.038	0.021	0.025	SAE4	-0.087	-0.048	-0.032	0.009	0.025
	$M = 100$						$M = 200$				
K_1	1	3	5	7	9	K_1	1	3	5	7	9
PCR	-0.327	-1.087	-1.924	-2.762	-3.674	PCR	-0.31	-1.098	-1.885	-2.739	-3.561
SAE1	0.356	0.381	0.387	0.383	0.386	SAE1	0.387	0.392	0.395	0.395	0.395
SAE2	0.343	0.375	0.382	0.386	0.382	SAE2	0.374	0.39	0.389	0.392	0.393
SAE3	0.315	0.368	0.378	0.379	0.381	SAE3	0.339	0.384	0.388	0.392	0.392
SAE4	-0.08	-0.052	-0.021	-0.031	0.005	SAE4	-0.091	-0.051	-0.042	-0.038	-0.012

Table 2: Simulation Results for Supervised Autoencoders

Note: This table reports the out-of-sample R^2 of predicted values across various predictive methods, including the benchmark Principal Component Regression (PCR) method and multiple SAE architectures of increasing complexity. K_1 corresponds to the number of factors in PCA and the number of neurons in the bottleneck layers of the SAEs. The dimension of Y, M, is varied while the dimension of X is fixed at $N = 50$. The sample size is fixed at $T = 200$.

5.1 Macroeconomic Forecasting

Our initial exercise aims to predict key monthly macroeconomic indicators, including industrial production growth, inflation, changes in the unemployment rate, and non-farm payroll growth. Our approach builds on the framework introduced by Stock and Watson (2002), but rather than utilizing a linear factor model, we incorporate nonlinear factors through the application of AEs.

For this study, we use the FRED-MD dataset, as compiled by McCracken and Ng (2016). This dataset includes a comprehensive range of economic categories such as output and income, labor market, consumption, orders and inventories, money and credit, interest rates and exchange rates, prices, and the stock market. It offers 119 potential predictors, covering the period from February 1960 to December 2019.[7]

We approach the task of prediction as one of missing data imputation. Let $X_t \in \mathbb{R}^{119}$ de-

[7] Following the procedure outlined by McCracken and Ng (2016), we preprocess the data by applying necessary transformations to the variables. We also exclude variables that are missing in earlier years or are unavailable without a long lag, ensuring the robustness and integrity of our results. We select this publicly accessible and widely used dataset to facilitate comparability with other studies in the literature.

	Industry Production						Inflation				
K_1	1	2	3	4	5	K_1	1	2	3	4	5
PCA	9.2	7.8	9.3	9.6	7.0	PCA	**0.0**	-1.3	-6.0	-6.3	-6.7
AE1	**10.1**	10.4	11.7	13.8	13.3	AE1	-0.2	**-0.7**	-1.1	**1.1**	0.3
AE2	8.7	**11.6**	11.4	**15.0**	**15.7**	AE2	-0.8	-3.0	**-1.0**	-1.2	**0.9**
AE3	7.2	6.5	**12.6**	13.6	13.4	AE3	-1.0	-2.0	-2.6	-1.0	0.5
	Unemployment Rate						Nonfarm Payroll				
PCA	13.0	11.6	14.7	13.1	12.2	PCA	**1.1**	**2.5**	1.0	3.5	-5.6
AE1	**16.8**	**19.1**	**20.2**	19.1	17.0	AE1	-1.4	2.2	**6.7**	**4.3**	0.5
AE2	14.1	17.6	17.7	**19.3**	**17.6**	AE2	-5.4	0.5	1.5	2.9	**1.7**
AE3	**16.8**	16.9	14.6	15.7	14.4	AE3	-5.5	-8.6	-1.6	-10.0	-8.2

Table 3: Out-of-sample R^2s

Note: This table reports the out-of-sample R^2 values in percentages for four different target variables, comparing PCA and various AE architectures across different factor counts. Bolded values indicate the best-performing model for each case. The architectures of these AEs match those used in the simulations.

note the predictor variables, while $Y_{t+1} \in \mathbb{R}^4$ represent the target variables for the subsequent month. Suppose we observe (X_t, Y_{t+1}) for $t = 1, \ldots, T-1$, in addition to X_T. To predict Y_{T+1}, we construct an $N \times T$ matrix, where the t-th column corresponds to $(X_t, Y_{t+1})^\top$. For the last column, the final four entries, which represent Y_{T+1}, are treated as missing. We then employ AEs and PCA to impute the missing values.

We begin by training these models on data from February 1960 to January 1990 ($T = 360$), after which we evaluate their performance by comparing the imputed values to the true observed values. This evaluation process is iterated monthly over a 30-year period, with each iteration expanding the training set by one month and shifting the evaluation window forward accordingly. For benchmarking, we use an Autoregressive (AR(1)) model as in Stock and Watson (2002), fitted with an expanding window for predictions. The out-of-sample R^2 values, relative to the output from the AR(1) model's output, are reported in Table 3.

The findings in Table 3 reveal that AE architectures generally outperform PCA in predicting key macroeconomic indicators, particularly as the number of factors increases. For industrial production, AE models show a clear advantage over PCA, with AE2 achieving the highest out-of-sample R^2 (15.7%) at $K_1 = 5$, suggesting that nonlinear factor structures significantly enhance predictive accuracy. Inflation is the most challenging to predict—-PCA's performance is consistently worse than the benchmark, with AE1 achieving the best R^2 of 1.1% at $K_1 = 4$. For the unemployment rate, AEs consistently outperform PCA, with AE1 yielding the highest R^2 values (20.2%) at $K_1 = 3$. In predicting nonfarm payroll growth, PCA performs well at lower factor counts, but AE1 becomes superior as factor count increases, reaching a peak R^2 of 6.7% at $K_1 = 3$. Overall, these results demonstrate that AEs, particularly at higher factor counts, offer substantial predictive improvements over PCA, highlighting the value of nonlinear modeling in forecasting macroeconomic variables.

5.2 Predicting the Cross-Section of Factor Returns

Our second exercise examines an asset pricing application centered on factor timing. Over the past several decades, a significant body of research in asset pricing has focused on uncovering and understanding the factors that explain the cross-sectional variation in expected returns. This research has fueled the widespread adoption of factor-based investing, where portfolios are allocated based on these systematic drivers of returns. Just as aggregate market returns have been found to exhibit predictable patterns, the returns of individual factors may also be predictable, presenting investment opportunities for dynamically adjusting positions on different factors based on their expected performance.

Extensive research has explored the predictability of individual factors, such as value (Cohen et al. (2003)) and momentum (Cooper et al. (2004); Daniel and Moskowitz (2016)), as well as the simultaneous prediction of multiple factors (Stambaugh et al. (2012); Akbas et al. (2015)). For instance, Arnott et al. (2023) investigate factor momentum and demonstrate that the past returns of factors can predict their future returns in the cross-section. Specifically, they show that a long-short portfolio—long on factors with above-median returns in the previous month and short on those with below-median returns—yields significant abnormal returns.

Building on this literature, Haddad et al. (2020) examine the predictability of the five principal components of factors, i.e., factor portfolios, using portfolio-level book-to-market (BM) ratios. These ratios are calculated by aggregating individual factors' BM ratios with portfolio weights derived from the eigenvectors of the returns covariance matrix. The BM ratio for each factor is, in turn, calculated by weighting the individual equities' BM ratios. Ultimately, predictions of principal components then translate into predictions of factors using these eigenvectors as weights.

In contrast to this multi-step procedure, our approach seeks to directly predict the cross-section of factor returns using each factor's book-to-market ratio, alongside additional characteristics such as momentum measures derived from factor returns over varying horizons. Our approach centers on the use of SAEs examined in simulations. The SAE architecture enables the extraction of nonlinear, low-dimensional components from these characteristics of all factors ($X_{i,t}$), while simultaneously learning to predict the cross-sectional of factor returns ($y_{i,t+1}$) for the next period.

Our analysis uses the extended monthly dataset from Haddad et al. (2020), accessed via Serhiy Kozak's website, spanning January 1974 to December 2019. This dataset includes 50 long-short, characteristic-sorted decile portfolios, along with the BM ratios for each factor. In addition to BM ratios, we calculate trailing 1-month (mom1), 6-month (mom6), and 12-month (mom12) returns as alternative predictors. The inclusion of mom12 requires the sample period to begin in 1975.

To train the SAE, we adopt an expanding window approach. Specifically, the first 15 years of data serve as the training set, while the following 5 years form the validation set, used for selecting tuning parameters, including the learning rate and pruning ratio.[8] We do not tune the number of factors but report results for each specified number. Performance is then evaluated in the subsequent year. This procedure is repeated 25 times, with the training set expanding by one year and the validation set shifting accordingly at each iteration. For comparison, we implement the PCA-based prediction procedure outlined in Haddad et al. (2020). Since PCA involves one tuning parameter—the number of factors—we do not tune it, instead combining the training and validation sets to run PCA.

Table 4 reports the average out-of-sample R^2 of predicted factor returns, benchmarked against the factor's average return over each expanding window. Additionally, we construct a long-short portfolio by taking long positions in the top 10 (20%) factors and short positions in the bottom 10 factors based on the predicted values.

For BM, the maximum R^2 achieved using the PCA-based approach is 0.599%, attained with 4 PCs, which translates to an annualized Sharpe ratio of 0.452. Using MOM1 as the input for PCA yields slightly lower positive R^2 values, but achieves a higher Sharpe ratio of approximately 0.654. However, the performance of PCA with MOM6 and MOM12 inputs is underwhelming, delivering negative out-of-sample R^2 values and low Sharpe ratios—less than 0.23 for MOM6 and 0.357 for MOM12. SAEs outperform PCA in nearly all comparable cases, often by a substantial margin in terms of out-of-sample R^2 values. For instance, the investment strategy using MOM1 as input achieves a Sharpe ratio of 1.01 with the SAE3 model and 4 factors. Similarly, for BM, the SAE3 model achieves a Sharpe ratio of 0.675. When MOM6 and MOM12 are used as inputs, SAE models reach Sharpe ratios of 0.759 and 0.834, respectively. The strong positive predictability associated with MOM1 confirms

[8] Empirically, our learning rate varies over the set {0.0005, 0.001, 0.005, 0.01}, and the pruning ratio ranges from 0.05 to 0.95 in increments of 0.05.

the evidence of factor momentum, particularly evident in returns sorted by the short-term factor returns, aligning with the findings of Arnott et al. (2023).

5.3 Causal Analysis with Corrupted Data

In the final exercise, we revisit the study by Agarwal et al. (2021) on casual analysis with corrupted data, focusing on a specific type of noise introduced for differential privacy, represented by Laplacian noise deliberately added to protect respondent privacy.

The economic context involves recovering the effect of import competition from China on U.S. labor markets, as studied in the influential work by Autor et al. (2013). Their panel dataset is organized at the commuting zone (CZ) level, encompassing 722 CZs across two time periods: the 1990s and 2000s. Each CZ is represented by a vector of 30 covariates, including variables from the authors' preferred specification as well as auxiliary variables detailed in their appendix.[9]

The dataset features repeated measurements of underlying economic factors, exhibiting a strong factor structure. Table 5 highlights this low-dimensional structure by comparing the variance explained by AEs and PCA. AEs consistently recover more variance than PCA for a given number of factors. Notably, while PCA requires over 13 factors to achieve 90% explanatory power, AEs need only 5–7 components.

Following Agarwal et al. (2021), we add synthetic Laplacian noise to the original dataset, referred to as the clean data, and compare the performance of PCA and AEs in denoising the corrupted inputs. The denoised outputs are then fed into the same causal analysis framework to evaluate the effects of these data-cleaning methods.

Specifically, we introduce noise into the original 1990s dataset, $X^\star \in \mathbb{R}^{30\times722}$, to obtain the corrupted data X, with SNRs set at 0.5, 1, and 2. PCA and AEs are applied to X and the MSE between the original data $X^\star$ and the reconstructed output $\widehat{X}$ is computed.

Figure 7 presents the average reconstruction error across all variables. Consistent with our simulation studies, AEs demonstrate superior performance in recovering the underlying data, achieving significantly smaller MSEs compared to PCA. As the SNR decreases, performance

[9] These variables are drawn from Column 6 in Table 3 and Appendix Table 2 of Autor et al. (2013). They include percentages of employment in manufacturing, college-educated population, and foreign-born population; percentages of employment among women and in routine occupations; average offshorability index of occupations; Census division dummies; and percentages of the working-age population: employed in manufacturing, employed in non-manufacturing, unemployed, not in the labor force, and receiving disability benefits. Additional variables include average log weekly wages (manufacturing and non-manufacturing), average benefits per capita (individual transfers, retirement, disability, medical, federal income assistance, unemployment, and TAA), and average household income per working-age adult (total and wage/salary).

Panel A: Out-of-sample R^2s											
	BM						MOM1				
K_1	1	2	3	4	5	K_1	1	2	3	4	5
PCA	-0.34	0.063	0.187	0.599	0.32	PCA	0.099	0.444	0.145	0.458	0.561
SAE1	0.408	0.473	0.361	0.47	0.645	SAE1	0.492	0.528	0.444	0.385	0.696
SAE2	0.429	0.548	0.54	0.638	0.563	SAE2	0.238	0.355	0.506	0.569	0.572
SAE3	0.53	0.473	0.54	0.695	0.54	SAE3	0.222	0.208	0.41	0.774	0.666
	MOM6						MOM12				
PCA	-0.009	-0.196	-0.235	-0.101	-0.031	PCA	-0.086	-0.415	-0.383	-0.011	-0.0
SAE1	0.457	0.556	0.595	0.653	0.612	SAE1	0.404	0.448	0.548	0.526	0.666
SAE2	0.438	0.316	0.637	0.605	0.637	SAE2	0.377	0.44	0.532	0.535	0.621
SAE3	0.335	0.204	0.605	0.559	0.662	SAE3	0.503	0.556	0.492	0.569	0.505
Panel B: Out-of-sample Sharpe Ratios											
	BM						MOM1				
PCA	0.158	0.351	0.33	0.452	0.425	PCA	0.133	0.529	0.475	0.654	0.597
SAE1	0.66	0.7	0.673	0.486	0.663	SAE1	0.662	0.861	0.676	0.806	0.856
SAE2	0.633	0.393	0.507	0.52	0.484	SAE2	0.512	0.652	0.704	0.748	0.802
SAE3	0.558	0.586	0.46	0.675	0.564	SAE3	0.744	0.662	0.809	1.007	0.779
	MOM6						MOM12				
PCA	-0.062	0.078	0.094	0.226	0.211	PCA	0.155	0.12	0.065	0.357	0.298
SAE1	0.757	0.759	0.636	0.572	0.551	SAE1	0.561	0.663	0.579	0.687	0.661
SAE2	0.612	0.515	0.525	0.507	0.706	SAE2	0.645	0.76	0.831	0.396	0.534
SAE3	0.41	0.529	0.611	0.673	0.519	SAE3	0.834	0.802	0.751	0.658	0.373

Table 4: Out-of-sample R^2s and Annualized Sharpe Ratio

Note: The table summarizes the empirical performance of predicting the cross-section of expected factor returns. The upper panel presents out-of-sample R^2 values (in percentages) for SAEs and PCA across different numbers of factors, using BM, MOM1, MOM6, and MOM12 as predictors. The lower panel reports Sharpe ratios for long-short portfolios constructed from sorted next-month predictions, rebalanced monthly.

K_1	1	3	5	7	9	11	13	15	17	19
PCA	24.1	50.9	64.9	74.1	81.7	87.3	90.8	93.8	96.3	97.7
AE1	38.6	76.7	87.4	92.3	94.7	96.1	97.2	98.2	98.9	98.9
AE2	44.5	82.0	91.1	93.8	95.8	97.1	98.1	98.4	99.1	99.2
AE3	51.2	86.2	92.4	95.3	96.7	97.6	98.0	98.7	99.2	99.3

Table 5: Cumulative Percentage of Variance Explained by Extracted Factors

Note: This table reports the cumulative percentage of variance explained by an incremental number of factors for PCA and AEs, respectively.

deteriorates, resulting in larger reconstruction errors. We next explore how these errors affect the final causal analysis. For consistency with Autor et al. (2013), we use their specified

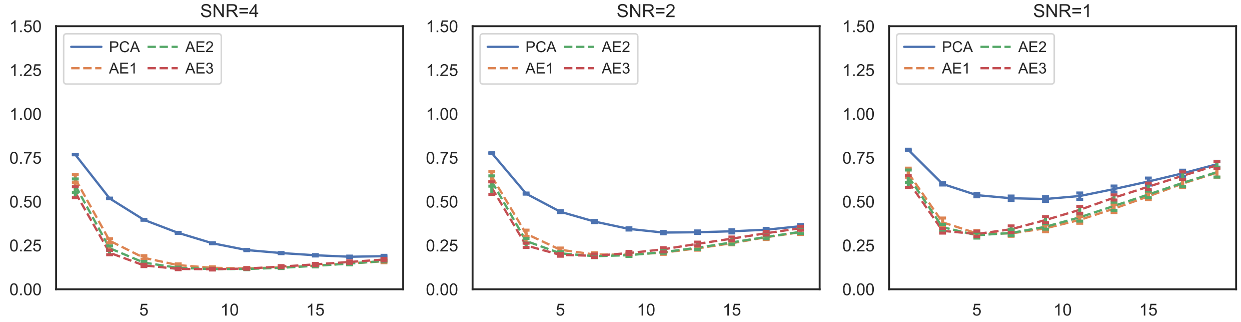

Figure 7: Effectiveness in Eliminating Synthetic Measurement Error

Note: This figure compares the MSEs for AEs and PCA across different numbers of factors in eliminating synthetic errors from the covariates.

variables for the two-stage least squares (TSLS) estimation, although the data-cleaning step benefits from the inclusion of auxiliary variables, which enhance the recovery of the factor structure.

For consistency with Autor et al. (2013), we use their specified variables for the two-stage least squares (TSLS) estimation, while the data-cleaning step benefits from the inclusion of auxiliary variables to enhance the recovery of the factor structure. We repeat the denoising process 100 times and present the distribution of TSLS estimates based on AEs and PCA, with the number of factors fixed at $K_1 = 5$ in Figure 8. For comparison, we also compute the causal effects directly using the noisy data without any cleaning. When applying TSLS to the outputs of AEs and PCA, the estimated causal effects are close to the value reported in the original paper ($-0.596 \pm 1.96 \cdot 0.099$): specifically, -0.548 for AE1, -0.589 for AE2, -0.625 for AE3, and -0.658 for PCA. Nevertheless, as shown in Figure 8, the results from AEs yield more accurate estimates of the causal effect, as they are closer to the red dashed line, which represents the causal effect derived from clean data.

6 Conclusion

This paper establishes a theoretical foundation for a nonparametric unsupervised learning problem—the application of deep AEs within nonlinear factor models—demonstrating their effectiveness in extracting latent common components from high-dimensional inputs. By extending this framework to include SAEs, we pave the way for broader and more versatile

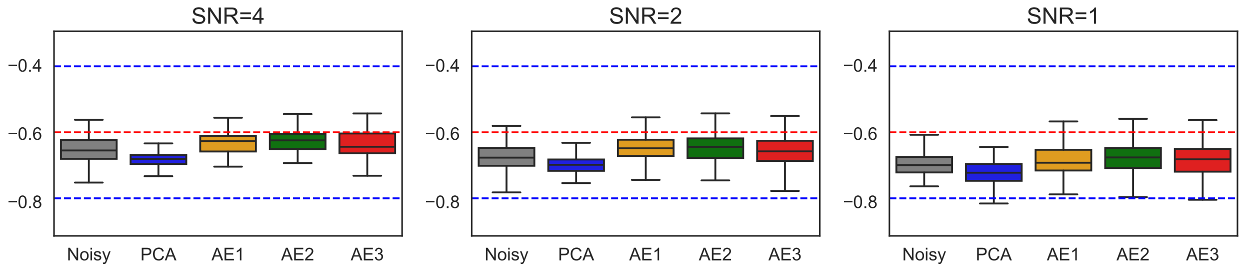

Figure 8: Effect of Noise Elimination on Causal Analysis Outcomes

Note: This figure presents boxplots of the TSLS estimates based on 100 repeated experiments with synthetic noise added to the covariates. The "Noisy" case represents the estimates obtained directly from the corrupted data, while the remaining boxplots correspond to estimates after applying respective data-cleaning methods using PCA and AEs.

applications in economics. These models equip researchers and practitioners with robust tools to address the complexities of analyzing large-scale and intricate datasets, unlocking new opportunities for both predictive accuracy and deeper explanatory insights.

While this work primarily focuses on estimation and prediction using a specific class of AEs and SAEs, it opens several promising directions for future research. A natural extension involves formal inference, particularly in the context of causal analysis, to better quantify uncertainty in causal effects. Expanding the framework to include more general AE structures beyond the disjoint output class could enhance their flexibility and applicability across diverse settings. Furthermore, developing principled methodologies for model selection, architecture optimization, and regularization strategies would improve both interpretability and computational efficiency. Finally, pursuing theoretical analyses to understand the integration of AEs with other machine learning frameworks, such as generative models or variational inference, could unlock opportunities for novel and distinct economic applications.

References

A. Abadie. Using synthetic controls: Feasibility, data requirements, and methodological aspects. *Journal of Economic Literature*, 59(2):391–425, June 2021.

A. Abadie, A. Diamond, and J. Hainmueller. Synthetic control methods for comparative case studies: Estimating the effect of california's tobacco control program. *Journal of the American statistical Association*, 105(490):493–505, 2010.

A. Agarwal, D. Shah, D. Shen, and D. Song. On robustness of principal component regression. *Journal of the American Statistical Association*, 116(536):1731–1745, 2021.

S. C. Ahn and A. R. Horenstein. Eigenvalue ratio test for the number of factors. *Econometrica*, 81(3):1203–1227, 2013.

F. Akbas, W. J. Armstrong, S. Sorescu, and A. Subrahmanyam. Smart money, dumb money, and capital market anomalies. *Journal of Financial Economics*, 118(2):355–382, 2015.

Y. Amemiya and I. Yalcin. Nonlinear Factor Analysis as a Statistical Method. *Statistical Science*, 16(3):275 – 294, 2001.

R. D. Arnott, V. Kalesnik, and J. T. Linnainmaa. Factor momentum. *The Review of Financial Studies*, 36(8):3034–3070, 01 2023.

S. Athey, M. Bayati, N. Doudchenko, G. Imbens, and K. Khosravi. Matrix completion methods for causal panel data models. *Journal of the American Statistical Association*, 116, 10 2017.

D. H. Autor, D. Dorn, and G. H. Hanson. The china syndrome: Local labor market effects of import competition in the united states. *American economic review*, 103(6):2121–2168, 2013.

J. Bai. Inferential theory for factor models of large dimensions. *Econometrica*, 71(1):135–171, 2003.

J. Bai and S. Ng. Determining the number of factors in approximate factor models. *Econometrica*, 70(1):191–221, 2002.

J. Bai and S. Ng. Matrix completion, counterfactuals, and factor analysis of missing data. *Journal of the American Statistical Association*, 116(536):1746–1763, 2021.

J. Bai and S. Ng. Approximate factor models with weaker loadings. *Journal of Econometrics*, 235(2):1893–1916, 2023.

P. Baldi and K. Hornik. Neural networks and principal component analysis: Learning from examples without local minima. *Neural Networks*, 2:53–58, 1989.

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

P. L. Bartlett, P. M. Long, G. Lugosi, and A. Tsigler. Benign overfitting in linear regression. *Proceedings of the National Academy of Sciences*, 117:30063 – 30070, 2019.

B. Bauer and M. Kohler. On deep learning as a remedy for the curse of dimensionality in nonparametric regression. *The Annals of Statistics*, 2019.

B. S. Bernanke and J. Boivin. Monetary policy in a data-rich environment. *Journal of Monetary Economics*, 50(3):525–546, 2003.

H. Bourlard and Y. Kamp. Auto-association by multilayer perceptrons and singular value decomposition. *Biological cybernetics*, 59(4):291–294, 1988.

G. Chamberlain and M. Rothschild. Arbitrage, factor structure, and mean-variance analysis on large asset markets. *Econometrica*, 51:1281–1304, 1983.

X. Chen and X. Shen. Sieve extremum estimates for weakly dependent data. *Econometrica*, 66(2):289–314, 1998.

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

Y. Chen, X. Li, and S. Zhang. Structured latent factor analysis for large-scale data: Identifiability, estimability, and their implications. *Journal of the American Statistical Association*, 115:1756 – 1770, 2017.

Z. Cheng, H. Sun, M. Takeuchi, and J. Katto. Deep convolutional autoencoder-based lossy image compression. In *2018 Picture Coding Symposium (PCS)*, pages 253–257. IEEE, 2018.

R. B. Cohen, C. Polk, and T. Vuolteenaho. The value spread. *The Journal of Finance*, 58 (2):609–641, 2003.

G. Connor and R. A. Korajczyk. Performance measurement with the arbitrage pricing theory: A new framework for analysis. *Journal of financial economics*, 15(3):373–394, 1986.

M. J. Cooper, R. C. Gutierrez Jr., and A. Hameed. Market states and momentum. *The Journal of Finance*, 59(3):1345–1365, 2004.

K. Daniel and T. J. Moskowitz. Momentum crashes. *Journal of Financial Economics*, 122 (2):221–247, 2016.

J. Etezadi-Amoli and R. P. McDonald. A second generation nonlinear factor analysis. *Psychometrika*, 48(3):315–342, 1983.

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

Y. Feng. Optimal estimation of large-dimensional nonlinear factor models, 2023.

J. Frankle and M. Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv: Learning*, 2018.

H. Freeman and M. Weidner. Linear panel regressions with two-way unobserved heterogeneity. *Journal of Econometrics*, 237(1):105498, 2023.

L. Gondara. Medical image denoising using convolutional denoising autoencoders. In *2016 IEEE 16th international conference on data mining workshops (ICDMW)*, pages 241–246. IEEE, 2016.

M. Griebel and H. Harbrecht. Approximation of bi-variate functions: Singular value decomposition versus sparse grids. *IMA Journal of Numerical Analysis*, 34, 01 2014.

S. Gu, B. Kellly, and D. Xiu. Autoencoder asset pricing models. *Journal of Econometrics*, 222:429–450, 2021.

A. Habibian, T. v. Rozendaal, J. M. Tomczak, and T. S. Cohen. Video compression with rate-distortion autoencoders. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

V. Haddad, S. Kozak, and S. Santosh. Factor timing. *The Review of Financial Studies*, 33: 1980–2018, 05 2020.

D. L. Hanson and F. T. Wright. A Bound on Tail Probabilities for Quadratic Forms in Independent Random Variables. *The Annals of Mathematical Statistics*, 42(3):1079 – 1083, 1971.

B. Hassibi and D. Stork. Second order derivatives for network pruning: Optimal brain surgeon. In S. Hanson, J. Cowan, and C. Giles, editors, *Advances in Neural Information Processing Systems*, volume 5. Morgan-Kaufmann, 1992.

T. Hastie, A. Montanari, S. Rosset, and R. J. Tibshirani. Surprises in high-dimensional ridgeless least squares interpolation. *The Annals of Statistics*, 50(2):949 – 986, 2022.

K. He, X. Chen, S. Xie, Y. Li, P. Dollar, and R. Girshick. Masked autoencoders are scalable vision learners. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15979–15988, Los Alamitos, CA, USA, jun 2022. IEEE Computer Society.

G. E. Hinton and R. Zemel. Autoencoders, minimum description length and helmholtz free energy. *Advances in neural information processing systems*, 6, 1993.

P. Huber, E. Ronchetti, and M.-P. Victoria-Feser. Estimation of generalized linear latent variable models. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 66(4):893–908, 2004.

G. W. Imbens and D. B. Rubin. *Causal Inference in Statistics, Social, and Biomedical Sciences*. Cambridge University Press, 2015.

Y. Jiao, G. Shen, Y. Lin, and J. Huang. Deep nonparametric regression on approximate manifolds: Nonasymptotic error bounds with polynomial prefactors. *The Annals of Statistics*, 51(2):691–716, 2023.

D. A. Kenny and C. M. Judd. Estimating the nonlinear and interactive effects of latent variables. *Psychological Bulletin*, 96(1):201–210, 1984.

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

D. P. Kingma and M. Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations (ICLR)*, 2014.

M. Kohler and S. Langer. On the rate of convergence of fully connected deep neural network regression estimates. *The Annals of Statistics*, 49(4):2231 – 2249, 2021.

L. Le, A. Patterson, and M. White. Supervised autoencoders: Improving generalization performance with unsupervised regularizers. *Advances in neural information processing systems*, 31, 2018.

Y. LeCun. *Connexionist Learning Models*. Phd thesis, Universite Pierre et Marie Curie (Paris), 1987.

Y. LeCun, J. Denker, and S. Solla. Optimal brain damage. In D. Touretzky, editor, *Advances in Neural Information Processing Systems*, volume 2. Morgan-Kaufmann, 1989.

H. Liu, A. Havrilla, R. Lai, and W. Liao. Deep nonparametric estimation of intrinsic data structures by chart autoencoders: Generalization error and robustness. *Applied and Computational Harmonic Analysis*, 68:101602, 2024.

X. Lu, Y. Tsao, S. Matsuda, and C. Hori. Speech enhancement based on deep denoising autoencoder. In *Interspeech*, volume 2013, pages 436–440, 2013.

A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey. Adversarial autoencoders, 2016.

J. Masci, U. Meier, D. Cireşan, and J. Schmidhuber. Stacked convolutional auto-encoders for hierarchical feature extraction. In *Artificial Neural Networks and Machine Learning–ICANN 2011: 21st International Conference on Artificial Neural Networks, Espoo, Finland, June 14-17, 2011, Proceedings, Part I 21*, pages 52–59. Springer, 2011.

M. W. McCracken and S. Ng. Fred-md: A monthly database for macroeconomic research. *Journal of Business & Economic Statistics*, 34(4):574–589, 2016.

R. McDonald. A general approach to nonlinear factor analysis. *Psychometrika*, 27(4):397–415, 1962.

S. Mei, A. Montanari, and P.-M. Nguyen. A mean field view of the landscape of two-layer neural networks. *Proceedings of the National Academy of Sciences*, 115(33):E7665–E7671, 2018.

S. Mei, T. Misiakiewicz, and A. Montanari. Mean-field theory of two-layers neural networks: dimension-free bounds and kernel limit. In A. Beygelzimer and D. Hsu, editors, *Proceedings of the Thirty-Second Conference on Learning Theory*, volume 99 of *Proceedings of Machine Learning Research*, pages 2388–2464. PMLR, 25–28 Jun 2019.

I. Moustaki and M. Knott. Generalized latent trait models. *Psychometrika*, 65:391–411, 2000.

R. Nakada and M. Imaizumi. Adaptive approximation and generalization of deep neural network with intrinsic dimensionality. *Journal of Machine Learning Research*, 21(174): 1–38, 2020.

W. K. Newey. Convergence rates and asymptotic normality for series estimators. *Journal of Econometrics*, 79(1):147–168, 1997.

A. Ng et al. Sparse autoencoder. *CS294A Lecture notes*, 72(2011):1–19, 2011.

A. Onatski. Determining the number of factors from empirical distribution of eigenvalues. *Review of Economics and Statistics*, 92, 02 2005.

S. Rifai, P. Vincent, X. Muller, X. Glorot, and Y. Bengio. Contractive auto-encoders: Explicit invariance during feature extraction. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, pages 833–840, Madison, WI, USA, 2011. Omnipress.

M. Rudelson and R. Vershynin. Hanson-wright inequality and sub-gaussian concentration. *Electronic communications in probability*, 18, 06 2013.

J. Schmidt-Hieber. Nonparametric regression using deep neural networks with ReLU activation function. *The Annals of Statistics*, 48(4):1875 – 1897, 2020.

J. Schmidt-Hieber. The kolmogorov–arnold representation theorem revisited. *Neural Networks*, 137:119–126, 2021.

A. Skrondal and S. Rabe-Hesketh. *Generalized Latent Variable Modeling: Multilevel, Longitudinal, and Structural Equation Models*. CRC Press, Boca Raton, FL, 2004.

P. Speckman. Spline Smoothing and Optimal Rates of Convergence in Nonparametric Regression Models. *The Annals of Statistics*, 13(3):970 – 983, 1985.

R. F. Stambaugh, J. Yu, and Y. Yuan. The short of it: Investor sentiment and anomalies. *Journal of Financial Economics*, 104(2):288–302, 2012. Special Issue on Investor Sentiment.

J. H. Stock and M. W. Watson. Forecasting inflation. *Journal of monetary economics*, 44 (2):293–335, 1999.

J. H. Stock and M. W. Watson. Forecasting using principal components from a large number of predictors. *Journal of the American Statistical Association*, 97(460):1167–1179, 2002.

C. J. Stone. Optimal Global Rates of Convergence for Nonparametric Regression. *The Annals of Statistics*, 10(4):1040 – 1053, 1982.

L. Theis, W. Shi, A. Cunningham, and F. Huszár. Lossy image compression with compressive autoencoders. In *International Conference on Learning Representations*, 2017.

A. Tsigler and P. L. Bartlett. Benign overfitting in ridge regression. *J. Mach. Learn. Res.*, 24(1), Mar. 2024.

M. Udell and A. Townsend. Why are big data matrices approximately low rank? *SIAM Journal on Mathematics of Data Science*, 1(1):144–160, 2019.

P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol. Extracting and composing robust features with denoising autoencoders. ICML '08, pages 1096–1103, New York, NY, USA, 2008. Association for Computing Machinery.

F. Wang. Maximum likelihood estimation and inference for high dimensional generalized factor models with application to factor-augmented regressions. *Journal of Econometrics*, 229(1):180–200, 2022.

L. Wei, H. Lin, S. Zheng, and J. Liu. Generalized factor model for ultra-high dimensional correlated variables with mixed types. *Journal of the American Statistical Association*, 118:1–42, 10 2021.

J. Xie, L. Xu, and E. Chen. Image denoising and inpainting with deep neural networks. In F. Pereira, C. Burges, L. Bottou, and K. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.

J. Xu. Rates of convergence of spectral methods for graphon estimation. In *International Conference on Machine Learning*, 2017.

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

H.-T. Zhu and S.-Y. Lee. Statistical analysis of nonlinear factor analysis models. *British Journal of Mathematical and Statistical Psychology*, 52(2):225–242, 1999.