Deep Learning in Modelling Exchange Rate

*Peng Wei [†] Yi Cao [‡] Yizhe Dong [§]

February, 2022

Abstract

We investigate empirical asset pricing in the foreign exchange market from the perspective of cross-sectional description and time series prediction in the context of deep learning. We propose two models, the deep latent factor model and the deep prediction model. The former focuses on cross-sectional description and maintains the economic interpretation of factor models with the no-arbitrage restriction, while the latter focuses on pure prediction without considering the risk-return trade-off. An architecture of sequence modelling is employed in both models as a homogeneous constituent to extract historical information and incorporate cross-country interactions. We conduct a comprehensive comparative analysis of model performance across different architectures and complexities. Both models yield the best performance in cross section and time series respectively, show clear cross-country interaction patterns, agree on the same groups of influential characteristics that consistently dominate, and highlight the beneficial effects of incorporating long-range historical data for both problems. Our study provides a complete spectrum of how deep learning can be used to model exchange rate return with or without considering the economic interpretation.

JEL code: C45, C58, G12
Keywords: Exchange rates, Latent factor model, Prediction model, Deep learning, Attention network.

*We thank Stephanie Chan, Alistair Haig, Hao Li, Brendan Walsh for helpful comments. Peng Wei gratefully acknowledge financial support from the Franklin Templeton Investment Management.

[†]Management Science and Business Economics Group, Business School, University of Edinburgh, 29 Buccleuch Place, Edinburgh EH8 9JS, UK. Email: p.wei@ed.ac.uk.

[‡]Management Science and Business Economics Group, Business School, University of Edinburgh, 29 Buccleuch Place, Edinburgh EH8 9JS, UK. Email: jason.caoyi@gmail.com.

[§]Management Science and Business Economics Group, Business School, University of Edinburgh, 29 Buccleuch Place, Edinburgh EH8 9JS, UK. Email: yizhe.dong@ed.ac.uk.

1 Introduction

Great interest in using comprehensive complex characteristics to model asset returns has been arisen from the latest development of machine learning methods. These methods are studied from two canonical perspectives, namely cross-section (Kelly et al. (2019); Gu et al. (2021)) and time series (Fischer and Krauss (2018); Filippou et al. (2020)). For decades, these two themes have dominated empirical asset pricing research, with the former attempting to characterize and explain variations in asset returns, while the latter attempting to predict market risk premiums in time series.

The most common framework for describing the cross-sectional asset return is the factor model with observable factors based on previously established knowledge about average return fluctuations [1]. Recent literature opens a new strand that allows latent factors and makes no ex-ante assumptions regarding their definition. In latent factor models, the unobservable factors and factor loadings are estimated concurrently using statistical methods such as principal components analysis (PCA) [2]. Boosted by the burgeoning popularity of machine learning in finance, latent factor models have provoked considerable discussion over the recent years.

Kelly and Xiu (2021) document the remarkable effectiveness of machine learning methods in revealing complex nonlinear relationships and overcoming limitations of traditional latent factor via their flexible functional forms and ability to deal with large data sets. As prominent examples, Lettau and Pelger (2020a,b) generalize PCA to RP-PCA with a penalty term to account for the pricing errors in the means, indicating that RP-PCA can detect weak factors and dominates PCA in estimating latent factors even with large datasets; Kelly et al. (2019) and Gu et al. (2021) incorporate a large collection of equity characteristics that serve as conditioning information for the time-varying betas, using a linear functional form of instrumented principal components analysis (IPCA) and an extended nonlinear autoencoder, respectively; Chen et al. (2020) use generative adversarial network (GAN) and long short-term memory (LSTM) to account for massive amounts of conditioning information and temporal variation.

Different from factor models, which attempt to reconstruct asset returns with pre-specified risk exposures and risk premia structure, return prediction in time series, on the other hand, focuses exclusively on forecasting future terms without considering the risk-return structure (see the work of Fischer and Krauss (2018); Filippou et al. (2020)). Gu et al. (2018) investigate equity asset pricing using machine learning methods in both the cross-section and time series.

However, in the Forex market, forecasting return is a notoriously difficult task compared to the equity market. The commonly known Meese-Rogoff puzzle suggests that economic models underperform in out-of-sample forecasting of major bilateral exchange rates compared to the random walk models (Meese and Rogoff (1983a,b)) [3]. To date, no single model has yet to emerge as the best forecasting model. Some literature examine the instability and non-linearity of the relationship between macroeconomic fundamentals and exchange rate fluctuations (e.g., Stock and Watson (1996), Stock and Watson (1999), Rossi et al. (2006) and Engel et al. (2007).), while others contribute to predictor selection, such as prices, money supplies and output (e.g., Mark and Sul (2001) and Kilian and Taylor (2003)). A more expansive list of predictors and broader functional forms by machine learning for forecasting exchange rate are investigated in recent studies, such as Engel et al. (2015), which investigate a combination of fundamentals and principal component extracted factors, and Filippou et al. (2020), which document the outperformance of standard neural network in forecasting exchange rates using a rich set of predictors.

To date, the literature of using machine learning models to predict or explain the exchange rate return is far less than the ones on equity market. A great lack of thorough methodological or empirical work on this topic leaves three outstanding questions: 1) Cross section or time series: The cross-sectional model structure of latent factor has been less studied in Forex market (see work on equity market Kelly and Xiu (2021); Gu et al. (2021); Kelly et al. (2019)). Meanwhile, Filippou et al. (2020) assume homogeneous parameter

[1] Backed by the arbitrage pricing theory (APT) (Ross (1976)), factor models describe the cross-sectional asset returns from the perspective of risk exposures and risk premium and have been workhorses for modelling the cross-sectional variations for decades. In accordance with APT's economic interpretation, factor models seek to provide a compact statistical description of assets' cross-sectional risk-return structure. In this context, the estimation and interpretation of factor models stand out as the central topic in this field. Factor model has constituted an extensive literature, see, for example, Fama and French (1993), Carhart (1997) and Hou et al. (2015).

[2] See Connor and Korajczyk (1986) and Bai and Ng (2002), among others.

[3] See a number of possible explanations Isard et al. (1983), Schinasi and Swamy (1989) and Moosa (2013), among others.

weights across countries in the time series model structure. However, this hypothetical simplification has no obvious theoretical or intuitive justification. It is unclear which model structures produce the most prominent exchange rate explanation and forecast. 2) Partial or full interaction: The work of Gu et al. (2018, 2021); Filippou et al. (2020) address the interaction between macroeconomic fundamentals and asset characteristics by deterministic Kronecker product, while neglecting interactions among assets. However, recent literature Branger et al. (2021); Gofman et al. (2020); Barunik et al. (2020) suggest that interactions among assets can be propagated through complex networks and therefore be non-negligible. No existing work addresses architecture with full interactions among the characteristics of assets through a non-deterministic "data-defined" format yet. 3) One period or long memory: most studies assume that the return is predicted by a function of factors, loadings or predictors that are at present or one-period lagged. There is a lack of a comprehensive model structure capable of exploiting long sequence of predictors and extracting representative historical information embedded in the temporal correlations of predictors.

In this paper, with the aim to answer the three outstanding questions, we explore the asset pricing problem in the Forex market from two canonical perspectives, namely cross-sectional description and time series prediction. To accomplish it, we extend the factor model in Gu et al. (2021) and redesign the deep prediction model in Filippou et al. (2020), respectively. We extract the representative historical dynamics from a rich set of predictors by an attention network and incorporate interaction effects over all countries by fully connected network structures. Although sharing some homogeneous constituent architectures, the structure of the two models stems from completely different underlying assumptions. With the deep prediction model primarily capturing complex nonlinear relationships, the latent factor model also maintains economic interpretations on this basis.

In our cross-sectional study, we extend the work of Kelly et al. (2019) and Gu et al. (2021) and propose the Deep Latent Factor (DLF) model. The asset pricing model proposed by Kelly et al. (2019) employs an instrumented PCA that allows the factor loadings to incorporate and linearly condition on a broad set of stock-level characteristics. Gu et al. (2021) extends their work by allowing the factor loadings to nonlinearly condition on the stock characteristics using a conditional autoencoder neural network. The conditional autoencoder captures information from one-period lagged characteristics and naturally assumes these factors, which are, economically, a portfolio of individual asset returns, are static linear combinations of characteristic-managed portfolios between every two refits.[4]

Our deep latent factor model extends the conditioning information from one-period lagged characteristics to a longer look-back window and extracts state variables via sequence modelling techniques. The factor portfolio weights are nonlinearly conditioned on the historical characteristics with no ex-ante assumptions. Additionally, we adapt to the Forex intuition and expand the characteristic set to include all common and country-specific characteristics in the cross section, allowing cross-country interaction effects.

From the perspective of time series prediction, we propose a deep prediction model that forecasts exchange rate fluctuations in the next period. Our model architecture improves the deep neural network model in Filippou et al. (2020), a close predecessor to our work, mainly in two ways. First, we employ time-series attention network (sequence modelling techniques) in our architecture to incorporate information from historical fundamental characteristics in a longer window, which is proved effective in subsequent empirical studies. Second, our model allows cross-country interactions and learns a separate set of parameters for each currency in the training step, instead of learning a common set of parameters shared across currencies trained by pooled data. Hence, we denote the deep Prediction model with Attention Network and Full Connection as PRED-TSAN-FC.

Although some homogeneous constituent architectures are shared, the two models are fundamentally different in their design initiatives. The deep latent factor (DLF) model is constructed incorporating the factor asset pricing model structure while the deep prediction model (PRED-TSAN-FC) mainly captures complex nonlinear relationships. Using those two models, we show the predictability of the exchange rate return in both perspectives. Moreover, we provide a complete spectrum of how deep learning can be used to model exchange rate return with or without economic interpretation.

In our empirical study of G10 currencies, we evaluate the models based on their contemporaneous ex-

[4]In the conditional autoencoder architecture, the factor layer is set to be a linear hidden layer, thus the coefficients in factor portfolios are weight parameters learned in the training step, which are only updated when refitting the model.

planatory performance, measured by the contemporaneous R^2 [5] and $RMSE$ [6], and predictive performance, measure by the predictive $RMSE$ and the economic value - Sharpe ratios of long-short carry trade portfolios. The deep latent factor model (DLF) provides an improved out-of-sample predictive performance compared to extant models in terms of both statistical and economic performance. The contemporaneous explanatory performance tends to rise as model complexity grows, whereas simpler architectures tend to yield better predictive performance. On the other hand, despite the use of all regularization and ensemble methods, the performance of the deep prediction model (PRED-TSAN-FC) is highly volatile and does not exhibit significant monotonicity with respect to the model complexity. When assessing the overall performance, considering the average performance over different model complexities, the deep prediction model outperforms comparative prediction models and provides a slightly better performance compared to the random walk benchmark.[7] A significant performance enhancement is suggested by both the deep latent factor model and the deep prediction model on the sequence modelling architecture, which is used to extract state variables from historical characteristics. In both models, the characteristic sensitivity reveals a distinct cross-country interaction pattern and a dominance of interest rate-related characteristics in terms of characteristic importance, which is consistent over time.

The contribution of our study is twofold. First, we contribute to the empirical asset pricing literature in the Forex market by an in-depth methodological study from two perspectives, namely cross-sectional description through a deep latent factor model (DLF) and time series prediction through a deep prediction model (PRED-TSAN-FC). The former maintains the economic interpretation of factor models, extending the work of Kelly et al. (2019) and Gu et al. (2021), while the latter aims to forecast future returns and improves the closest predecessor model of Filippou et al. (2020) in multiple aspects. We compare the models in terms of statistical and economic performance, and identify obvious cross-country interactions and consistent dominant characteristics in both the deep latent factor model and deep prediction model. Second, we conduct an in-depth comparative empirical analysis of neural network models with various architectures and complexities in both cross-sectional description and time-series prediction. We compare the statistical and economic performance of our candidate models and explore the effectiveness of different model specifications. We highlight the significant beneficial effects of employing sequence modelling architecture in both problems.

The remainder of the paper is organized as follows. Section 2 discusses the specification and estimation of the models. Section 3 presents our empirical studies of the deep latent factor model and deep prediction model. Section 4 concludes.

2 Methodology

This section describes the models that guide our empirical work. We first introduce the overall architectures of our deep latent factor model and deep prediction model in Section 2.1 and 2.2 respectively, with the former focused on cross-sectional variations in asset returns and the latter on forecasting future returns in time series. Then, we illustrate in detail the key constituent architectures employed by the two models, namely the time-series attention network and the feed-forward network in Section 2.3 and 2.4 respectively. Finally, in Section 2.5, we discuss the regularization techniques and loss functions that we employed to estimate the latent factor model and prediction model.

2.1 Deep Latent Factor Model

In this subsection, we focus on our deep latent factor model, which is designed to capture the cross-sectional return variations in the Forex market. It follows the latent factor strand of literature and extends the work of Kelly et al. (2019) and Gu et al. (2021).

[5] As defined in Kelly et al. (2019) and Gu et al. (2021).

[6] The root mean squared error is a widely employed metric in exchange rate prediction, as defined in Meese and Rogoff (1983a,b).

[7] The random walk without drift, also known as the no-change benchmark, has been a widely studied robust benchmark in the literature of exchange rate forecasting since Meese and Rogoff (1983a,b).

We start from the canonical form of the factor model. Let r_t^i represents the return of asset i in period t, for $i = 1, \ldots, N$ and $t = 1, \ldots, T$, a K-factor model can be described in mathematical form as:

$$r_t^i = {\beta^i}^\top f_t + u_t^i \tag{1}$$

where f_t is a $K \times 1$ vector of common factor returns in period t, β^i is a $K \times 1$ vector of factor loadings for asset i, u_t^i is the idiosyncratic component of r_t^i and is uncorrelated with factor returns f_t.

In the context of latent factors, the factor returns, factor loadings and idiosyncratic components are unobservable. The factors and factor loadings are estimated simultaneously from a panel of realized returns using statistical techniques such as PCA, which captures the cross-sectional behaviors based on information in asset returns. To incorporate a broader range of information beyond returns, Kelly et al. (2019); Gu et al. (2021) propose their model with the highest level description as

$$r_t^i = \beta(z_{t-1}^i)^\top f_t + u_t^i \tag{2}$$

where the factor exposure $\beta(z_{t-1}^i)$ is linearly (Kelly et al. (2019)) or non-linearly (Gu et al. (2021)) conditioned on an arbitrary length vector of one-period lagged asset characteristics z_{t-1}^i, that is, $\beta(z_{t-1}^i)^\top = {z_{t-1}^i}^\top \Gamma$. The linear implementation in Kelly et al. (2019) is termed as instrumented principal components analysis (IPCA), which allows the factor model to incorporate observable characteristics serving as instrumental variables for time-varying conditional loadings of the latent factors, while the nonlinearity in Gu et al. (2021) is implemented through a conditional autoencoder model with interaction effects of factor exposures. The conditional autoencoder is an unsupervised model from the autoencoder family and can be thought of as a nonlinear, neural network extension of IPCA. [8]

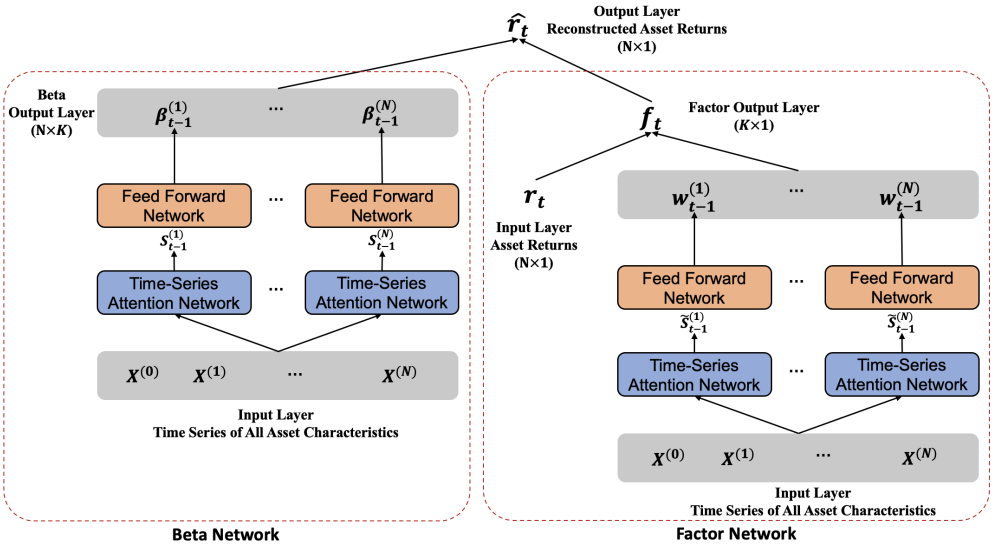
The economic interpretation of factors suggests that factors are portfolios (linear combinations) of individual asset returns. To maintain the linear structure of factors with a reasonable number of trainable parameters, Gu et al. (2021) employ a single linear layer on the factor network in their conditional autoencoder model, where a vector of characteristic-managed portfolio returns[9] passes through a single linear hidden layer to form the lower dimensional vector of factor returns. Limited by the high computation cost of machine learning models, it's hard to refit a model recursively in a monthly frequency, which also increases the propensity to overfitting to minor monthly variations in data. Thus, a common way in the literature is to refit the model on an annual basis, as discussed in Gu et al. (2021), Gu et al. (2018) and Cong et al. (2021a). However, for a neural network model, refitting on an annual basis means that the parameters of the model are fixed during the twelve months between two refits. In this case, the factors are naturally assumed to be static linear combinations of input asset returns with fixed coefficients within every twelve months, which is not theoretically or intuitively justified. The Gu et al. (2021) modification of initializing with characteristic-managed portfolio returns instead of individual stock returns alleviates the problem partially and reduce the number of trainable parameters substantially, but at the expense of an ex-ante assumption regarding the characteristic-managed portfolio weights and the pre-requisite of no common characteristics.

We make improvements to the factor model (2) in three aspects. First, we extend the conditioning information from one-period lagged asset characteristics to a longer look-back window by sequence modelling algorithms; more precisely, we summarize the information in historical asset characteristics as state variables by a low-dimensional vector of market hidden state. Second, we allow the factor portfolio weights to be conditioned on asset characteristics and vary dynamically over time without any ex-ante assumptions. Third, we expand the set of predictors for each currency to include all common and country-specific characteristics in the cross section, taking into account the interaction effects among different countries. This differs from most structural Forex pricing models, such as the monetary model, in which only the relevant two countries' fundamental differences are used as predictors in the bilateral exchange rate. Thus, our model can be described in mathematical form as:

[8] Autoencoder is an unsupervised dimensionality reduction technique that aims at learning a compressed representation for the input set of data. The idea of autoencoder has been popular in the neural network field for decades, serving as an effective tool to convert the high-dimensional data to low-dimensional code (Hinton and Salakhutdinov (2006) and Goodfellow et al. (2016)). A standard autoencoder neural network is typically constituted by two parts: an encoder network that maps the input data into a reduced set of representation code, which is the dimensionality reduction process, and then a decoder network that maps the learned reduced code back to a reconstruction of the input.

[9] The characteristic-managed portfolio returns are OLS estimates of regressing individual stock returns on the stock-level characteristics.

Figure 1: This figure shows the overall architecture of the Deep Latent Factor Model. The model consists of a Beta Network on the left side and a Factor Network on the right side. At the highest level, the model can be described in equation 4.



$$r_t^i = \beta(z_{t-1}, ..., z_{t-T})^\top f(r_t, z_{t-1}, ..., z_{t-T}) + u_t^i \tag{3}$$

where the conditional factor exposure is a neural network model of the lagged historical characteristics of all countries, and the factors are asset portfolios with their weights also conditioned on lagged historical characteristics of all countries and modelled by neural networks.

The remainder of this subsection discusses our model architecture in detail. Our deep latent factor model integrates hidden market states extracted from the dynamics of a diverse set of characteristics and allows for cross-country interactions. It maintains the economic interpretation of the linear factor model at the highest level, specifying factors as linear combinations of underlying asset returns with dynamic coefficients.

2.1.1 Overall Architecture

We start from the overall architecture of our deep latent factor model. Let N be the number of foreign currencies and K be the number of latent factors which is smaller than N. The exchange rates are expressed as the values of the domestic currency, in this paper US dollar, against the foreign currencies, computed as the number of foreign currency units per U.S. dollar (i.e., U.S. dollar as the base currency).[10]

Figure 1 illustrates our overall architecture, which consists of a Beta Network on the left side and a Factor Network on the right side. At the highest level, the model can be described in mathematical form as:

$$r_t = \beta_{t-1} f_t + u_t, \tag{4}$$

where r_t is a $N \times 1$ vector of individual exchange rate returns, β_{t-1} is a $N \times K$ matrix of factor loadings modeled by the Beta Network from lagged characteristics, f_t is a $K \times 1$ vector of latent factor returns in period t modeled by the Factor Network, u_t is a $N \times 1$ vector of idiosyncratic errors and is uncorrelated with f_t.

In our model architecture, both the factor loadings (in Beta network) and factors (in Factor networks) are conditioned on characteristics of all countries in the cross section, namely the U.S. and the N countries of foreign currencies, rather than conditioning on characteristics of the relevant two countries in the bilateral exchange rate as conventional monetary models.

There are two critical constituent architectures that are employed by both the Beta and Factor networks: the Time-Series Attention Network (TSAN) and the Feed-forward Network (FFN). The TSAN is designed

[10]The terms "domestic" or "foreign" do not refer to any geographical region, but rather to a particular side of the deal.

to extract the market hidden state from historical characteristics, and the FFN has been one of the most basic and quintessential networks in machine learning. The architectures of TSAN and FFN networks are discussed in depth in Section 2.3 and 2.4 respectively, so we skip the redundant illustration here.

In next two sections, we discuss the architectures of the Beta and Factor networks, respectively.

2.1.2 Beta Network

We first illustrate the input layer of our Beta Network. Let m denotes the number of characteristics for each foreign country and m^* denotes the number for the US. We name T as the length of our look-back window of historical characteristics; more precisely, at time t, we condition the Beta Network on the characteristics of all countries in period from time t–T to time $t-1$. Then, our input historical characteristics, denoted as $\{X^{(0)}, X^{(1)}, \ldots, X^{(N)}\}$, are a collection of time series where $X^{(0)}$ is an $m^* \times T$ matrix denoting the time series of common characteristics (including the U.S. characteristics), and $X^{(1)}, \ldots, X^{(N)}$ are $m \times T$ matrices denoting the historical characteristics of the N foreign countries.

In the Beta Network, we employ a fully connected structure between the input layer and the TSAN. The Beta Network maps the input historical characteristics of all countries, $\{X^{(0)}, X^{(1)}, \ldots, X^{(N)}\}$, into an output $N \times K$ matrix of factor loadings, β_{t-1}. For each currency, it entails an independent TSAN to extract the current hidden market state from the dynamics of historical characteristics. We denote $s_{t-1}^{(i)}$ as the extracted hidden state of currency i, where $i = 1, \ldots, N$. For each currency i, the extracted current hidden state then passes through an independent Feed Forward Network (FFN) to form a $K \times 1$ vector of factor loadings $\beta_{t-1}^{(i)}$. The output $N \times K$ matrix of factor loadings β_{t-1} is thus constructed by merely concatenating and reshaping the factor loadings $\beta_{t-1}^{(1)}, \ldots, \beta_{t-1}^{(N)}$.

2.1.3 Factor Network

The Factor Network aims to maintain the economic interpretation of factors, namely, factors are portfolios of individual currency returns and allow the portfolio weights to be conditional on countries' historical characteristics and vary dynamically over time.

To describe the Factor Network in mathematical form at the highest level:

$$f_t = W_{t-1} r_t, \tag{5}$$

where f_t is a $K \times 1$ vector of latent factor returns, W_{t-1} is a $K \times N$ matrix of factor portfolio weights conditioned on the time series of historical characteristics, r_t is a $N \times 1$ input vector of individual currency returns.

The Factor Network maps the input vector of individual currency returns r_t and historical characteristics $\{X^{(0)}, X^{(1)}, \ldots, X^{(N)}\}$ (the same as input of the Beta Network) to the output lower dimensional vector of latent factor returns f_t. The architecture is designed that each of the latent factor returns is a weighted combination of the input individual currency returns, with their weights learned from the historical asset characteristics and varying dynamically over time.

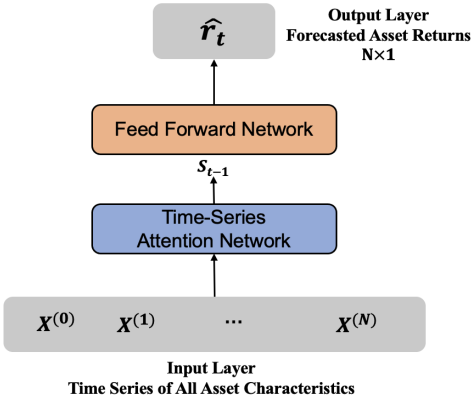
The Factor Network adopts a similar architecture as the Beta Network to model the portfolio weight matrix. For each currency i, where $i = 1, \ldots, N$, the hidden state $\tilde{s}_{t-1}^{(i)}$ extracted by the TSAN passes through a FFN to form a $K \times 1$ vector of portfolio weights, $w_{t-1}^{(i)}$. The N weight vectors are then stacked to form a $K \times N$ matrix of portfolio weights W_{t-1} for the construction of latent factor returns f_t.

At last, the final output $N \times 1$ vector of reconstructed asset returns $\hat{r}_t$ is then constructed by merely taking the inner product of the factor loadings β_{t-1} and factor returns f_t.

2.2 Deep Prediction Model

This subsection illustrates the architecture of our deep prediction model, which is designed to forecast the exchange rate fluctuations in time series. A key difference between this prediction model and the deep latent

Figure 2: This figure shows the architecture of the Deep Prediction Model. The TSAN architecture is to extract a vector of state variables from characteristics in a long history. A separate set of parameters is trained for each currency, with the predictor set containing characteristics of all countries in the cross section and allowing for cross-country interactions.



factor model in Section 2.1 is that in the deep prediction model, the currency returns are forecasted entirely based on the characteristics with no factor and loading structures. The objective of the prediction model is to forecast future exchange rate returns in time-series using lagged characteristics, not to explore the cross-sectional dependence structure of returns through return reconstruction.

The deep prediction model, at the highest level, can be described mathematically as:

$$r_t^i = E_{t-1}(r_t^i) + u_t^i, \tag{6}$$

where

$$E_{t-1}(r_t^i) = g(z_{t-1}, ..., z_{t-T}). \tag{7}$$

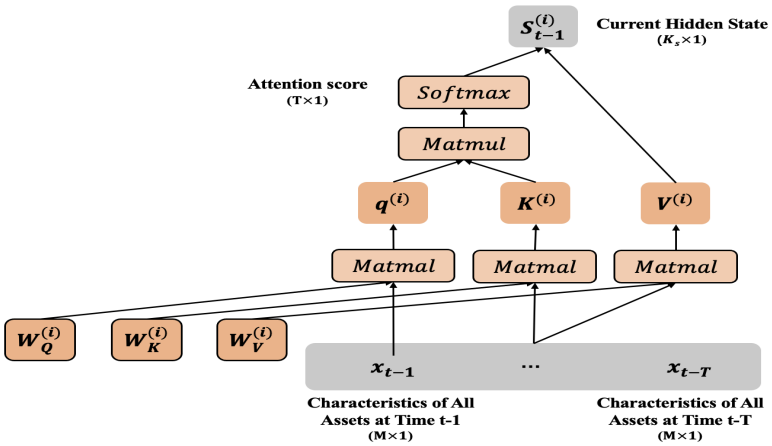
Our deep prediction model $g(\cdot)$ is a neural network function of the characteristic time series, with the architecture illustrated in Figure 2. The input historical characteristics $\{X^{(0)}, X^{(1)}, \ldots, X^{(N)}\}$, which is the same as the Beta Network input, pass through a single TSAN to extract the market's overall hidden state s_{t-1}. The $K_s \times 1$ hidden state vector is then fed into an FFN to generate a $N \times 1$ vector of forecasted individual currency returns $\hat{r}_t$. The architectures of the TSAN and FFN are identical to those mentioned in the deep latent factor model and are illustrated in detail in subsequent Section 2.3 and 2.4, respectively. Thus, our deep prediction model maps the historical characteristics of all countries to the forecasted returns for the next term.

A close predecessor to our work is Filippou et al. (2020). They consider an ensemble forecast, taking the average of a linear panel regression forecast and a deep neural network forecast, to predict monthly exchange rates for a group of developed countries based on a broad set of predictors. Despite some relevance with their deep neural network model, the architecture of our deep prediction model differs significantly in two aspects. The first distinction is that we employ a TSAN architecture to extract a vector of state variables from characteristics in a long history, instead of using one-period lagged characteristics as employed in Filippou et al. (2020). The second difference is that in our deep prediction model, a separate set of parameters is trained for each currency, with the predictor set containing characteristics of all countries in the cross section and allowing for cross-country interactions; while in Filippou et al. (2020)'s work, a common set of parameters is shared across currencies and trained using pooled data, with the predictor set containing solely the characteristics associated with the target currency.

2.3 Time-Series Attention Network (TSAN)

This section illustrates the architecture of our time-series attention network (TSAN) which is used to extract current hidden state from historical characteristics and is employed in both the deep latent factor model and

Figure 3: Time-Series Attention Network (TSAN)



deep prediction model introduced in Section 2.1 and 2.2.

In machine learning, sequence modelling typically involves dealing with sequences of data where the individual data points are strongly correlated and cannot be assumed as independent and identically distributed points. Well-known examples of sequence-to-sequence maps include speech recognition and machine translation, where contextual information from past inputs is critical for future outputs.

Recurrent neural networks (RNNs) and attention mechanisms are two approaches that have been firmly established as state of the art for sequence modelling tasks.[11] Recurrent neural networks, such as long short-term memory (LSTM)[12] and gated recurrent units (GRU)[13], generate a sequence of hidden states along the positions where the current hidden state is a function of the previous hidden states and the current input. The sequential nature of the computation naturally precludes parallelization within data, making it difficult to learn long range dependencies at longer sequence lengths. Attention mechanisms instead allow modelling of dependencies without considering their distance in the sequences thus enable parallel computation. Self-attention particularly learns a representation of a single sequence by relating different positions of the sequence.

The strong correlation within the time series of macroeconomic variables and the corresponding market environments makes the attention mechanisms an attractive choice for modelling the hidden dynamics of the market. Chen et al. (2020) has suggested the essentiality of extracting the hidden pattern in macroeconomic time series before feeding into machine learning models, where a LSTM is employed to estimate the hidden macroeconomic state variables. Cong et al. (2021a) introduce a Cross Asset Attention Network to learn the interrelationships among the assets from the historical states of all assets, and Cong et al. (2021b) overview the sequence modelling using neural networks and conduct a comparative analysis of the models in return forecasting.

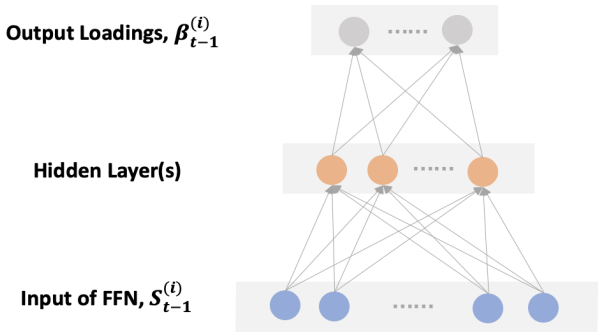
Here we introduce a Time-Series Attention Network (TSAN) which extracts the current hidden state of the market from the historical macroeconomic characteristics. In our model architecture as shown in Figure 1 and Figure 2, we employ the TSAN architecture in both our deep latent factor model and deep prediction model to extract the associated current hidden state of the market. In both models, the TSAN learns the current hidden state from historical characteristics of all currencies in the cross section, so that incorporates both the time-series dependencies and cross-country dependencies.

2.3.1 Model Architecture of TSAN

In the deep latent factor model, we employ an independent and identical TSAN for each currency to extract its own current hidden state from the historical characteristics. Additionally, the hidden state representations

[11] See Bahdanau et al. (2014) and Vaswani et al. (2017), among others.
[12] See Hochreiter and Schmidhuber (1997).
[13] See Chung et al. (2014).

Figure 4: Feed-forward Network (FFN)



in Beta Network and Factor Network are extracted independently, as we allow the hidden states in factor loadings and returns to capture different information from the historical characteristics. Whereas in the deep prediction model, the market hidden state is universal across currencies. As illustrated in Figure 2, the TSAN is shared by all currencies. It extracts an overall hidden state of the market from historical characteristics, which is then fed into an FFN to generate the predicted currency returns.

Without loss of generality, Figure 3 illustrates the architecture of our Time-Series Attention Network (TSAN). The input time series of historical characteristics $\{X^{(0)}, X^{(1)}, \ldots, X^{(N)}\}$ are first stacked and reshaped into a new sequence of vectors $X = \{x_{t-1}, \ldots, x_{t-T}\}$, in which x_{t-k} represents a $M \times 1$ vector of characteristics of all the $N+1$ countries in period $t-k$, where $k = 1, \ldots, T$ and $M = m^* + m * N$. The new sequence is then fed into the TSAN to extract an output $K_s \times 1$ vector of hidden state $s_{t-1}^{(i)}$, where K_s is a hyperparameter specifying the dimension of hidden state.

An attention network typically maps a query and a set of key-value pairs to an output, where the output is a weighted combination of the values with the weights, also called the attention score, computed from the query and the keys. To describe the TSAN in mathematical form:

$$q^{(i)} = W_Q^{(i)} x_{t-1} \tag{8}$$

$$K^{(i)} = W_K^{(i)} X \tag{9}$$

$$V^{(i)} = W_V^{(i)} X \tag{10}$$

$$s_{t-1}^{(i)} = V^{(i)}\ softmax({K^{(i)}}^\top q^{(i)}) \tag{11}$$

where $W_Q^{(i)}$, $W_K^{(i)}$ and $W_V^{(i)}$ are $K_s \times M$ matrices of parameters and are learned from the data in the training step; $q^{(i)}$ is a $K_s \times 1$ query vector, $K^{(i)}$ is a $K_s \times T$ key matrix, and $V^{(i)}$ is a $K_s \times T$ value matrix. As we focus on the current hidden state, our query is set to be a single vector computed by a dot product of the query parameter matrix $W_Q^{(i)}$ with the characteristics of period $t-1$, rather than characteristic of all historical periods. To facilitate illustration, the superscript i, where $i = 1, \ldots, N$, is specified in the architecture to denote currency in the deep latent factor model; however, in the deep prediction model, the TSAN is shared by all currencies and the superscript can thus be ignored.

2.4 Feed-forward Network (FFN)

This section describes the architecture of our feed-froward network (FFN), which is an essential component of both the deep latent factor model and deep prediction model discussed above in this section.

Feed-forward networks are the quintessential deep learning models and form the basis of many important machine learning applications.[14] A feed-forward network typically consists of an input layer, an output layer and one or more hidden layers in between, which are vector valued with different dimensions. It defines a deterministic mapping from the input to the output and learns the parameters that have the best approximation. Feed-forward networks are said to be universal approximators (Hornik et al. (1989)) as their approximation properties have been widely studied and found to be very general.[15]

Similar to the employment of TSAN, in the deep latent factor model, we employ an independent and identical FFN for each currency in Beta Network and Factor Network, respectively, to map from the hidden state to the factor loadings and factor portfolio weights. Whereas in the deep prediction model, as illustrated in Figure 2, the FFN is shared by all currencies and map the overall market hidden state to the predicted currency returns.

Throughout the models, we employ the same hyperparameters for architecture of all FFNs. Without loss of generality, Figure 4 illustrates the architecture of FFN in Beta Network of the deep latent factor model. For each currency i, an independent FFN is employed to map its current hidden state $s_{t-1}^{(i)}$ to its factor loadings $\beta_{t-1}^{(i)}$.

To describe the FFN in mathematical form, denote L as the number of hidden layers and $K^{[l]}$ as the number of neurons in each hidden layer $l = 1, \ldots, L$. In the first hidden layer,

$$z_{t-1}^{i,[l]} = g(W^{i,[0]} s_{t-1}^{(i)} + b^{i,[0]}),\ l = 1 \tag{12}$$

where $W^{i,[0]}$ is a $K^{[l]} \times K^s$ matrix of weight parameters, and $b^{i,[0]}$ is a $K^{[l]} \times 1$ vector of bias parameters. Both the weight and bias parameters are learned from data in the training step. We employ the rectified linear unit (ReLU), $g(y) = max(y, 0)$, as our nonlinear activation function $g(\cdot)$ throughout the models.

When $L > 1$, in the upper hidden layers, we have

$$z_{t-1}^{i,[l]} = g(W^{i,[l-1]} z_{t-1}^{i,[l-1]} + b^{i,[l-1]}),\ l = 2, \ldots, L \tag{13}$$

where $W^{i,[l-1]}$ is a $K^{[l]} \times K^{[l-1]}$ weight matrix and $b^{i,[l-1]}$ is a $K^{[l]} \times 1$ vector of bias.

The output vector of factor loadings for asset i is then generated by

$$\beta_{t-1}^{(i)} = W^{i,[L]} z_{t-1}^{i,[L]} + b^{i,[L]}, \tag{14}$$

where $W^{i,[L]}$ is a $K^{[l]} \times K$ matrix of weights and $b^{i,[L]}$ is a $K \times 1$ vector of bias.

The FFNs in Factor Network work in a similar way to map from the hidden state $\tilde{s}_{t-1}^{(i)}$ of each currency i to the vector of its portfolio weights $w_{t-1}^{(i)}$, thus are not discussed further. In the deep prediction model, the FFN is shared by all currencies and can thus ignore the superscript i in the illustration.

2.5 Regularization and Loss Function

Regularization, as one of the central concerns in machine learning, aims to reduce the generalization error of an algorithm (i.e. make the algorithm performs well not only on training data but also on new data) possibly at the expense of increased training error. Many different regularization strategies have been developed, and there is no best form of solution. Thus, the choice of regularization in a particular task expresses the preference for different solutions which is assumed to best suit the task. In our model, we choose the LASSO regularization and apply the penalty specifically to the input characteristics layers.

2.5.1 LASSO Regularization

LASSO, also known as the l_1 norm regularization, is a widely used regularization technique to reduce overfitting in the field of machine learning (Tibshirani (1996)). A LASSO regularization appends a l_1

[14] See, for example, Bishop (2006) and Goodfellow et al. (2016).
[15] See Hornik (1991) and Kreinovich (1991), among others.

norm penalty, the sum of the absolute value of the weight parameters, to the objective function of the machine learning model. Minimizing an objective function with LASSO regularization tends to produce sparse solutions. It encourages some weight parameters to continuously shrink to zero thus performs both parameter shrinkage and variable selection, hence can generate more stable and interpretable models. A similar idea is the ridge regression which applies a l_2 norm penalty, the sum of the squares of the weight parameters, to the objective function. However, the ridge regression only shrinks the size of the weight parameters without setting any of them to zero, thus it does not perform covariate selection and therefore does not help improve the interpretability of the model.

LASSO has been employed in the previous literature for predictors identification and characteristics selection and has resulted in improved out-of-sample performance.[16] Unlike the conventional way of regularizing neural network models, in which all weight parameters are penalized, we apply LASSO regularization just to the weight parameters of the input characteristic layers. In our model architecture design, we condition the factor loadings and factor returns on the characteristics of all the countries in the cross section, not just the relevant two countries of the bilateral exchange rate. As LASSO regularization diminishes the weight parameters for less important characteristics, by observing the characteristics importance in the trained regularized model, we can investigate which characteristics provide more information for explaining the exchange rate movements and study the interactive effects between characteristics of different countries.

2.5.2 Loss Function and Optimizer

The standard objective function summarizes the squared error between real returns and estimated returns over all the N currencies and T time periods. Let $J(\theta;\cdot)$ denotes the standard objective function:

$$J(\theta;\cdot) = \frac{1}{NT}\sum_{i=1}^{N}\sum_{t=1}^{T}(r_t^i - \hat{r}_t^i)^2, \tag{15}$$

where θ is a proxy for all the parameters in the model, r_t^i and $\hat{r}_t^i$ are the real return and estimated return of currency i in period t. Specifically, in the deep latent factor, $\hat{r}$ denotes the reconstructed return, whereas in the deep prediction model, it represents the predicted return.

The LASSO regularization adds a parameter norm penalty $\Omega(\theta)$ to the standard objective function $J(\theta;\cdot)$. Let $\tilde{J}(\theta;\cdot)$ denote the regularized loss function to be minimized, then

$$\tilde{J}(\theta;\cdot) = J(\theta;\cdot) + \lambda\Omega(\theta), \tag{16}$$

where $\lambda \in [0, +\infty)$ is a hyperparameter that weights the contribution of the LASSO penalty term $\Omega(\theta)$, larger values of λ correspond to stronger regularization.

We define the regularization term $\Omega(\theta)$ differently for our latent factor model and prediction model. In the deep latent factor model, we set the regularization term $\Omega(\theta)$ as

$$\Omega(\theta) = ||W_{Beta}^{[0]}||_1 + ||W_{Factor}^{[0]}||_1, \tag{17}$$

where $W_{Beta}^{[0]}$ and $W_{Factor}^{[0]}$ indicate the weight parameters for the input characteristic layer of the Beta Network and Factor Network, respectively. Whereas in the deep prediction model, the regularization term $\Omega(\theta)$ is defined as

$$\Omega(\theta) = \lambda||W^{[0]}||_1, \tag{18}$$

where $W^{[0]}$ indicates the weight parameters of the input characteristic layer.

In machine learning, three datasets are commonly used in different stages of building the model: a training set, a validation set and a test set. A training set is used to initially fit the model and estimate the parameters. Then a validation set is used to tune the model's hyperparameters. It provides an unbiased evaluation of the fitted model for a comparison among models with different set of hyperparameters. A test set is a heldout dataset that has never been used in training, it is used to provide a final evaluation of the

[16] See, for example, Chinco et al. (2019) and Freyberger et al. (2020).

fitted model. Following this template, we divide our data into three separate sets: training, validation and test sets. The hyperparameter λ in the LASSO penalty is then tuned in validation set.

For the optimization of the loss function, we employ the adaptive moment estimation algorithm (Adam) as our optimizer, which is one of the most widely used optimizers in machine learning (Kingma and Ba (2014)). The stochastic gradient descent method (SGD) is adopted for optimizing the objective function to reduce the computational burden and achieve faster convergence.

3 An Empirical Study of G10 Currencies

3.1 Data

We work with a sample of G10 currencies, with monthly returns based on spot rates from January 2000 through December 2020. Exchange rates are end of month values of US dollar vis-à-vis the remaining nine G10 currencies [17], defined as units of foreign currencies per unit of US dollar. The monthly currency return is computed as the monthly change of log exchange rate.

We build a collection of 34 common and country-specific characteristics that are motivated from the international economics and currency pricing literature and can be classified into four categories: 14 macroeconomic fundamentals, six interest rate-related variables, eight technical indicators, and the remaining six commodity-related common characteristics.[18] We obtain the macroeconomic fundamentals from the IMF and Bloomberg databases, interest rate data from the central banks of the G10 countries, and technical and commodities data from the Reuters database, with some of the technical data further processed manually.

Most of the characteristics are updated monthly, with six being updated quarterly and one being updated annually. Given that most of the macroeconomic fundamentals are released ex post with a delay, we use lagged economic fundamentals with their first-released data to avoid using future information, as Faust et al. (2003) suggests that the exchange rate prediction models tend to perform better when using original released data than revised data. Thus, for the prediction of month t, we use the most recent data available at the end of month t-1; more precisely, we use the most recent monthly updated fundamentals as of month t-1, the most recent quarterly updated fundamentals as of month t-4, and the most recent annually updated fundamentals as of month t-6 (Kelly et al. (2019), Gu et al. (2021)). We normalize each of the country-specific characteristics in the cross section with zero mean and unit variance for each month. The missing values are filled with the cross-sectional mean following Gu et al. (2018).

Rather than using the difference in characteristics between the domestic and foreign countries as predictors, as some structural models such as the monetary model do, we feed those country-specific characteristics into the neural network separately without any ex-ante combination, as the input characteristic layer of our model architecture suggests. In other words, we make no ex-ante assumptions about the relative coefficients of the fundamentals and allow the model to learn the coefficients from the data. In our input characteristic layers where $X^{(0)}, X^{(1)}, \ldots, X^{(N)}$ is fed into the model, $X^{(0)}$ is a matrix of common characteristics including US and commodity-related characteristics, $X^{(1)}, \ldots, X^{(N)}$ are matrices of country-specific characteristics for the remaining nine G10 currencies respectively, i.e., we work on a model with $N = 9$. Hence, the factor loadings and factor returns for each currency is conditioned on a total of 279 characteristics.[19]

3.2 Hyperparameters

We initially divide the 21 years of sample data into three disjoint subsets: 10 years of training set (2000-2009), 3 years of validation set (2010-2012) and 8 years of test set (2013-2020). In our input characteristic layers, we set our look-back window to be 12 months. That is, we set $T = 12$ and use characteristics from month $m - 12$ to month $m - 1$ when we model the returns for month t.

[17] Euro (EUR), Pound sterling (GBP), Japanese yen (JPY), Australian dollar (AUD), New Zealand dollar (NZD), Canadian dollar (CAD), Swiss franc (CHF), Norwegian krone (NOK) and Swedish krona (SEK)

[18] See Table A1 in Appendix for a full list.

[19] See Table A2 and Table A3 in Appendix for a detailed list.

The non-negative hyperparameter λ, which controls the strength of shrinkage in the LASSO penalty, is tuned based on the validation performance. Limited by the relatively high computational cost of recursively searching for an optimal λ for each refit and each architecture, we decide the value of λ based on the validation performance in a one-time fit and in a model with three hidden layers and four latent factors, the most complicated architecture among our candidate models. Then we fix the value of λ and no longer tune it when we refit the model.

After determining the value of λ, we no longer need a validation set thus can merge it into the training set. For example, to generate an estimate for the year 2013, we fit the model using data from 2000 to 2012 as training set. The model is refitted recursively at the end of each year. Rather than using a sliding window refit, we make advantage of all available data and expand the training set by one year in each refit to include the most recent twelve months.

For hyperparamemters with regard to the number of latent factors and hidden layers, we follow the setting of Gu et al. (2021) and consider a range of model architectures with different complexities. We evaluate three different architectures: a model with one hidden layer containing 16 neurons, a model with two hidden layers containing 16 and 8 neurons respectively and a model with three hidden layers containing 16, 8 and 4 neurons respectively. The specification of 16 neurons in the first hidden layer is approximately the square root of the number of predictors (which is a widely accepted guideline in machine learning). For each of the three model architectures, we further consider four different specifications: a range from one to four latent factors. Thus we have in total twelve candidate models under evaluation. We set our TSAN to have four hidden states, namely $K_s = 4$, following the specification of Chen et al. (2020).

Additionally, we employ an ensemble approach to improve the stability of the predictions. That is, rather than predicting based on a single fit, we generate the model predictions by averaging estimates from multiple independently trained models. For latent factor models, we set the number of independent fits to ten, while for prediction models, we set it to twenty due to the highly volatile performance.

3.3 Performance Evaluation: Deep Latent Factor Model

3.3.1 Model Comparison Set

We compare our deep latent factor model with a range of latent factor models. The first model is the principle component analysis (PCA) which employs solely assets return data without regard for asset characteristics, assuming linear functional form and static factor loadings and factor portfolio weights.

Our second comparison model is the conditional autoencoder model (CA) proposed by Gu et al. (2021), which conditions the factor loadings on the asset's own one-period lagged characteristics, as described in (2), and employs a nonlinear neural network functional form via autoencoder. In CA, the factor loadings vary dynamically with asset characteristics while the factor portfolio weights remain static.

The third comparison model is the fully connected conditional autoencoder model (FCA). As illustrated in Figure 5, the FCA model improves CA by allowing the factor loadings to condition on the characteristics of all the assets in the cross section, rather than just the asset itself. The FCA can be described as:

$$r_t^i = \beta(z_{t-1})^\top f_t + u_t^i \tag{19}$$

where the factor loadings $\beta(z_{t-1})$ is nonlinearly conditioned on the one-period lagged characteristics of all assets z_{t-1}, rather than the its own characteristics z_{t-1}^i as described in (2). Thus, the model allows for interaction between the characteristics of multiple assets and allows the factor loadings of one asset to be impacted by the characteristics of other assets.

The fourth comparison model, denoted FCA-DynFac, extends the FCA model by allowing the factor portfolio weights to condition also on the one-period lagged characteristics of all assets. Figure 6 illustrates the architecture of FCA-DynFac. The model can be described as:

$$r_t^i = \beta(z_{t-1})^\top f(r_t, z_{t-1}) + u_t^i \tag{20}$$

where both the factor loadings and factor portfolio weights are neural network models of the one-period lagged characteristics of all assets in the cross section. Hence, the factor portfolio weights are no longer static and can fluctuate dynamically over time.

Figure 5: This figure shows the architecture of the fully connected conditional autoencoder model (FCA). FCA model improves conditional autoencoder model (CA) of Gu et al. (2021) by allowing the factor loadings to condition on the characteristics of all the assets in the cross section, rather than just the asset itself.

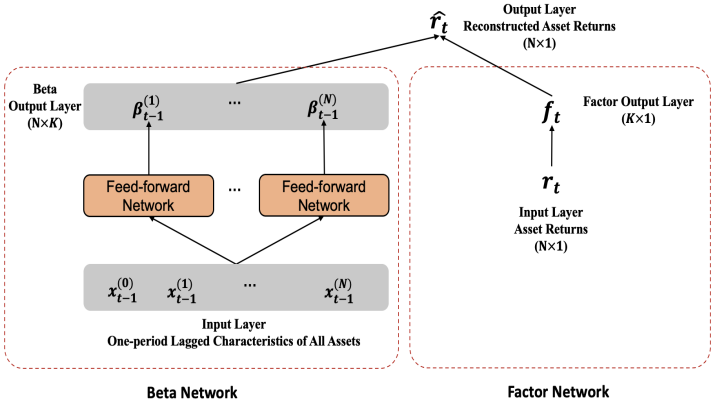


Figure 6: This figure shows the architecture of FCA-DynFac model. It extends the FCA model by allowing the factor portfolio weights to condition also on the one-period lagged characteristics of all assets.

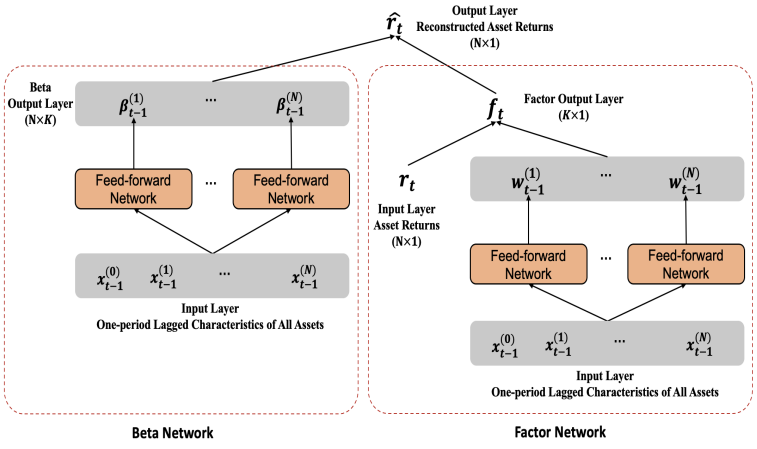
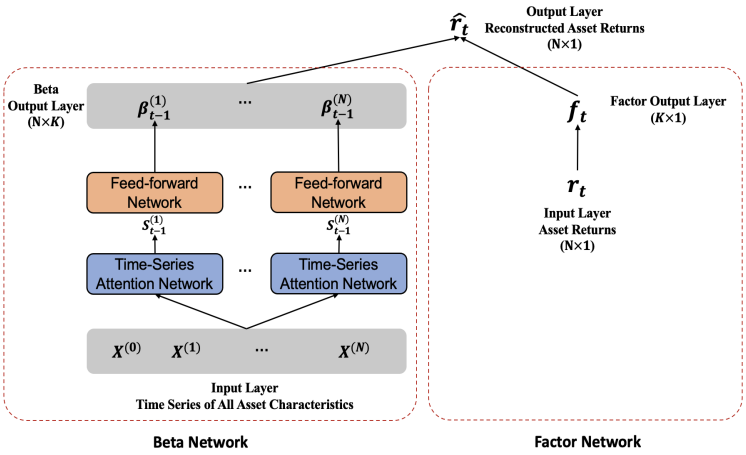


Figure 7: This figure shows the architecture of the FCA-TSAN model. FCA-TSAN extends the FCA model by including TSAN in its factor network, to extract market hidden states from the characteristics.



The fifth comparison model, which we note FCA-TSAN, extends the FCA model by including the time-series attention network (TSAN), as described in Section 2, in its factor network, to extract market hidden states from the characteristics. Figure 7 illustrates the architecture of the FCA-TSAN. In this model, the beta network employs an identical architecture as illustrated in Figure 1 while the factor portfolio weights remain static. We can describe FCA-TSAN in mathematical form as:

$$r_t^i = \beta(z_{t-1}, ..., z_{t-T})^\top f_t + u_t^i \tag{21}$$

where the conditional factor exposure is a neural network nonlinear function of the lagged historical characteristics of all assets in the cross section.

Our deep latent factor model (DLF), as described in (3) and illustrated in Figure 1, is distinguished from FCA-TSAN in that it allows factor portfolio weights to condition on the historical characteristics of all assets, with the time-series attention network (TSAN) extracting market hidden states from the characteristics. Thus, the factor portfolio weights can vary dynamically over time with the asset characteristics.

Table 1 summarize crucial components of five comparison models. Those components are associated with model improvements in the interaction effect over all countries, dynamic factor portfolio weights and historical state extraction via attention network.

Table 1: This table summarizes the primary architecture differences among five comparison deep factor models. ✓indicates the inclusion of components.

	CA	FCA	FCA-DynFac	FCA-TSAN	DLF
Interaction over countries	-	✓	✓	✓	✓
Dynamic Factor Portfolio Weight	-	-	✓	-	✓
Attention network on loading	-	-	-	✓	✓
Attention network on factor	-	-	-	-	✓

For each of the comparison model, we consider a range of model architectures corresponding to different levels of complexity. For PCA, we consider four different architectures: PCA with a number of latent factors ranging from one to four. For the remaining comparison models, we consider twelve different architectures: models with a number of hidden layers ranging from one to three, and each of these models with a number of latent factors ranging from one to four, respectively.

3.3.2 Statistical Performance Evaluation

First, we assess the model's overall contemporaneous explanatory power in explaining exchange rate movements. We use the out-of-sample cross-sectional contemporaneous R^2 (R^2_{Cont}) and root mean squared error (RMSE_{Cont}) as our metrics, which summarize the model's performance in all currencies all over the test set periods and are defined as:

$$R^2_{Cont} = 1 - \frac{\sum_{i,t\in Test}(r^i_t - \hat{r}^i_t)^2}{\sum_{i,t\in Test} {r^i_t}^2}, \tag{22}$$

$$RMSE_{Cont} = \sqrt{\frac{1}{NT}\sum_{i,t\in Test}(\hat{r}^i_t - r^i_t)^2}. \tag{23}$$

The out-of-sample R^2_{Cont} measures the contemporaneous descriptive ability of the factor portfolios and is commonly used in linear asset pricing. The root mean squared error is a widely employed metric in exchange rate prediction, as in Meese and Rogoff (1983a,b), hence we use the out-of-sample contemporaneous root mean squared error (RMSE_{Cont}) to help evaluate the pricing accuracy of the model.

Table 2 and 3 report the out-of-sample R^2_{Cont} and RMSE_{Cont} of DLF and the five comparison models with different architectures. As the number of latent factors increases, the performance generally improves monotonically (with a greater R^2_{Cont} and a smaller RMSE_{Cont}). Whereas performance varies differently with respect to the number of hidden layers for different models. The performance of FCA, FCA-DynFac and DLF improves monotonically as the number of hidden layers grows, CA and FCA-TSAN, on the other hand, do not exhibit a significant change in performance as the number of hidden layers varies. In comparison to the number of hidden layers in architecture, the number of latent factors has a greater influence on model performance.

The highest out-of-sample R^2_{Cont} (lowest out-of-sample contemporaneous RMSE), 84.62% (1.03), is presented by FCA with three hidden layers and four latent factors, closely followed by several slightly weaker performances delivered by models with four latent factors.

Interestingly, the neural network models fail to outperform the simplest linear model, PCA, in terms of the contemporaneous explanatory power. The PCA loadings remain remarkably stable over time when performed on an expanding window. This underperformance of neural network models, particularly those with simpler architectures, is likely a result of overfitting due to the limited data set in the training step.

Next, we evaluate the model's predictive accuracy by the out-of-sample predictive root mean squared error (RMSE_{Pred}):

$$RMSE_{Pred} = \sqrt{\frac{1}{NT}\sum_{i,t\in Test}(\hat{r}^i_{t,Pred} - r^i_t)^2}, \tag{24}$$

where

$$\hat{r}^i_{t,Pred} = {\beta^i_{t-1}}^\top f_{t-1}. \tag{25}$$

The RMSE_{Pred} compares the out-of-sample predictive ability of the candidate models. Table 4 reports the out-of-sample RMSE_{Pred} of the models with different architectures. The neural network models outperform PCA consistently in terms of RMSE_{Pred}, indicating the advantage of employing neural networks to capture time variation through characteristics in prediction.

The best overall model in terms of out-of-sample predictive ability is DLF with one factor. The lowest RMSE_{Pred} of 2.70 is delivered by DLF with one latent factor and one hidden layer, followed by a slightly weaker performance of DLF with one latent factor and two hidden layers, which delivers a RMSE_{Pred} of 2.71.

In contrast to the contemporaneous explanatory performance, the predictive performance tends to decrease as the number of latent factors grows in each model. Additionally, increasing the number of hidden layers also exhibits a generally negative effect on the predictive performance. That is, the model tends to perform better with simple architectures.

In our model architecture design, FCA-DynFac differs from FCA in that it allows dynamic weights derived from characteristics in the factor portfolios. The superior performance of FCA-DynFac, except for

Table 2: R^2_{Cont} (%) results. In this table, we report the out-of-sample contemporaneous R^2 of the six latent factor models, i.e., PCA, FCA, FCA-DynFac, FCA-TSAN and DLF, with the number of latent factors ranging from 1 to 4 and the number of hidden layers ranging from 1 to 3 (except for PCA, which does not have multiple hidden layers).

Number of Hidden Layers	Model	Number of Latent Factors			
		1	2	3	4
	PCA	58.83	69.33	78.53	84.23
1	CA	42.85	64.35	74.03	82.11
	FCA	20.23	57.47	73.06	81.79
	FCA-DynFac	<0	<0	<0	<0
	FCA-TSAN	49.36	69.71	77.47	84.36
	DLF	34.68	65.33	76.86	83.54
2	CA	39.72	64.94	74.50	82.49
	FCA	28.80	63.59	75.18	82.74
	FCA-DynFac	12.89	16.61	14.56	8.58
	FCA-TSAN	52.98	66.78	77.76	83.53
	DLF	45.02	67.00	76.53	83.61
3	CA	45.03	66.95	76.73	82.56
	FCA	31.91	64.81	77.45	84.62
	FCA-DynFac	19.65	38.36	48.38	54.17
	FCA-TSAN	51.75	69.14	77.13	83.72
	DLF	46.95	68.88	76.41	82.43

those with a single hidden layer, demonstrates the advantages of employing conditional dynamic weights in factor portfolios. DLF further extends FCA-DynFac by allowing factor loadings and factor portfolio weights to condition on historical characteristics rather than one-period lagged characteristics via TSAN architectures. The additional outperformance of DLF over FCA-DynFac reveals the benefits of extracting market hidden states from historical characteristics in factor loadings and factor portfolio weights. Interestingly, FCA-TSAN consistently underperforms FCA. These two models differ in that FCA excludes the TSAN architectures in its Beta Network, whereas both two models employ a single linear hidden layer in the Factor Network and assume static portfolio weights. Thus, the consistent underperformance of FCA-TSAN suggests a negative effect of TSAN architectures in Beta Network when employing static factor portfolios.

3.3.3 Economic Performance Evaluation

To further assess the model performance in terms of the economic value, we construct long short portfolios based on the out-of-sample predictive returns and compare the Sharpe ratios of each model.

The currency excess return of a U.S. investor holding foreign currency k is defined as:

$$rx^k_{t+1} = i^k_t - i_t - \Delta s^k_{t+1}, \tag{26}$$

where i^k and i denotes the one-month interest rate in country k and the U.S., respectively, and Δs denotes the log spot rate change or return.

For each model, we construct self-financing long-short portfolios in two ways. The first portfolio longs the currency with the highest forecasted excess return and shorts the currency with the lowest, whereas the second portfolio buys the currencies with the top two highest predictive returns and sells the bottom two currencies, which are all equal-weighted. At the end of each month, we rebalance the portfolios by sorting the currencies based on their out-of-sample predictive excess returns described in (25).

Table 5 reports the out-of-sample Sharpe ratios of the long-one-short-one portfolios for each model with a range of architectures. Comparing the overall performance of the six candidate models, the results are generally consistent with the statistically performance as measured by RMSE_{Pred}. In terms of economic value, the five neural network models consistently outperform PCA and tend to perform better with simpler architectures. DLF delivers the best overall performance, with the highest Sharpe ratio of 0.77 achieved by

Table 3: RMSE_{Cont} (%) results. In this table, we report the out-of-sample contemporaneous RMSE of the six latent factor models, i.e., PCA, FCA, FCA-DynFac, FCA-TSAN and DLF, with the number of latent factors ranging from 1 to 4 and the number of hidden layers ranging from 1 to 3 (except for PCA, which does not have multiple hidden layers).

Number of Hidden Layers	Model	Number of Latent Factors			
		1	2	3	4
	PCA	1.69	1.46	1.22	1.04
1	CA	1.99	1.57	1.34	1.11
	FCA	2.35	1.71	1.36	1.12
	FCA-DynFac	2.86	3.65	4.14	4.31
	FCA-TSAN	1.87	1.45	1.25	1.04
	DLF	2.48	2.36	2.23	2.22
2	CA	2.04	1.56	1.33	1.10
	FCA	2.22	1.59	1.31	1.09
	FCA-DynFac	2.45	2.40	2.43	2.51
	FCA-TSAN	1.80	1.52	1.24	1.07
	DLF	2.33	1.97	1.87	1.84
3	CA	1.95	1.51	1.27	1.10
	FCA	2.17	1.56	1.25	1.03
	FCA-DynFac	2.36	2.06	1.89	1.78
	FCA-TSAN	1.83	1.46	1.26	1.06
	DLF	2.23	1.88	1.76	1.63

Table 4: RMSE_{Pred} (%) results. In this table, we report the out-of-sample predictive RMSE of the six latent factor models, i.e., PCA, FCA, FCA-DynFac, FCA-TSAN and DLF, with the number of latent factors ranging from 1 to 4 and the number of hidden layers ranging from 1 to 3 (except for PCA, which does not have multiple hidden layers).

Number of Hidden Layers	Model	Number of Latent Factors			
		1	2	3	4
	PCA	3.30	3.42	3.52	3.57
1	CA	3.17	3.27	3.32	3.38
	FCA	2.91	3.08	3.27	3.35
	FCA-DynFac	3.20	3.69	4.12	4.51
	FCA-TSAN	3.15	3.30	3.36	3.44
	DLF	2.70	2.85	2.84	2.93
2	CA	2.93	3.17	3.28	3.34
	FCA	3.00	3.21	3.33	3.44
	FCA-DynFac	2.88	2.93	3.01	3.17
	FCA-TSAN	3.26	3.30	3.41	3.47
	DLF	2.71	2.86	3.01	2.96
3	CA	3.05	3.24	3.30	3.36
	FCA	3.10	3.24	3.39	3.46
	FCA-DynFac	2.74	3.00	3.18	3.25
	FCA-TSAN	3.30	3.33	3.38	3.48
	DLF	2.79	3.02	3.02	3.10

Table 5: Sharpe Ratios of Long-One-Short-One Portfolios

Number of Hidden Layers	Model	Number of Latent Factors			
		1	2	3	4
	PCA	-0.13	0.02	-0.20	0.02
1	CA	0.17	0.14	-0.19	0.10
	FCA	0.38	0.38	-0.14	-0.11
	FCA-DynFac	-0.12	0.08	0.40	0.02
	FCA-TSAN	0.23	0.22	0.15	-0.03
	DLF	0.49	0.32	-0.03	0.77
2	CA	-0.04	0.23	-0.15	-0.05
	FCA	0.09	-0.09	-0.11	0.01
	FCA-DynFac	0.28	-0.07	-0.27	0.31
	FCA-TSAN	-0.05	0.42	0.37	0.15
	DLF	0.31	0.37	0.39	0.39
3	CA	0.08	-0.17	-0.02	-0.08
	FCA	-0.02	-0.22	0.07	0.06
	FCA-DynFac	0.43	0.13	-0.17	0.13
	FCA-TSAN	0.17	0.15	0.14	0.10
	DLF	-0.08	0.22	0.05	0.14

Note: In this table, we report the out-of-sample Sharpe ratios of long-short portfolios for the six latent factor models, i.e., PCA, FCA, FCA-DynFac, FCA-TSAN and DLF, with the number of latent factors ranging from 1 to 4 and the number of hidden layers ranging from 1 to 3 (except for PCA, which does not have multiple hidden layers). In each portfolio, we long the currency with the highest predictive excess return and short the currency with the lowest.

DLF with one hidden layer and four latent factors.

As model complexity varies, the portfolio Sharpe ratios are not as monotonic as the RMSE_{Pred}. The performances exhibit more volatility, as betting on a single currency could result in a single wrong bet having a substantial effect on the entire Sharpe ratio. Hence, we report the Sharpe ratios of the long-two-short-two portfolios as an additional reference. As shown in Table 6, all neural network models deliver positive Sharpe ratios regardless of the architecture. The highest Sharpe ratio, 0.56, is again achieved by DLF with one hidden layer and four latent factors.

3.3.4 Pricing Errors

Our model setting, similar to Gu et al. (2021), imposes a no-arbitrage restriction by allowing no intercept in the factor model. In contrast to the prediction model which contains no factor structure, the factor model encourages the return predictability to come only through factor risk exposures. Thus, in this section we test the out-of-sample pricing errors of our DLF model. The pricing error for currency i is defined as:

$$\alpha^i = E[r_t^i] - E[\hat{r}_t^i]. \tag{27}$$

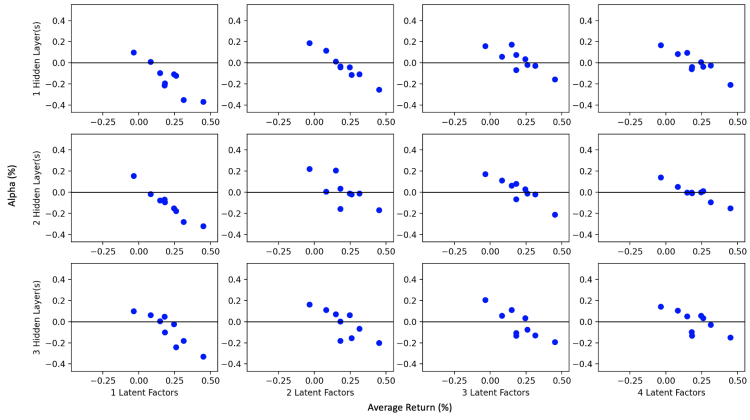
We test whether the pricing errors are statistically indistinguishable from zero by observing the t-statistics of the alphas for the nine currencies. We conduct the test for each of our twelve architectures: DLF with number of hidden layers ranging from one to three and number of latent factors ranging from one to four. Among all the architectures, none of the t-statistics of alphas for the nine currencies exceeds 3.0, indicating that the no-arbitrage restriction is satisfied in the data.

Figure 8 reports the alphas against average realized returns of the nine currencies for the twelve different architectures: DLF with the number of hidden layers ranging from one to three and the number of latent factors ranging from one to four. Each subplot represents one of the twelve architectures, and each scatter represents one of the nine currencies. As shown in the figure, the scatters become less spread out as the number of latent factors increases. This corresponds to the DLF contemporaneous explanatory performance as illustrated in Table 3, where RMSE_{Cont} decreases as the complexity of the DLF architecture grows, indicating that the pricing errors shrink as the complexity of the DLF architecture increases.

Table 6: Results of Sharpe Ratios of Long-Two-Short-Two Portfolios. In this table, we report the out-of-sample Sharpe ratios of long-short portfolios for the six latent factor models, i.e., PCA, FCA, FCA-DynFac, FCA-TSAN and DLF, with the number of latent factors ranging from 1 to 4 and the number of hidden layers ranging from 1 to 3 (except for PCA, which does not have multiple hidden layers). In each portfolio, we long the currencies with the top two highest predictive returns and short the bottom two currencies, which are all equal-weighted.

Number of Hidden Layers	Model	Number of Latent Factors			
		1	2	3	4
	PCA	0.15	0.25	-0.07	-0.01
1	CA	0.27	0.06	0.43	0.24
	FCA	0.30	0.42	0.24	0.12
	FCA-DynFac	0.12	0.35	0.23	0.26
	FCA-TSAN	0.24	0.38	0.27	0.13
	DLF	0.15	0.14	0.35	0.56
2	CA	0.15	0.28	0.16	0.40
	FCA	0.13	0.25	0.41	0.13
	FCA-DynFac	0.10	0.09	0.11	0.04
	FCA-TSAN	0.15	0.37	0.31	0.09
	DLF	0.19	0.17	0.39	0.30
3	CA	0.00	0.20	0.32	0.05
	FCA	0.27	0.13	0.29	0.29
	FCA-DynFac	0.34	0.11	0.22	0.04
	FCA-TSAN	0.41	0.27	0.33	0.09
	DLF	0.15	0.02	0.05	0.17

Figure 8: Pricing Errors results. The figure reports the alphas (%) against average realized returns (%) of the nine currencies for twelve different DLF model architectures: DLF with the number of hidden layers ranging from one to three and the number of latent factors ranging from one to four.



3.3.5 Interactive Effects and Characteristic Importance

We assess the characteristic importance and cross-country interactive effects based on the characteristic sensitivity, which measures the sensitivity of the output returns with respect to the characteristic. A higher sensitivity indicates that the characteristic has a greater influence on the output currency returns. Similar to Chen et al. (2020), we define the characteristic sensitivity based on the average absolute gradient. That is, for currency i, the sensitivity of a particular characteristic, x_m, is defined as the average absolute gradient of the output returns with respect to this characteristic. Mathematically,

$$S_i(x_m) = \frac{1}{T} \sum_{t \in Test} |\frac{\partial \hat{r}_t^i}{\partial x_m}|, \tag{28}$$

where $S_i(x_m)$ denotes the sensitivity of currency i with respect to characteristic x_m, T is the number of periods in the test set.

Figure 9 illustrates the sensitivity of output returns to all characteristics for each currency. To facilitate comparison of sensitivity to a particular characteristic for different currencies, we convert the characteristic sensitivity to its proportion of all characteristics in that currency. To describe mathematically,

$$\tilde{S}_i(x_m) = \frac{S_i(x_m)}{\sum_m S_i(x_m)}. \tag{29}$$

The results are based on a DLF model with three hidden layer and four latent factors, the most complex architecture among those provide the highest contemporaneous R^2. Other model architectures provide virtually the same results thus are not reported here.

Figure 9(a) reports the sensitivity of output currency returns with respect to the country-specific characteristics, with each sub-heatmap indicating the sensitivity of returns for all nine currencies to the characteristics of a particular country. The sensitivity to common characteristics, namely the US and commodity-related characteristics, is illustrated Figure 9(b).

Notably, the characteristic sensitivities exhibit an obvious cross-country interactive pattern, as illustrated by the vertical pattern in the heatmap colors. The characteristics that have a greater impact on the output return of one currency also tend to have a greater impact on other currencies. These cross-country interactive effects support our assumption that a currency's performance is affected not only by the characteristics of the relevant two countries in the bilateral exchange rate, but also by the characteristics of other countries in the cross section.

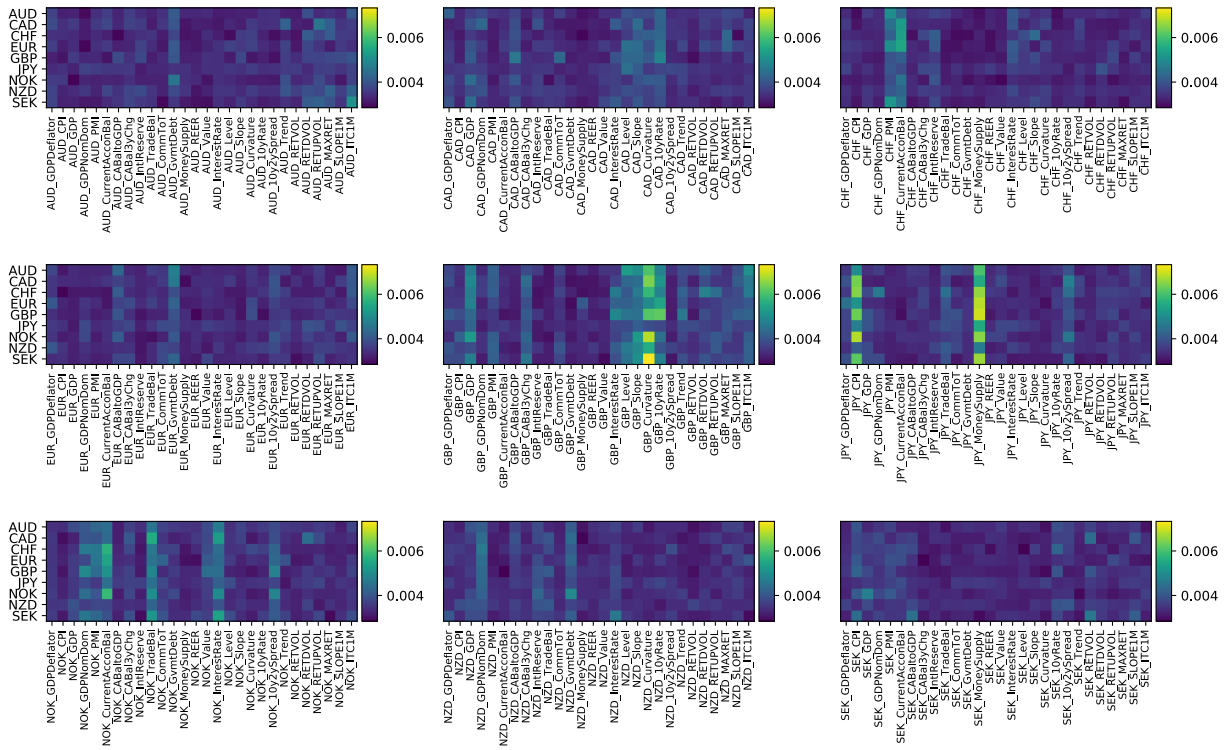
We further look at the characteristic importance based on the overall sensitivity, define as the characteristic's average sensitivity across all currencies:

$$S(x_m) = \frac{1}{N} \sum_i S_i(x_m), \tag{30}$$

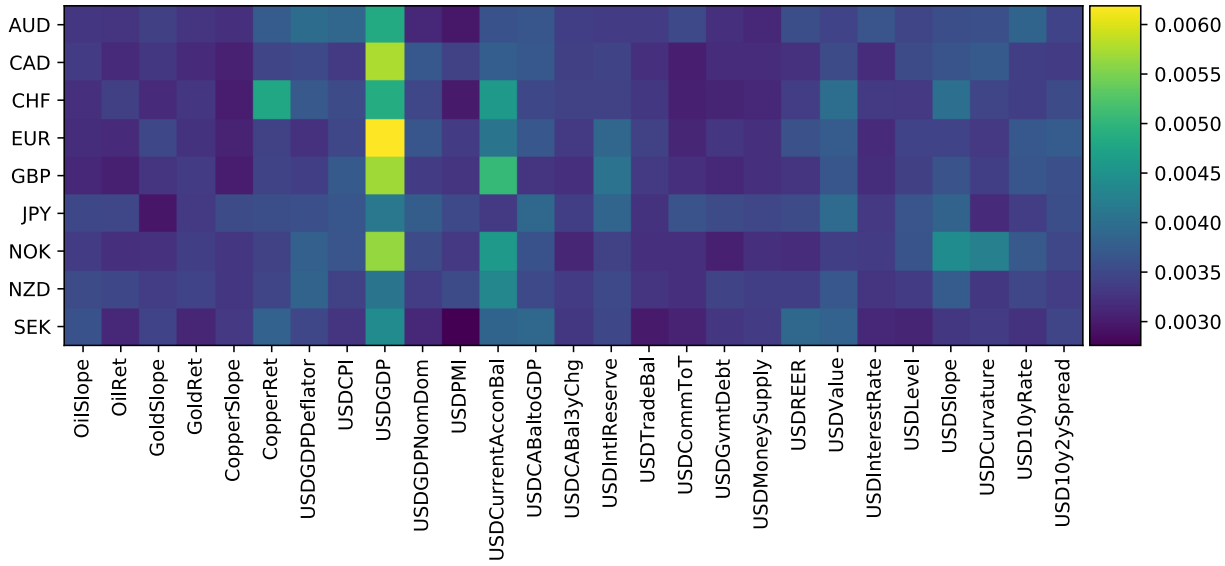
We rank our total of 279 characteristics based on their characteristic importance. Figure 10 depicts the top 50 characteristics. There is little evidence of top-tier dominance shown in the characteristic importance, as the top 50 contribute just around 21% of total characteristic importance. Thus, to better observe the characteristic importance, we categorize the characteristics into four groups – macroeconomic, interest rate-related, technical, and commodity-related, as described in Table A1. Characteristics in the first three groups can be further classified by country. Figure 11 illustrates the average characteristic importance of each group. The UK interest rate group has the highest average importance and well outweighs the second group. The top groups are dominated by interest rate-related and technical characteristics, indicating that monthly exchange rate fluctuations might be driven more by higher frequency rate data rather than the relatively low frequency macroeconomic data.

To further investigate the dynamics of dominant characteristics, we look into the characteristic importance for each year during the out-of-sample periods. Figure 12 illustrate how the rankings of dominant characteristic groups evolve over time. Although the average characteristic importance of various groups is not substantially different, the ranking is relatively consistent, as shown in the figure. The dominant characteristics, for example the UK interest rate group, continue to outweigh other characteristic groups in terms of the overall sensitivity, and the top groups remain dominated by higher frequency rate-related characteristics.

Figure 9: Cross-Country Interactive Effects Results. The figure reports the sensitivity of output returns to all characteristics for each currency. Figure (a) reports the sensitivity to country-specific characteristics. Each sub-heatmap illustrates the sensitivity of returns for all nine currencies to the characteristics of a particular country. Figure (b) reports the sensitivity to common characteristics.



(a) Sensitivities to Country-Specific Characteristics



(b) Sensitivities to Common Characteristics

Figure 10: Characteristic Importance. The figure reports the top 50 characteristics among the total 279 characteristics ranked by their characteristic importance. Different bar colors depict characteristics of different countries.

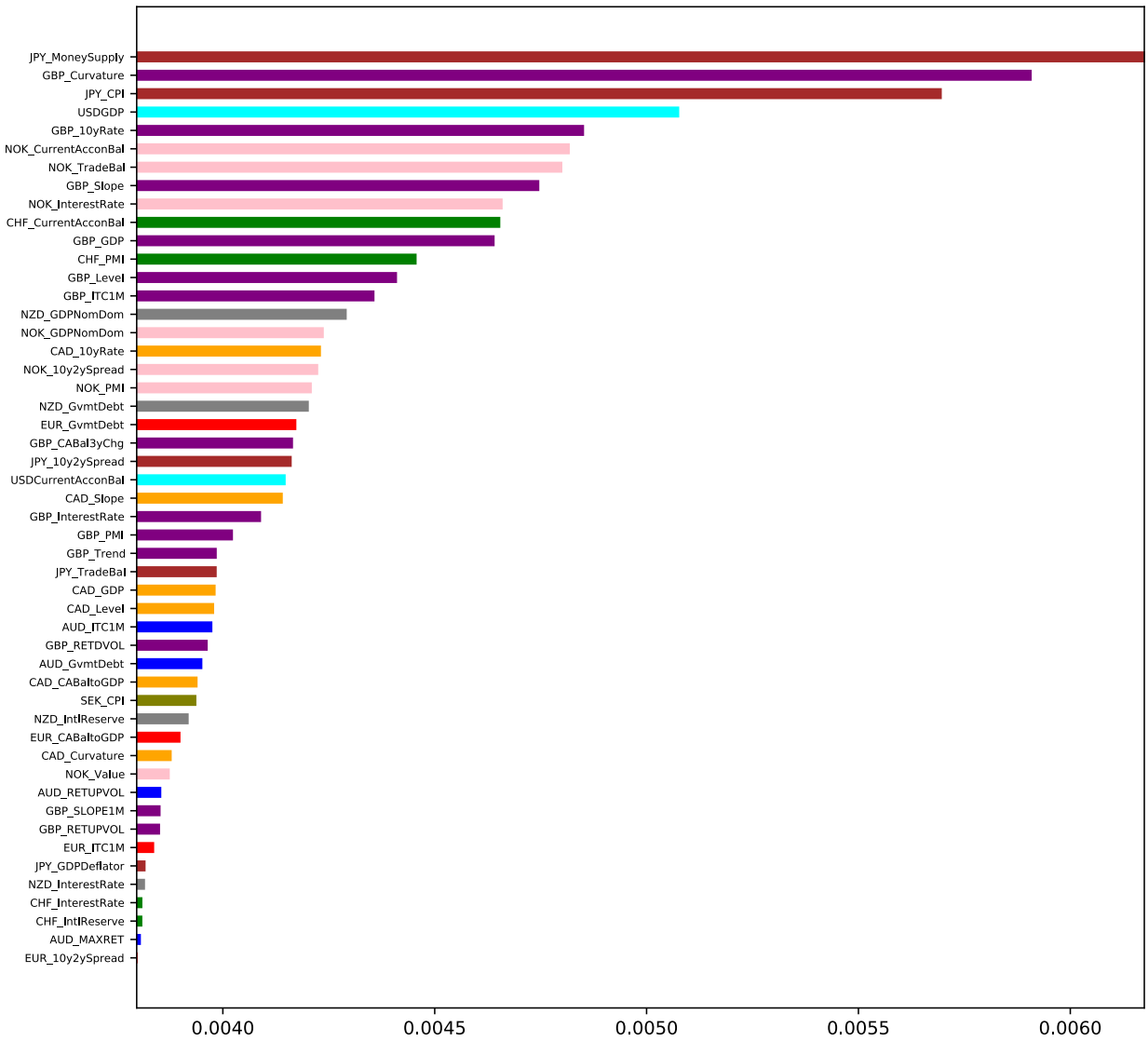


Figure 11: Characteristic Importance by Group. The figure reports the average characteristic importance of each group. The characteristics are classified into four categories: macroeconomic, interest rate-related, technical, and commodity-related, with the first three further grouped by country.

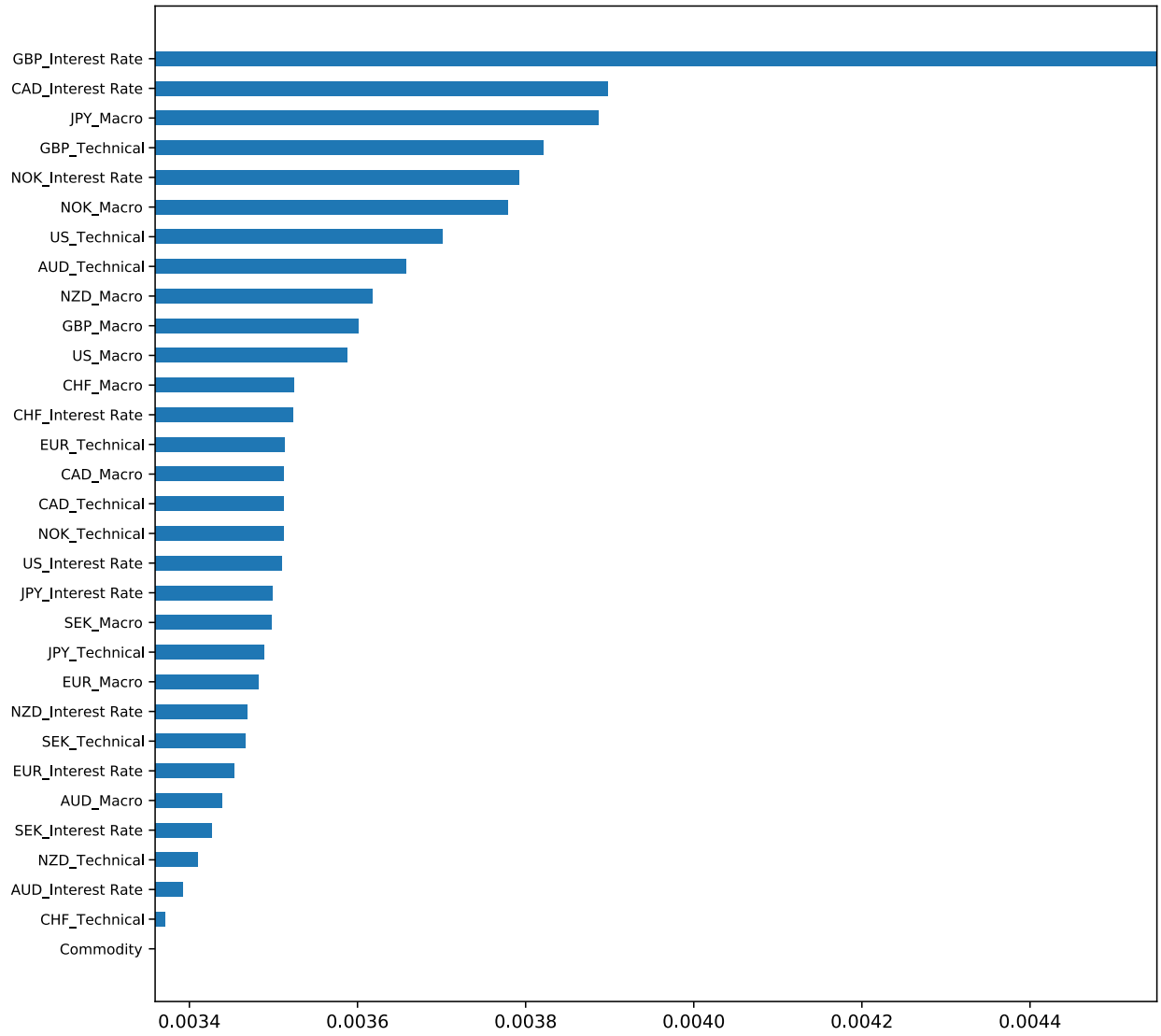


Figure 12: Characteristic Importance over Time. The figure reports the average characteristic importance of the groups for each year during the out-of-sample periods. It depicts how the rankings of dominant characteristic groups evolve over time. A larger point in the figure indicates a greater importance of the characteristic.

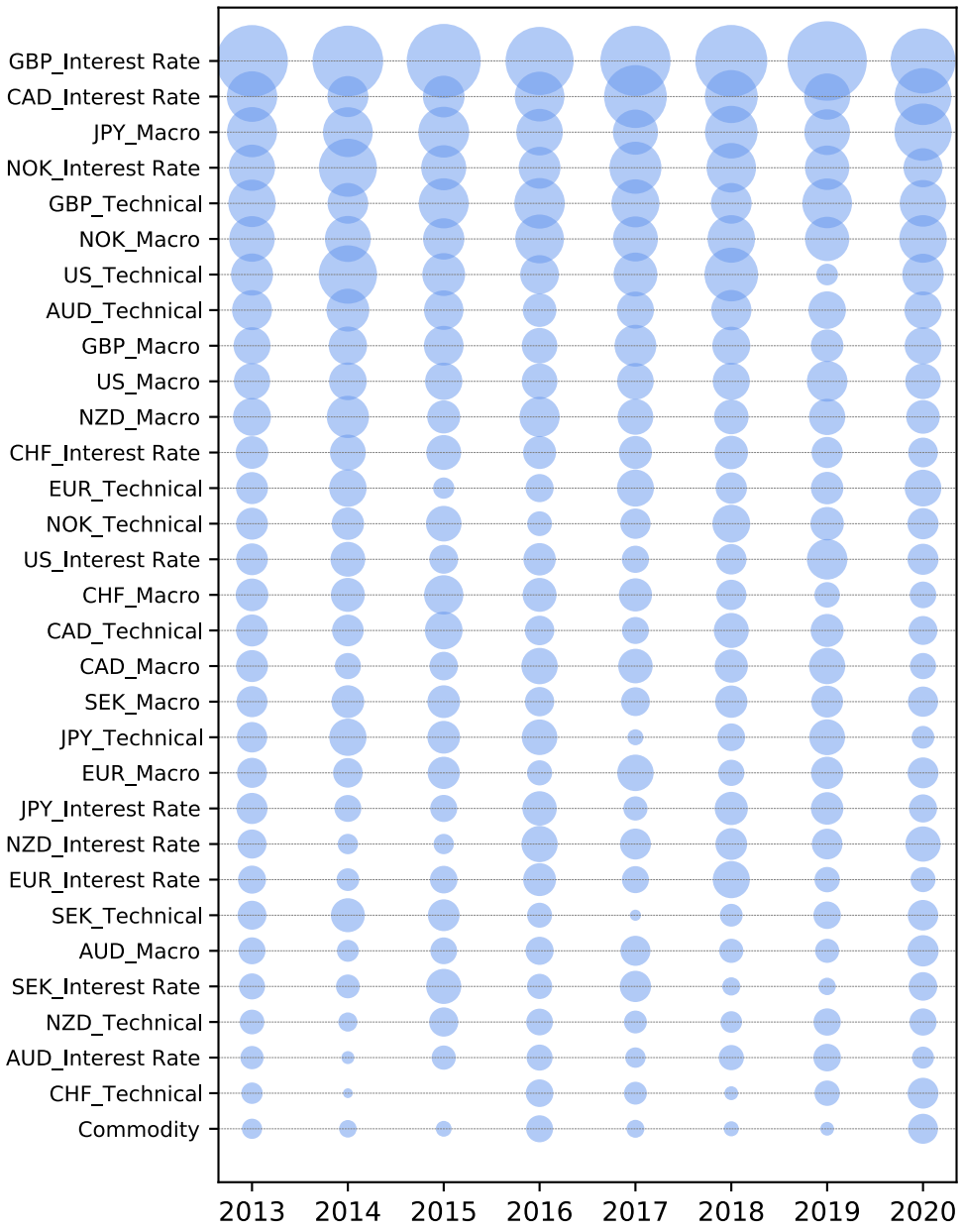
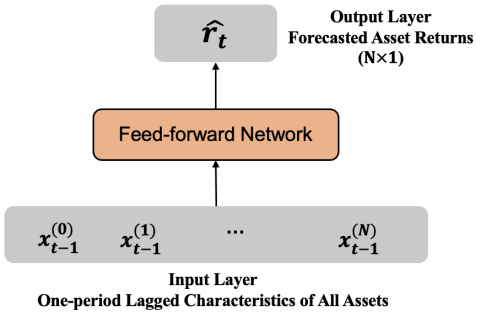


Figure 13: PRED-FC



3.4 Performance Evaluation: Deep Prediction Model

Our deep prediction model, which we note PRED-TSAN-FC, are designed to provide exchange rate return prediction in time series. As introduced in Section 2.2 and illustrated in Figure 2, the prediction model maps the historical characteristics of all countries to the forecasted returns for the next term. To evaluate the forecasting performance, we compare PRED-TSAN-FC to several other models in our model comparison set against the random walk benchmark, a widely studied robust benchmark in the literature of exchange rate forecasting.

Since Meese and Rogoff (1983a,b), the random walk has been shown to be the best predictor and used extensively across papers in the literature. In this paper, we employ a random walk forecast without drift ("RW") as our predictor benchmark, which is well-known in the literature to be a harder model to beat than its alternative, the random walk with drift (Rossi (2013)). It can be defined as:

$$E_{t-1}(r_t^i) = 0. \tag{31}$$

3.4.1 Model Comparison Set

Our first comparison model, noted as PRED-FC, are designed as illustrated in Figure 13. The model architecture is distinct from the PRED-TSAN-FC design in that it does not contain the TSAN architecture (the network that extracts hidden states from long historical characteristics). Thus, instead of time series of characteristics, the model input is the one-period lagged characteristics of all assets. The input characteristics vector passes through an FFN, identical to the architecture described in Section 2, to form an $N \times 1$ vector of forecasted individual currency returns $\hat{r}_t$.

Our second comparison model, denoted PRED-TSAN, is distinct from PRED-TSAN-FC by employing a distinct subnetwork for each currency. Each subnetwork is comprised of a TSAN and an FFN architecture. All the TSANs and FFNs have the same architecture as detailed in Section 2 and share the same hyperparameters. As illustrated in Figure 14, the architecture of PRED-TSAN is identical to that of Beta Network in DLF except for the output layer: in PRED-TSAN, for each currency i, the input characteristic time series pass through a TSAN and an FFN to form a scalar representing the predicted return on that currency, $\hat{r}_t^{(i)}$.

Our third comparison model, denoted PRED-PC, is motivated by Filippou et al. (2020) and takes a "partially" connected architecture in the input characteristics layer. Unlike PRED-FC where the model maps the lagged characteristics of all assets to an $N \times 1$ vector of forecasted currency returns, PRED-PC maps the lagged characteristics of one asset to a scalar of forecasted return for that asset. As illustrated in Figure 15, for each currency i, the input one-period lagged characteristics, i.e., a combination of common characteristics $x_{t-1}^{(0)}$ and country-specific characteristics $x_{t-1}^{(i)}$, pass through an FFN to form a scalar of predicted return on that currency, $\hat{r}_t^{(i)}$. When training the model, we pool the data so that the learnt parameters are the same across currencies. In other words, all currencies share the same prediction model.

Our fourth comparison model, noted as PRED-TSAN-PC, extends PRED-PC by including a time-series attention network (TSAN), as describe in section 2, before FFN to extract market hidden states from the

Figure 14: PRED-TSAN

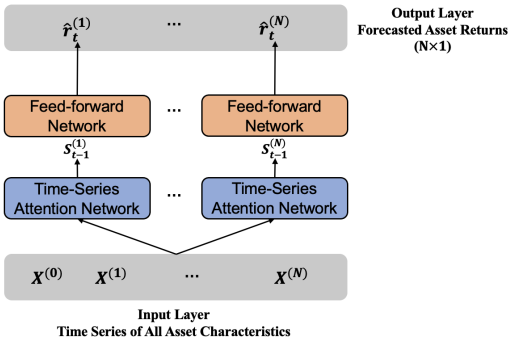
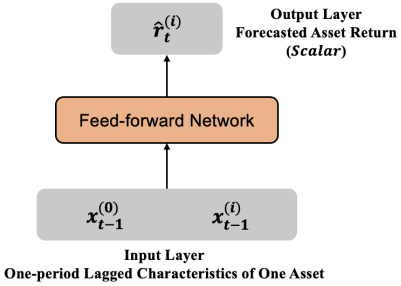


Figure 15: PRED-PC



historical characteristics. As illustrated in Figure 16, for each currency i, the input characteristics time series, comprised of common characteristics $X^{(0)}$ and country-specific characteristics $X^{(i)}$, pass through a TSAN to derive a hidden state vector $s_{t-1}^{(i)}$. The hidden state then pass through an FFN to form the output scalar of predicted return on that currency, $\hat{r}_t^{(i)}$. Similar to PRED-PC, we pool the data in the training step and share the model across all currencies.

We summarize the distinctive architectures of five comparison models in Table 7. The constituent architectures, which are associated with the full or partial interaction effects over countries and the historical state extraction via attention network, reflect different performances in empirical analysis.

Table 7: This table summarizes the primary architecture differences among five prediction models. ✓indicates the inclusion of components.

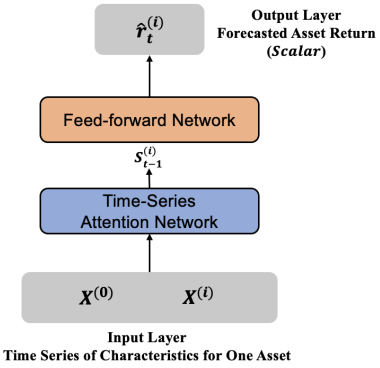
	PRED-FC	PRED-TSAN	PRED-PC	PRED-TSAN-PC	PRED-TSAN-FC
Full interaction of characteristics over countries	✓	✓	-	-	✓
Partial interaction of characteristics	-	-	✓	✓	-
Attention network to extract hidden states	-	✓	-	✓	✓
Distinct subnetwork for each currency	-	✓	-	-	-

3.4.2 Statistical Performance Evaluation

To compare the predictive performance of our prediction model, PRED-TSAN-FC, to the two models in the comparison set, we employ four different loss functions. The first is the root mean square forecast error (RMSFE), as employed in Meese and Rogoff (1983a,b) and Meese and Rogofp (1988), which is defined as:

$$RMSFE = \sqrt{\frac{1}{NT} \sum_{i,t \in Test} (\hat{r}_t^i - r_t^i)^2} \tag{32}$$

Figure 16: PRED-TSAN-PC



The second loss function is the mean absolute errors (MAE), as in Meese and Rogoff (1983b), which is defined as:

$$MAE = \sqrt{\frac{1}{NT} \sum_{i,t \in Test} |\hat{r}_t^i - r_t^i|} \tag{33}$$

The third loss function is the Theil's (1966) U statistic[20], which is a widely used measure in the literature[21] defined as the ratio of model's RMSFE to the RMSFE of the benchmark model. The final one is the R^2 statistic,[22] which measures the relative accuracy of the model prediction against the benchmark prediction and is defined as:

$$R^2 = 1 - \frac{\sum_{(i,t) \in Test} (\hat{r}_t^i - r_t^i)^2}{\sum_{(i,t) \in Test} {r_t^i}^2}, \tag{34}$$

A Theil U-statistic less than one indicates a superior model performance compared to the benchmark. The R^2 evaluates the proportional reduction in mean squared errors of the model prediction relative to the benchmark prediction. Thus, a positive R^2 implies an outperformance in model prediction over the benchmark, whilst a greater R^2 suggests a better performance.

Table 8 reports the out-of-sample performances of the five prediction models and the random walk benchmark. PRED-FC and PRED-PC fail to outperform the random walk benchmark systematically in terms of all the measure statistics and model complexities. Whereas PRED-TSAN, PRED-TSAN-PC and PRED-TSAN-FC consistently outperform PRED-FC and PRED-PC regardless of the model architecture or evaluation statistic used, with PRED-TSAN-FC dominating and delivering the best predictive performance in terms of each statistic and architecture. In our model architecture design, PRED-FC (PRED-PC) varies from PRED-TSAN-FC (PRED-TSAN-PC) only in that it excludes the TSAN architecture. When comparing the performance of those two, the significant underperformance of PRED-FC (PRED-PC) indicates the critical necessity of TSAN architecture in prediction model. In other words, identifying hidden market states from historical characteristics substantially improves the predictive accuracy.

In PRED-TSAN, we employ a distinct subnetwork for each currency. That is, we extract exclusive hidden market states which are then fed into an FFN to generate a forecasted return of this currency. Whereas in PRED-TSAN-FC all the currencies share the same hidden market states and FFN architecture. The primary distinction between these two models is the model complexity and number of trainable parameters: PRED-TSAN has times the number of trainable parameters as PRED-TSAN-FC with a more complicated architecture. The minor underperformance of PRED-TSAN is likely a result of the limited data set in the training step and the low signal-to-noise ratio in exchange rate data compared to those industrial

[20] See Theil (1966).
[21] See Rossi (2013).
[22] See Campbell and Thompson (2008) and Filippou et al. (2020).

Table 8: Statistical Performance. In this table, we report the out-of-sample statistical performance of five neural network prediction models, namely PRED-TSAN-FC and four comparison models, against the random walk benchmark. For each prediction model, we consider three different architectures with varying complexities, namely the number of hidden layers ranging from 1 to 3.

Number of Hidden Layers	Model	RMSFE	MAE	Theil U	R^2 (%)
	Random Walk	2.63	2.07	1	0
1	PRED-FC	3.22	2.54	1.225	-50.02
2		3.09	2.41	1.175	-38.13
3		2.94	2.31	1.120	-25.49
1	PRED-TSAN	2.69	2.12	1.024	-4.85
2		2.67	2.11	1.017	-3.48
3		2.69	2.13	1.025	-4.99
1	PRED-PC	3.50	2.78	1.331	-77.15
2		3.34	2.63	1.272	-61.75
3		3.19	2.51	1.214	-47.35
1	PRED-TSAN-PC	2.68	2.11	1.021	-4.31
2		2.65	2.09	1.009	-1.74
3		2.64	2.07	1.002	-0.48
1	PRED-TSAN-FC	2.62	2.05	0.996	0.72
2		2.64	2.08	1.006	-1.15
3		2.65	2.09	1.008	-1.68

data-intensive tasks in which deep learning thrives. This can also be supported by the outperformance of PRED-TSAN-FC with less hidden layers compared to the deeper versions.

3.4.3 Economic Performance Evaluation

Similar to the performance evaluation of latent factor model, we further look into the economic performance of prediction models by constructing long-short portfolios based on the out-of-sample predictive returns. The currency excess return is defined as described in (26). For each prediction model, we construct two zero-investment long-short portfolios: the long-one-short-one portfolio and the long-two-short-two portfolio, which are equal-weighted and monthly rebalanced.

Table 9 reports the out-of-sample Sharpe ratios of the two long-short portfolios for each prediction model with different complexities. Comparing the performance of the long-one-short-one portfolios, the models without TSAN architectures, namely PRED-FC and PRED-PC, obviously underperform the other three neural network prediction models, which is consistent with the statistical performance shown in Table 8. The highest Sharpe ratio is delivered by PRED-TSAN with one hidden layer and PRED-TSAN-PC with three hidden layers. Based on the results, the economic performance does not necessarily conform with the statistical performance, in other words, a higher R^2 (or lower RMSFE) does not necessarily translate to a higher Sharpe ratio in long-short portfolios. This might be a result of the highly volatile performance of prediction models, despite all the regularization and ensemble methods used. When considering the average performance over different complexities, PRED-TSAN-FC delivers the best overall performance among the candidate prediction models. The performance of long-two-short-two portfolios is obviously less volatile compared to the long-one-short-one portfolios, with all neural network models generating positive Sharpe ratios. The highest Sharpe ratio is still delivered by PRED-TSAN but with two hidden layers, while PRED-TSAN-FC retains the best average performance.

3.4.4 Interactive Effects and Characteristic Importance

Similar to the latent factor model, we assess the characteristic importance and cross-country interactive effects based on the characteristic sensitivity defined in (28) and (29).

Table 9: Sharpe Ratios of Long-Short Portfolios. In this table, we report the Sharpe ratios of the long-one-short-one and long-two-short-two zero-investment portfolios for the five neural network prediction models, namely the PRED-TSAN-FC and four comparison models, against the random walk benchmark. For each prediction model, we consider three different architectures - the model with hidden layers ranging from 1 to 3.

Number of Hidden Layers	Model	Long-One-Short-One	Long-Two-Short-Two
	Random Walk	0.24	0.26
1	PRED-FC	0.16	0.35
2		-0.10	0.06
3		0.01	0.45
1	PRED-TSAN	0.42	0.26
2		-0.07	0.50
3		0.01	0.12
1	PRED-PC	0.12	0.33
2		-0.05	0.23
3		-0.11	0.33
1	PRED-TSAN-PC	-0.04	0.18
2		0.30	0.17
3		0.42	0.23
1	PRED-TSAN-FC	0.20	0.37
2		0.34	0.35
3		0.26	0.48

Figure 17 illustrates the sensitivity of output predicted returns with respect to all characteristics for each currency, based on a PRED-TSAN-FC model with one hidden layer, namely the architecture with the highest out-of-sample R^2. In the deep prediction model, there are still evident cross-country interactive patterns as shown by the vertical color pattern in the heatmaps, which is consistent with results from the latent factor model and indicates the cross-country interactions between the characteristics and the monthly exchange rate movements.

We further examine the characteristic importance in the deep prediction model based on the overall sensitivity defined in (30). Figure 18 illustrates the top 50 characteristics ranked by their characteristic importance. The top 50 characteristics contributes to around 20% of total characteristic importance, with no single characteristic significantly dominating. We also examine the characteristic groups as shown in Figure 19, which are similarly categorized as in Section 3.3 and based on Table A1. The results are consistent with the latent factor model, all the top characteristic groups are technical or interest rate-related, indicating the dominance of the relatively high frequency rate-related characteristics in driving monthly exchange rate movements.

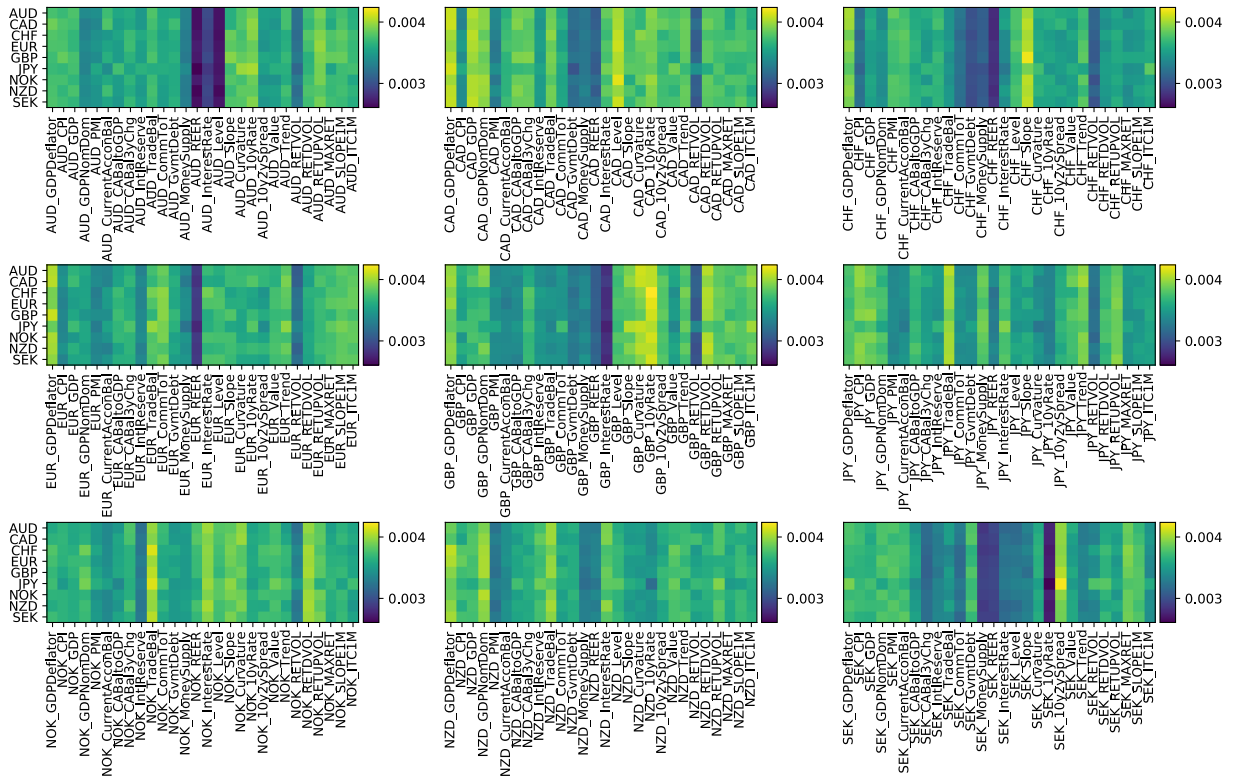
Figure 20 describes the evolution of characteristic importance over time during the out-of-sample periods. Although the rate-related characteristic groups seem to exhibit a relatively greater importance compared to macro characteristic groups, the differences are not as obvious as in the latent factor model. This might be a result of the notoriously low signal-to-noise ratio in return prediction compared to cross-sectional contemporaneous description.

3.5 Two Models Side by Side

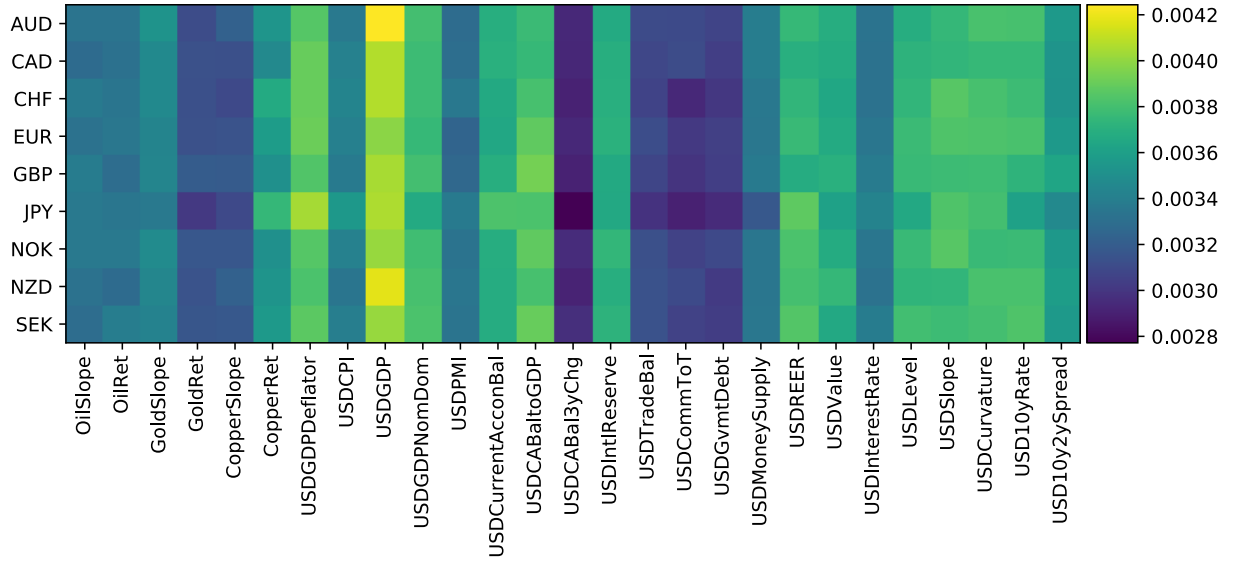
Although sharing some homogeneous constituent architectures, the deep latent factor model and the deep prediction model are fundamentally different in their design initiatives. The deep latent factor model is constructed with a predefined risk-return factor structure in which both the factor exposure and factor return are functions of asset characteristics, with the goal of reconstructing asset returns. While the deep prediction model seeks to minimize the forecasting error for future terms without taking into account the risk-return trade-off, it primarily captures the complex nonlinear relationships between future returns and asset characteristics.

Following the economic interpretation of factor models, the deep latent factor model provides a compact

Figure 17: Cross-Country Interactive Effects. The figure reports the sensitivity of output predicted returns to all characteristics for each currency. Figure (a) reports the sensitivity to country-specific characteristics. Each sub-heatmap illustrates the sensitivity of predicted returns for all nine currencies to the characteristics of a particular country. Figure (b) reports the sensitivity to common characteristics, namely the US and commodity-related characteristics.



(a) Sensitivities to Country-Specific Characteristics



(b) Sensitivities to Common Characteristics

Figure 18: Characteristic Importance. The figure reports the top 50 characteristics among the total 279 characteristics ranked by their characteristic importance. Different bar colors depict characteristics of different countries.

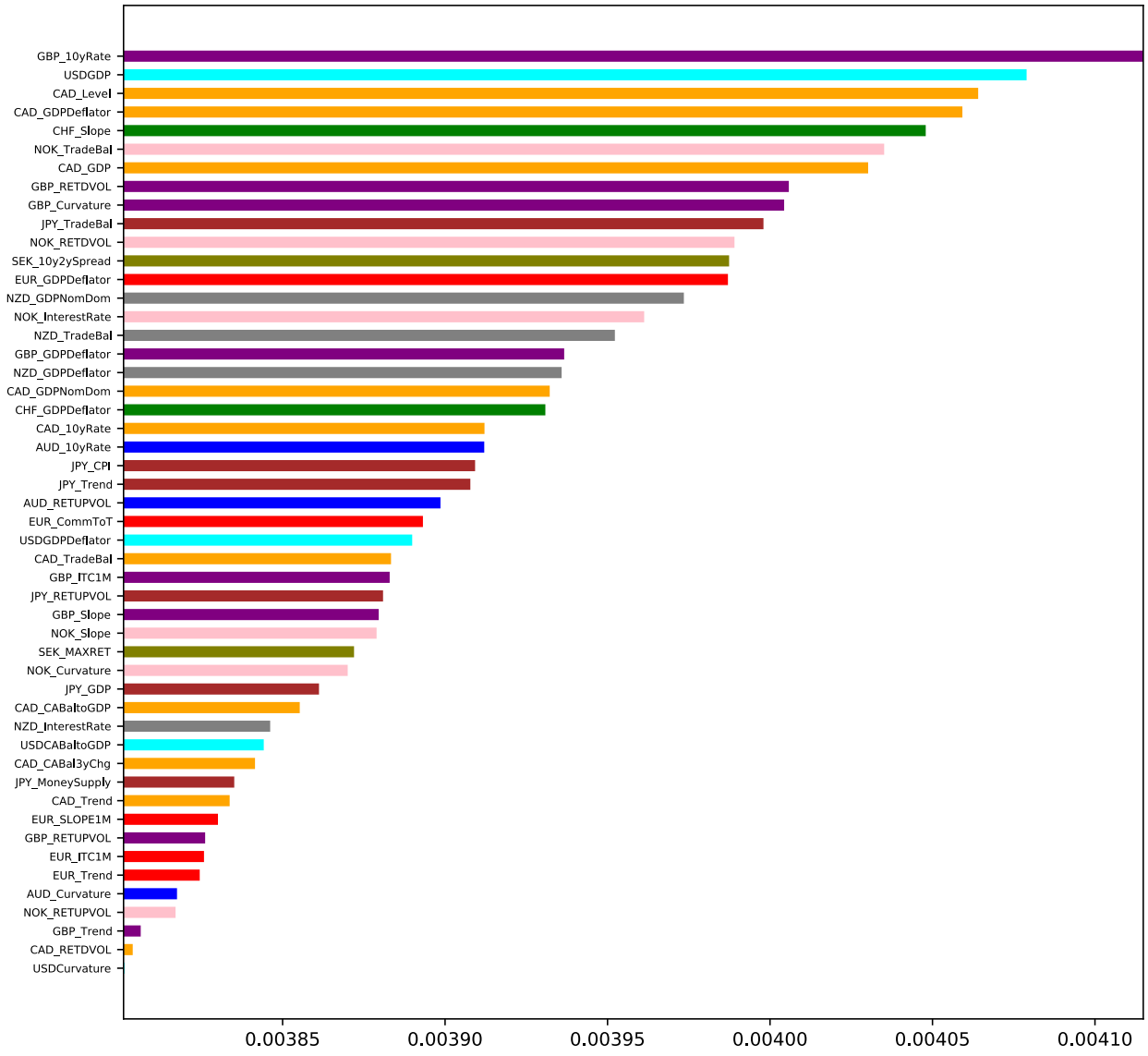


Figure 19: Characteristic Importance by Group. The figure reports the average characteristic importance of each group. The characteristics are classified into four categories: macroeconomic, interest rate-related, technical, and commodity-related, with the first three further grouped by country.

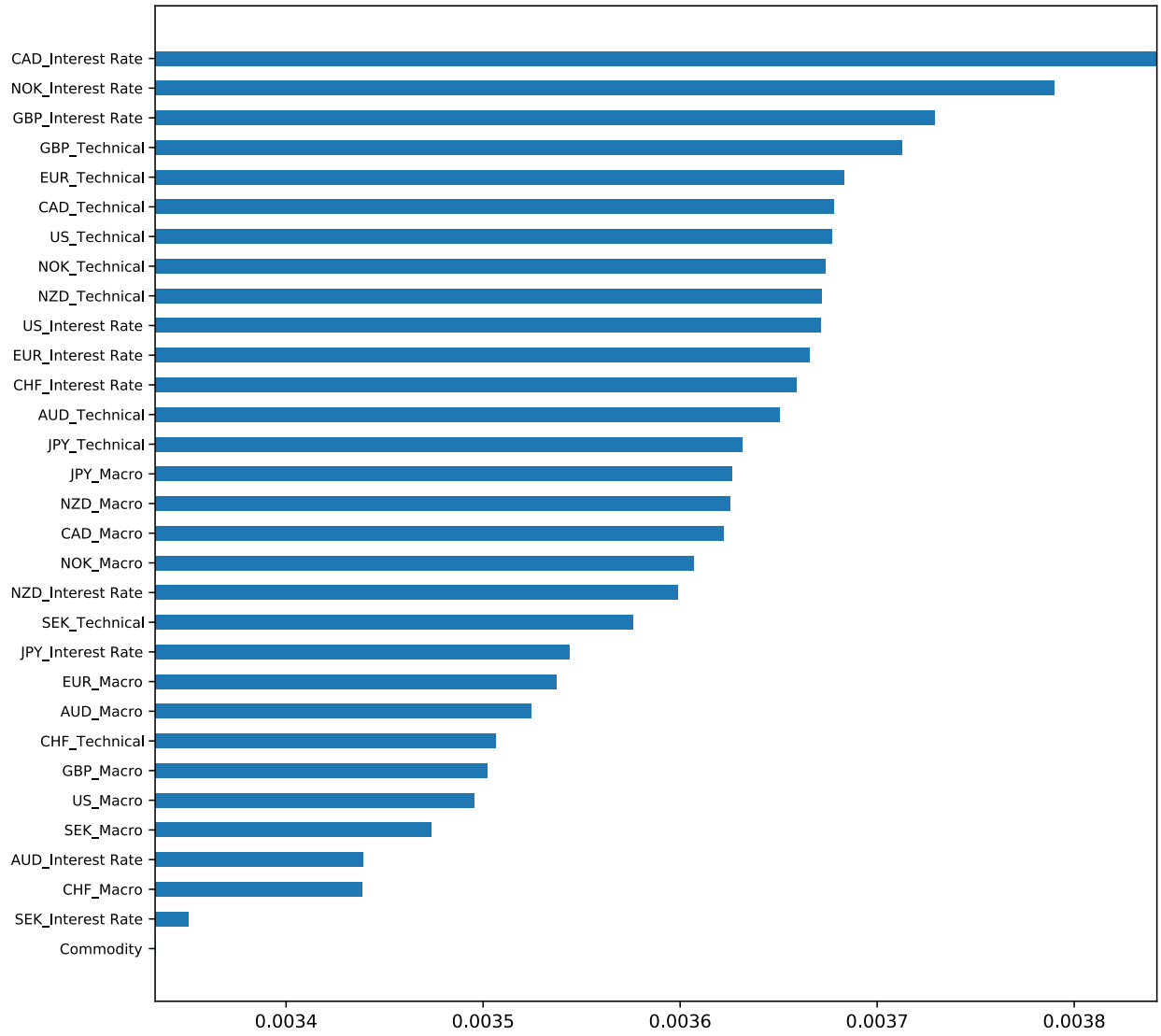
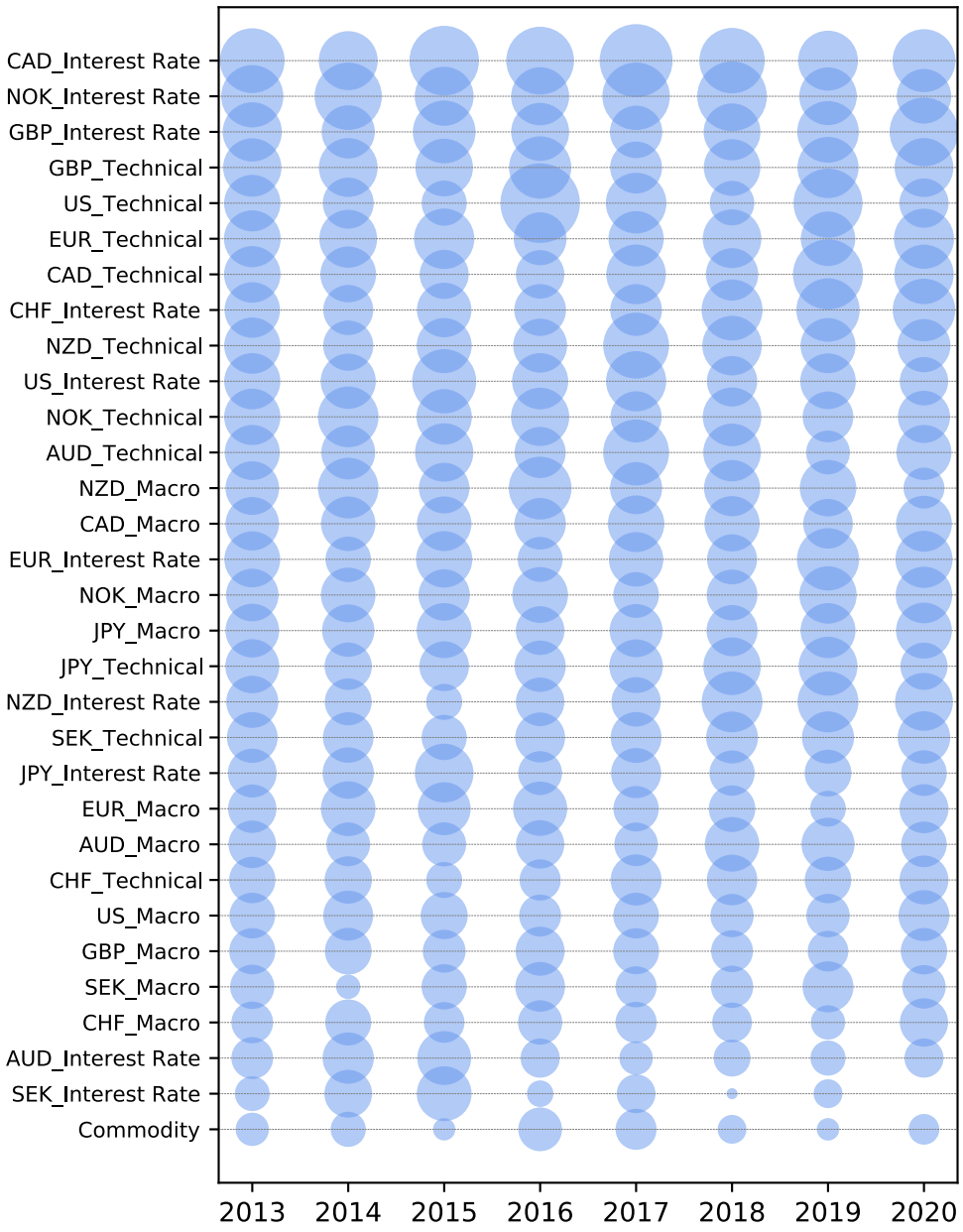


Figure 20: Characteristic Importance over Time. The figure reports the average characteristic importance of the groups for each year during the out-of-sample periods, based on the results of PRED-TSAN-FC with one hidden layer. It depicts how the rankings of dominant characteristic groups evolve over time. A larger point in the figure indicates a greater importance of the characteristic.



statistical description of the cross-sectional dependence structure, making it an excellent candidate for contemporaneous explanation. However, when forecasting future terms, as indicated by the RMSE_{Pred} in Table 4 and RMSFE in Table 8, the prediction model emerges as the superior alternative, albeit at the expense of economic sensibility.

In both models, the sequence modelling architecture, which is used to extract the hidden state from historical characteristics, exhibits a considerable beneficial effect on performance enhancement. The characteristic sensitivity suggests a clear cross-country interaction in both models with the influential characteristics consistently dominated by rate-related groups.

4 Conclusion

In this paper, we explore the asset pricing problem in the Forex market from two canonical perspectives, namely cross-sectional description and time series prediction. To accomplish this, we propose two asset pricing models in the context of deep learning, the deep latent factor model and deep prediction model. Our deep latent factor model maintains the economic interpretation of factor models and extends the work of Kelly et al. (2019) and Gu et al. (2021). The model adapts to the Forex market and imposes economic restriction of no-arbitrage. Our deep prediction model explores the efficiency of deep learning in forecasting exchange rate movements, without taking into account the risk-return trade-off. Both models employ sequence modelling techniques to capture information from historical characteristics and allows for cross-country interactions.

In our empirical analysis of G10 currencies, both the deep latent factor model and deep prediction model provide superior performance compared to models in the comparison set. The characteristic sensitivity analysis reveals obvious cross-country interactive patterns and identifies influential characteristics that consistently dominate. In our comparative analysis of model architectures, we highlight the beneficial effects of the time-series attention network (TSAN) for both problems, suggesting the critical importance of incorporating information from long-range historical data for both cross-sectional description and time-series prediction problems.

References

Bahdanau, D., Cho, K., Bengio, Y., 2014. Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 .

Bai, J., Ng, S., 2002. Determining the number of factors in approximate factor models. Econometrica 70, 191–221.

Barunik, J., Ellington, M., et al., 2020. Dynamic networks in large financial and economic systems. arXiv preprint arXiv:2007.07842 .

Bishop, C.M., 2006. Pattern recognition. Machine Learning 128.

Branger, N., Konermann, P., Meinerding, C., Schlag, C., 2021. Equilibrium asset pricing in directed networks. Review of Finance 25, 777–818.

Campbell, J.Y., Thompson, S.B., 2008. Predicting excess stock returns out of sample: Can anything beat the historical average? The Review of Financial Studies 21, 1509–1531.

Carhart, M.M., 1997. On persistence in mutual fund performance. The Journal of Finance 52, 57–82.

Chen, L., Pelger, M., Zhu, J., 2020. Deep learning in asset pricing. Available at SSRN 3350138 .

Chinco, A., Clark-Joseph, A.D., Ye, M., 2019. Sparse signals in the cross-section of returns. The Journal of Finance 74, 449–492.

Chung, J., Gulcehre, C., Cho, K., Bengio, Y., 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555 .

Cong, L.W., Tang, K., Wang, J., Zhang, Y., 2021a. Alphaportfolio: Direct construction through deep reinforcement learning and interpretable ai. Available at SSRN 3554486 .

Cong, L.W., Tang, K., Wang, J., Zhang, Y., 2021b. Deep sequence modeling: Development and applications in asset pricing. The Journal of Financial Data Science 3, 28–42.

Connor, G., Korajczyk, R.A., 1986. Performance measurement with the arbitrage pricing theory: A new framework for analysis. Journal of Financial Economics 15, 373–394.

Engel, C., Mark, N.C., West, K.D., 2015. Factor model forecasts of exchange rates. Econometric Reviews 34, 32–55.

Engel, C., Mark, N.C., West, K.D., Rogoff, K., Rossi, B., 2007. Exchange rate models are not as bad as you think [with comments and discussion]. NBER Macroeconomics Annual 22, 381–473.

Fama, E.F., French, K.R., 1993. Common risk factors in the returns on stocks and bonds. Journal of Financial Economics 33, 3–56.

Faust, J., Rogers, J.H., Wright, J.H., 2003. Exchange rate forecasting: the errors we've really made. Journal of International Economics 60, 35–59.

Filippou, I., Rapach, D., Taylor, M.P., Zhou, G., 2020. Exchange rate prediction with machine learning and a smart carry portfolio. Available at SSRN 3455713 .

Fischer, T., Krauss, C., 2018. Deep learning with long short-term memory networks for financial market predictions. European Journal of Operational Research 270, 654–669.

Freyberger, J., Neuhierl, A., Weber, M., 2020. Dissecting characteristics nonparametrically. The Review of Financial Studies 33, 2326–2377.

Gofman, M., Segal, G., Wu, Y., 2020. Production networks and stock returns: The role of vertical creative destruction. The Review of Financial Studies 33, 5856–5905.

Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep Learning. MIT Press. http://www.deeplearningbook.org.

Gu, S., Kelly, B., Xiu, D., 2018. Empirical asset pricing via machine learning. Technical Report. National Bureau of Economic Research.

Gu, S., Kelly, B., Xiu, D., 2021. Autoencoder asset pricing models. Journal of Econometrics 222, 429–450.

Hinton, G.E., Salakhutdinov, R.R., 2006. Reducing the dimensionality of data with neural networks. science 313, 504–507.

Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural Computation 9, 1735–1780.

Hornik, K., 1991. Approximation capabilities of multilayer feedforward networks. Neural Networks 4, 251–257.

Hornik, K., Stinchcombe, M., White, H., 1989. Multilayer feedforward networks are universal approximators. Neural Networks 2, 359–366.

Hou, K., Xue, C., Zhang, L., 2015. Digesting anomalies: An investment approach. The Review of Financial Studies 28, 650–705.

Isard, P., et al., 1983. What's wrong with empirical exchange rate models: some critical issues and new directions. Technical Report.

Kelly, B.T., Pruitt, S., Su, Y., 2019. Characteristics are covariances: A unified model of risk and return. Journal of Financial Economics 134, 501–524.

Kelly, B.T., Xiu, D., 2021. Factor models, machine learning, and asset pricing. Machine Learning, and Asset Pricing (October 15, 2021) .

Kilian, L., Taylor, M.P., 2003. Why is it so difficult to beat the random walk forecast of exchange rates? Journal of International Economics 60, 85–107.

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

Kreinovich, V.Y., 1991. Arbitrary nonlinearity is sufficient to represent all functions by neural networks: a theorem. Neural Networks 4, 381–383.

Lettau, M., Pelger, M., 2020a. Estimating latent asset-pricing factors. Journal of Econometrics 218, 1–31.

Lettau, M., Pelger, M., 2020b. Factors that fit the time series and cross-section of stock returns. The Review of Financial Studies 33, 2274–2325.

Mark, N.C., Sul, D., 2001. Nominal exchange rates and monetary fundamentals: evidence from a small post-bretton woods panel. Journal of International Economics 53, 29–52.

Meese, R., Rogoff, K., 1983a. The out-of-sample failure of empirical exchange rate models: sampling error or misspecification?, in: Exchange rates and international macroeconomics. University of Chicago Press, pp. 67–112.

Meese, R., Rogofp, K., 1988. Was it real? the exchange rate-interest differential relation over the modern floating-rate period. The Journal of Finance 43, 933–948.

Meese, R.A., Rogoff, K., 1983b. Empirical exchange rate models of the seventies: Do they fit out of sample? Journal of International Economics 14, 3–24.

Moosa, I., 2013. Why is it so difficult to outperform the random walk in exchange rate forecasting? Applied Economics 45, 3340–3346.

Ross, S.A., 1976. The arbitrage theory of capital asset pricing. Journal of Economic Theory 13, 341–60.

Rossi, B., 2013. Exchange rate predictability. Journal of Economic Literature 51, 1063–1119.

Rossi, B., et al., 2006. Are exchange rates really random walks? some evidence robust to parameter instability. Macroeconomic Dynamics 10, 20.

Schinasi, G.J., Swamy, P.A.V.B., 1989. The out-of-sample forecasting performance of exchange rate models when coefficients are allowed to change. Journal of International Money and Finance 8, 375–390.

Stock, J.H., Watson, M.W., 1996. Evidence on structural instability in macroeconomic time series relations. Journal of Business & Economic Statistics 14, 11–30.

Stock, J.H., Watson, M.W., 1999. Business cycle fluctuations in us macroeconomic time series. Handbook of Macroeconomics 1, 3–64.

Theil, H., 1966. Applied economic forecasting. North Holland .

Tibshirani, R., 1996. Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society: Series B (Methodological) 58, 267–288.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I., 2017. Attention is all you need, in: Advances in neural information processing systems, pp. 5998–6008.

Appendix

Table A1: Characteristics List

Macroeconomic Fundamental Characteristics	Frequency	Abbreviation in Model
CPI	Monthly	CPI
GDP Deflator Change	Quarterly	GDPDeflator
PMI	Monthly	PMI
Money Supply	Monthly	MoneySupply
International Trade Balance	Monthly	TradeBal
Government Debt	Annually	GvmtDebt
Commodity Terms of Trade	Monthly	CommToT
Real GDP Growth	Quarterly	GDP
GDP Nominal Demestic Currency	Quarterly	GDPNomDom
IMF's Estimate of Real Effective Exchange Rate	Monthly	REER
Total International Reserves, US Dollars	Monthly	IntlReserve
Current Account Balance, US Dollars	Quarterly	CurrentAcconBal
Current Account Balance, % of GDP	Quarterly	CABaltoGDP
Current Account 3-Year Change in Balance, % of GDP	Quarterly	CABal3yChg

Interest Rate Related Characteristics	Frequency	Abbreviation in Model
Short-term Interest Rate	Monthly	InterestRate
10-year Interest Rate	Monthly	10yRate
10y-2y Term Spread	Monthly	10y2ySpread
Term Structure of Interest Rates, Level	Monthly	Level
Term Structure of Interest Rates, Slope	Monthly	Slope
Term Structure of Interest Rates, Curvature	Monthly	Curvature

Technical Characteristics	Frequency	Abbreviation in Model
Value	Monthly	Value
Trend	Monthly	Trend
Exchange Rate Return Volatility in 1 Month	Monthly	RETVOL
Exchange Rate Return Downside Volatility in 1 Month	Monthly	RETDVOL
Exchange Rate Return Upside Volatility in 1 Month	Monthly	RETUPVOL
Maximum Exchange Rate Return in 1 Month	Monthly	MAXRET
Slope of Exchange Rate in 1 Month	Monthly	SLOPE1M
Intercept of Exchange Rate Return in 1 Month	Monthly	ITC1M

Commodities Related Characteristics	Frequency	Abbreviation in Model
ICE Brent Crude, 1-Month Return	Monthly	OilRet
COMEX Gold, 1-Month Return	Monthly	GoldRet
COMEX Copper, 1-Month Return	Monthly	CopperRet
ICE Brent Crude, Slope of Price in 1 Month	Monthly	OilSlope
COMEX Gold, Slope of Price in 1 Month	Monthly	GoldSlope
COMEX Copper, Slope of Price in 1 Month	Monthly	CopperSlope

Table A2: List of Country-Specific Characteristics

Country-Specific Characteristics	Abbreviation in Model
GDP Deflator Change	GDPDeflator
CPI	CPI
Real GDP Growth	GDP
GDP Nominal Demestic Currency	GDPNomDom
PMI	PMI
Current Account Balance, US Dollars	CurrentAcconBal
Current Account Balance, % of GDP	CABaltoGDP
Current Account 3-Year Change in Balance, % of GDP	CABal3yChg
Total International Reserves, US Dollars	IntlReserve
International Trade Balance	TradeBal
Commodity Terms of Trade	CommToT
Government Debt	GvmtDebt
Money Supply	MoneySupply
IMF's Estimate of Real Effective Exchange Rate	REER
Value	Value
Short-term Interest Rate	InterestRate
Term Structure of Interest Rates, Level	Level
Term Structure of Interest Rates, Slope	Slope
Term Structure of Interest Rates, Curvature	Curvature
10-year Interest Rate	10yRate
10y-2y Term Spread	10y2ySpread
Trend	Trend
Exchange Rate Return Volatility in 1 Month	RETVOL
Exchange Rate Return Downside Volatility in 1 Month	RETDVOL
Exchange Rate Return Upside Volatility in 1 Month	RETUPVOL
Maximum Exchange Rate Return in 1 Month	MAXRET
Slope of Exchange Rate in 1 Month	SLOPE1M
Intercept of Exchange Rate Return in 1 Month	ITC1M

Table A3: List of Common Characteristics

Common Characteristics	Abbreviation in Model
ICE Brent Crude, 1-Month Return	OilSlope
COMEX Gold, 1-Month Return	OilRet
COMEX Copper, 1-Month Return	GoldSlope
ICE Brent Crude, Slope of Price in 1 Month	GoldRet
COMEX Gold, Slope of Price in 1 Month	CopperSlope
COMEX Copper, Slope of Price in 1 Month	CopperRet
US GDP Deflator Change	USDGDPDeflator
US CPI	USDCPI
US Real GDP Growth	USDGDP
US GDP Nominal Demestic Currency	USDGDPNomDom
US PMI	USDPMI
US Current Account Balance, US Dollars	USDCurrentAcconBal
US Current Account Balance, % of GDP	USDCABaltoGDP
US Current Account 3-Year Change in Balance, % of GDP	USDCABal3yChg
US Total International Reserves, US Dollars	USDIntlReserve
US International Trade Balance	USDTradeBal
US Commodity Terms of Trade	USDCommToT
US Government Debt	USDGvmtDebt
US Money Supply	USDMoneySupply
US IMF's Estimate of Real Effective Exchange Rate	USDREER
US Value	USDValue
US Short-term Interest Rate	USDInterestRate
US Term Structure of Interest Rates, Level	USDLevel
US Term Structure of Interest Rates, Slope	USDSlope
US Term Structure of Interest Rates, Curvature	USDCurvature
US 10-year Interest Rate	USD10yRate
US 10y-2y Term Spread	USD10y2ySpread