AMRO
ASEAN +3 MACROECONOMIC RESEARCH OFFICE

Working Paper (WP/25-08)

Shock Propagation in LSTM Multivariate Time Series Systems

Jorge A. Chan-Lau and T. Long Quach

September 2025

[This page is intentionally left blank]

Shock Propagation in LSTM Multivariate Time Series Systems

Jorge A. Chan-Lau and T. Long Quach*

September 2025

Disclaimer: The findings, interpretations, and conclusions expressed in this paper represent the views of the authors and are not necessarily those of the ASEAN+3 Macroeconomic Research Office (AMRO) or its member authorities. Neither AMRO nor its member authorities shall be held responsible for any consequences from the use of the information contained therein.

Abstract

Long-short term memory networks (LSTMs) offer a promising approach for analyzing shock propagation in multivariate economic and financial systems. To harness this potential, we introduce the LSTM multiplier response function, analogous to the impulse response function of a standard linear VAR model. This response function presents several advantages. Firstly, it more effectively captures nonlinear dynamics inherent in complex systems compared to linear VAR models. Secondly, it accounts for the system's current and historical states, reflecting the intuition that negative shocks have amplified effects under adverse conditions. Thirdly, the method involves applying shocks directly to variables of interest, obviating the need for establishing causality or orthogonalizing the system. To illustrate, we compare LSTM and VAR models by fitting them to a multivariate economic system. Leveraging the superior forecasting accuracy of the LSTM, we demonstrate that the LSTM multiplier response function exhibits similar qualitative features to VAR impulse responses, highlighting its usefulness in economic and financial applications.

Keywords: LSTM, VAR, impulse response, LSTM multiplier, LSTM multiplier response, multivariate time series

JEL codes: C32, C45, C53

*ASEAN+3 Macroeconomic Research Office, 10 Shenton Way 15-08 MAS Building, Singapore 079117. We thank comments and suggestions from Aruhan Rui Shi, Harold Glenn A. Valera, Jerry Hwang and participants at the *2023 Workshop on Recent Developments and Future Prospects for the International Monetary System*, Hong Kong; and AMRO internal seminars. We are solely responsible for any errors or omissions. Please address correspondence to both authors: Chan-Lau, jchanlau@gmail.com (permanent) and jorge.chan-lau@amro-asia.org; Long Quach, quachtoanlong@gmail.com (permanent) and quach.toanlong@amro-asia.org.

Contents

List of Figures

List of Tables

1 Introduction

Multivariate time series analysis serves as an essential methodology in economics and finance, enabling researchers to model relationships between multiple variables that evolve simultaneously over time. These techniques enhance our understanding of economic systems by capturing interdependencies reflecting feedback mechanisms and spillover effects that univariate approaches cannot address (Sims 1980; Stock and Watson 2001). Understanding these interdependencies has led the development of specialized econometric methods such as, among others, vector autoregressions (VAR) (Sims 1980), regime switching models (Hamilton 1989), and conditional heteroskedastic models (Engle 1982; Bollerslev 1986). These methods and their extensions have become standard tools in applied economics.

Outside traditional econometrics, deep learning, a subset of machine learning that relies on neural networks with multiple layers, has been increasingly applied to the modeling of complex, non-linear patterns in data across domains such as computer vision and sequential analysis (LeCun, Bengio, and Hinton 2015; Goodfellow, Bengio, and Courville 2016). Deep learning models, as well as deep learning-based artificial intelligence models, are increasingly adopted in economics, including diverse applications in text and image analysis, causal inference, dynamic stochastic general equilibrium models, and policy evaluation under bounded rationality (Guo et al. 2023; Korinek 2023; Shi 2023; Fernández-Villaverde, Nuño, and Perla 2024; Dell 2025). In particular, deep learning architectures perform well at capturing long-term dependencies and might be well-suited in time series analysis (Lim and Zohren 2021).

One widely used deep learning model suitable for time series analysis is the Long Short-Term Memory network (LSTM) first introduced by Hochreiter and Schmidhuber (1997). The multilayer architecture of the LSTMs can capture well non-linear relationships in the data and its gate structure allows it to remember relevant long-term dependencies. The gate structure is a set of three functions that determine what new information is relevant (the input gate), what past information is no longer useful and can be discarded (the forget gate), and what past information is still important for the current prediction. Because LSTMs update their long-term memory as new data becomes available they can adapt to dynamic and evolving market and economic conditions. Moreover, they are able to handle a large number of features (explanatory variables) even if observed at different frequencies. Unsurprisingly, they are increasingly being used in economic and financial forecasting (Fischer and Krauss 2018; Shamsi et al. 2021; Hopp 2022; Kumarappan et al. 2024).

This paper, building on the authors' earlier work (Chan-Lau and Quach 2023), investigates the effectiveness of LSTMs for analyzing shock propagation in multivariate time series systems. To this end, it introduces the concept of the LSTM multiplier, defined as the difference between the LSTM forecasts when one variable is subjected to a shock and the one that would have occurred in the absence of such a shock, and the concept of the LSTM multiplier response, which shows the LSTM multiplier values for different time horizons. The multiplier response serves as the analogue of the impulse response function of a VAR model, over which it has two advantages. First, the multiplier response behavior is not invariant to current and past economic or financial conditions. Second, the shocks are applied directly to the variables in the system making unnecessary to establish causality or orthogonalizing the system, as VARs require.

We estimate the LSTM and a standard linear VAR model for monthly and quarterly US macroeconomic data. The estimation pursues two goals: first, to evaluate whether the LSTM provides superior forecasting performance; and second, to benchmark the LSTM multiplier response against the conventional impulse response function derived from the VAR. Our results indicate that the LSTM outperforms the VAR in forecasting accuracy at a quarterly frequency by a substantial margin. At the monthly frequency, both models perform similarly, with the VAR holding a slight edge. The LSTM multiplier response closely resembles that of the VAR impulse response functions for most shocks. In the cases where they differ, we argue the differences are driven by the fact that the LSTM responses are conditional on current and past conditions. Finally, we suggest that given the superior forecasting performance of the LSTM, its multiplier could be more accurate when evaluating how shocks propagate in the system.

In the remainder of the paper, Section 2 reviews some of the applications of LSTMs in time series analysis, and Section 3 describes the basic LSTM layer, which serves as the building block of the LSTM, introduces the concept of the LSTM multiplier and explains the hyperparameter tuning process. Section 4 discusses the empirical implementation of the model, including the dataset used and the hyperparameter space. Results are analyzed in Section 5, including a comparison of those corresponding to the LSTM and VAR models. Finally, Section 6 concludes.

2 A brief literature review

LSTM models have proved effective for single-variable forecasting, including univariate and multivariate time series models. Because they capture nonlinear patterns in the data well they often outperform traditional models like exponential smoothing and ARIMA models. In economic applications, LSTMs have exhibited good performance in forecasting GDP growth (Hamiane et al. 2023; Zhang, Wen, and Yang 2022; Hamiane et al. 2024; Xie et al. 2024; Zhao 2024), economic crises (Park and Yang 2022), and consumer price inflation over medium and long-term horizons (Lakshmi Narayanaa et al. 2023; Zhao 2024; Liu and Lan 2025; Paranhos 2025).

LSTMs also tend to perform well in financial applications (Buczynski et al. 2023). For example, LSTMs yield better forecasts of stock prices (Serin and Kemalbay 2024; Furizal et al. 2024; Pilla and Mekonen 2025) especially over short-term horizons (Kobiela et al. 2022). They have proved better than univariate time series models in financial risk prediction (Xu et al. 2024), interest rate forecasting (Salem, Jummah, and Albourawi 2024) as well as other areas (Siami-Namini, Tavakoli, and Namin

2018). Compared to the benchmark volatility model, the Generalized Autoregressive Conditional Heteroskedasticity (GARCH) model of Bollerslev (1986), LSTM provides better estimates of the value-at-risk (VaR) (Ormaniec et al. 2022). In high frequency series, however, LSTMs may not do as well as volatility models (Sezer, Gudelek, and Ozbayoglu 2020).

LSTMs have also been used in multivariate time series analysis to capture possible nonlinear dependencies among different variables in an economic or financial system. Cao, Li, and Li (2019) find that deep learning models of financial time series, including LSTMs, outperform VAR models, especially during high volatility periods during which nonlinearities are likely more pronounced that during calm periods. In economics, the results in Hopp (2022) suggest that LSTMs are better than dynamic factor models (DFMs) for nowcasting global services exports, global merchandise export values and volumes.

LSTM increasingly serve as one of the foundational components of hybrid forecasting models, a rapidly growing research area (Buczynski et al. 2023). Some examples of this work include Hollis, Viscardi, and Yi (2018), X. Zhang et al. (2019), and Ju and Liu (2021), who show that including an attention mechanism improves the forecasting accuracy of LSTMs. Lashina and Grishunin (2023) evaluate various popular forecasting models, including ARIMA, LSTM, VAR, support vector regression (SVR), and CatBoost, as well as an ensemble combining a DFM and an LSTM; among them, the latter performs best in predicting GDP growth. Sivakumar (2025), using Hidden Markov Models (HMMs), extract hidden states representing distinct economic conditions and find that including them as additional features within a LSTM improves the accuracy of economic forecasts. In addition, current research is exploring new LSTM architectures to enhance time series forecasting (Kong et al. 2024, Kraus et al. 2024).

Despite the potential advantages of using LSTMs in time series analysis there are some caveats to consider. Compared with standard time series analysis the design and training of LSTMs is complex and may require larger amounts of data to achieve optimal performance. Due to the complexity of their internal architecture, it is sometimes difficult to uncover or interpret the underlying economic relationships driving the system dynamics in a LSTM. In fact, LSTMs, as other deep learning models, are sometimes viewed as "black boxes." In response, ongoing scholarly work focuses on model explainability to enhance the ability to elucidate and communicate the rationale behind a model's predictions in a manner comprehensible to humans (Y. Zhang et al. 2021; Ji et al. 2025; Molnar 2025).

3 The LSTM architecture and LSTM multiplier

Economists, market analysts, and policy makers are interested in evaluating how shocks propagate in a multivariate time series system. The most widely used approach is to fit a VAR to the data. Afterwards either the impulse response function (Sims 1980) or local projections (Jordà 2005) are used to examine the impact of shocks on the time series dynamics. This section examines shock propagation in an LSTM multivariate system. To this end, the section introduces the concept of the LSTM multiplier (LSTMm) (Chan-Lau and Quach 2023) and the LSTM multiplier response (LMR).

3.1 The LSTM block

An LSTM is a specialized recurrent neural network designed to model long-term dependencies in the data (Hochreiter and Schmidhuber 1997). It has proved particularly good at processing and predicting sequences of data, like text or time series. What makes LSTMs special is their ability to "remember" important information for long periods while "forgetting" irrelevant details. This is achieved through a complex internal structure with a series of functions, or gates, that control the flow of information.

Figure 1 shows the flow of information inside the basic LSTM block architecture of Greff et al. (2017) which serves as the building block of the LSTM multivariate system used in the analysis.[1] In any period t, the information the LSTM uses is contained in three components: the input $\mathbf{X}_t$, or the new information; the long-term memory or memory cell $\mathbf{C}_{t-1}$; and the short-term memory or hidden state $\mathbf{H}_{t-1}$. The latter two are information available to the block and prior to receiving the new information. Intuitively, the state of the memory cell carries information from the distant past through the entire sequence of observations allowing the LSTM to remember it over long periods of time. The hidden state captures what the LSTM considers important information, conditional on the new information and its long-term memory. The information processing by the hidden state yields the short-term memory of the block.

The LSTM block processes the information in batches, that is, n observations at a time. Assuming that there are k inputs in each observation, $\mathbf{X} \in \mathbb{R}^{n \times k}$.[2] It is also assumed that the short-term and long-term memories are captured adequately with h features so $\mathbf{H}, \mathbf{C} \in \mathbb{R}^{n \times h}$. The input $\mathbf{I}$, forget $\mathbf{F}$, and output $\mathbf{O}$ gates are defined as follows:

$$\mathbf{I}_t\left(\mathbf{X}_t, \mathbf{H}_{t-1}\right) = \sigma\left(\mathbf{X}_t \mathbf{W}_{xi} + \mathbf{H}_{t-1} \mathbf{W}_{hi} + \mathbf{b}_i\right), \tag{1}$$

$$\mathbf{F}_t\left(\mathbf{X}_t, \mathbf{H}_{t-1}\right) = \sigma\left(\mathbf{X}_t \mathbf{W}_{xf} + \mathbf{H}_{t-1} \mathbf{W}_{hf} + \mathbf{b}_f\right), \tag{2}$$

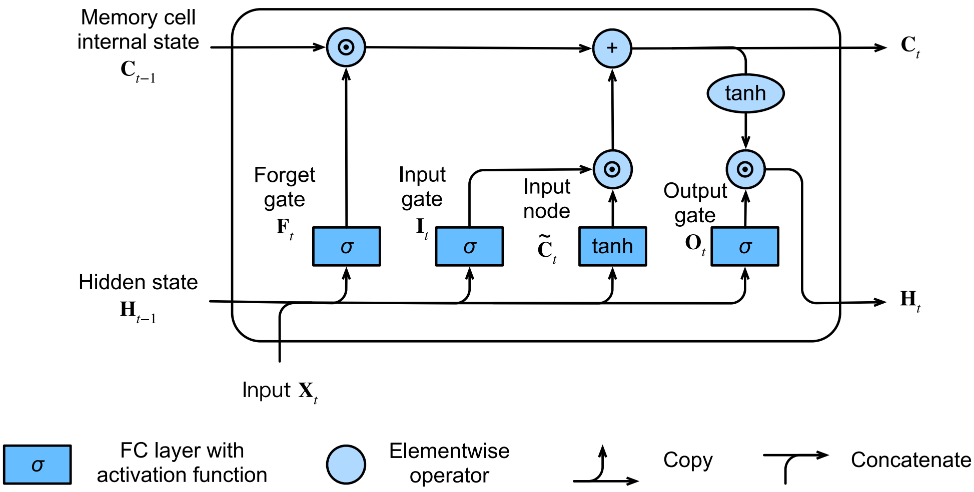
$$\mathbf{O}_t\left(\mathbf{X}_t, \mathbf{H}_{t-1}\right) = \sigma\left(\mathbf{X}_t \mathbf{W}_{xo} + \mathbf{H}_{t-1} \mathbf{W}_{ho} + \mathbf{b}_o\right), \tag{3}$$

where $\mathbf{W}_{xi}, \mathbf{W}_{xf}, \mathbf{W}_{xo} \in \mathbb{R}^{k \times h}$ and $\mathbf{W}_{hi}, \mathbf{W}_{hf}, \mathbf{W}_{ho} \in \mathbb{R}^{h \times h}$ are the weight parameters, $\mathbf{b}_i, \mathbf{b}_f, \mathbf{b}_o \in \mathbb{R}^{1 \times h}$ are bias parameters, and σ is the sigmoid function.

1. The description of the LSTM block follows that of A. Zhang et al. (2023).

2. In the description of the LSTM block, the input $\mathbf{X}_t$ is defined as a $n \times k$ matrix, where n represents the number of observations in the batch. The matrix formulation illustrates how the LSTM block processes the information in practice, as the training of the multi-step LSTM, as is the case with deep learning networks, typically handle multiple sequences simultaneously for computational efficiency (Goodfellow, Bengio, and Courville 2016).

Figure 1: Information flow in an LSTM block.



Source: A. Zhang et al. (2023)

Updating the memory cell requires computing first an input node, $\tilde{\mathbf{C}}_t \in \mathbb{R}^{n \times h}$ using the equation below:

$$\tilde{\mathbf{C}}_t \left(\mathbf{X}_t, \mathbf{H}_{t-1}\right) = \sigma \left(\mathbf{X}_t \mathbf{W}_{xc} + \mathbf{H}_{t-1} \mathbf{W}_{hc} + \mathbf{b}_c\right), \tag{4}$$

where $\mathbf{W}_{xc} \in \mathbb{R}^{k \times h}$ and $\mathbf{W}_{hc} \in \mathbb{R}^{h \times h}$ are the weight parameters, and $\mathbf{b}_c \in \mathbb{R}^{1 \times h}$ are the bias parameters for the input node. Once the input node is computed, the memory cell and the hidden states are updated:

$$\mathbf{C}_t = \mathbf{F}_t \odot \mathbf{C}_{t-1} + \mathbf{I}_t \odot \tilde{\mathbf{C}}_t, \tag{5}$$

$$\mathbf{H}_t = \mathbf{O}_t \odot \tanh(\mathbf{C}_t). \tag{6}$$

The gated architecture of the LSTM fundamentally alters how past information is processed compared to a linear VAR model. In the VAR framework, fixed coefficients impose static weights on lagged observations, meaning the influence of past data remains constant over time. By contrast, the LSTM dynamically modulates the relevance of historical information through its input and forget gates. While the model's weight and bias parameters remain fixed after training, the sigmoid and $\tanh$ activation functions enable context-dependent gating: the input gate selectively incorporates new information, while the forget gate discards obsolete patterns. This adaptive mechanism allows the LSTM to emphasize or suppress past states based on the current input sequence, creating time-varying dependencies that VAR's rigid linear structure cannot capture.

3.2 The multi-step LSTM

Assume that the number of lags to include in the model is P, and that the goal is to predict the observation $\mathbf{X}_t$ using the past information contained in $\mathbf{X}_{t-i}, \quad i = 1, \ldots, P$. The construction of the multi-step LSTM requires chaining P LSTM blocks sequentially. The first block processes the input $\mathbf{X}_{t-P}$, as this is the earliest observation available, with default values for the the memory cell $\mathbf{C}_{t-P-1}$ and the hidden state $\mathbf{H}_{t-P-1}$ set to a default value, typically zero vectors **0** $\in \mathbb{R}^{n\times h}$. After the period $t-P-1$ information is processed, the LSTM block yields the updated memory cell $\mathbf{C}_{t-P}$ and hidden state $\mathbf{H}_{t-P}$, which then are fed into the second LSTM block, which process them using as input $\mathbf{X}_{t-P+1}$. Each successive LSTM block updates the memory cell and hidden state using the next input observation until the last LSTM block is reached. Afterwards, the hidden state serves as input into a linear prediction layer to yield the predicted value $\hat{\mathbf{X}}_t$:

$$\hat{\mathbf{X}}_t = \mathbf{W}_{\text{out}}\mathbf{H}_{\mathbf{t-1}} + \mathbf{b}_{\text{out}}, \tag{7}$$

where $\mathbf{W}_{\text{out}}$ are the weight parameters and $\mathbf{b}_{\text{out}}$ the bias parameters (Figure 2).

Figure 2: Pseudocode: multi-step LSTM with P lags

Input: Sequence of P input vectors $\mathbf{X}_{t-P}, \mathbf{X}_{t-P+1}, \ldots, \mathbf{X}_{t-1}$.
Output: Forecast $\hat{\mathbf{X}}_t$ of target $\mathbf{X}_t$.
Parameters:
    P: number of LSTM blocks (look-back period).
    $\mathbf{W}_{xi}, \mathbf{W}_{xf}, \mathbf{W}_{xo}, \mathbf{W}_{xc}, \mathbf{W}_{out}$: weight parameters matrices.
    $\mathbf{b}_i, \mathbf{b}_f, \mathbf{b}_o, \mathbf{b}_c, \mathbf{b}_{out}$: bias parameters vectors.
Initialize $\mathbf{C}_{t-P-1} \leftarrow \mathbf{0}$ ▷ Initial cell state (long-term memory)
Initialize $\mathbf{H}_{t-P-1} \leftarrow \mathbf{0}$ ▷ Initial hidden state (short-term memory)
for $j = P$ **downto** 1 **do** ▷ Process LSTM blocks sequentially
    Input $\leftarrow \mathbf{X}_{t-j}$ ▷ Current input at time $t-j$
    PrevCellState $\leftarrow \mathbf{C}_{t-j-1}$ ▷ Previous cell state
    PrevHiddenState $\leftarrow \mathbf{H}_{t-j-1}$ ▷ Previous hidden state
    $\mathbf{I}_{t-j} \leftarrow \mathbf{I}$ (Input, PrevHiddenState) ▷ Decide what input to retain, eq. (1)
    $\mathbf{F}_{t-j} \leftarrow \mathbf{F}$ (Input, PrevHiddenState) ▷ Decide what to forget, eq. (2)
    $\mathbf{O}_{t-j} \leftarrow \mathbf{O}$ (Input, PrevHiddenState) ▷ Decide what to output, eq. (3)
    $\tilde{\mathbf{C}}_{\mathbf{t-j}} \leftarrow \tilde{\mathbf{C}}_{\mathbf{t-j}}$ (Input, PrevHiddenState) ▷ Compute the input node, eq. (4)
    $\mathbf{C}_{t-j} \leftarrow \mathbf{F}_{t-j} \odot$ PrevCellState $+ \mathbf{I}_{t-j} \odot \tilde{\mathbf{C}}_{t-j}$ ▷ Update the memory cell, eq. (5)
    $\mathbf{H}_{\mathbf{t-j}} \leftarrow \mathbf{O}_{t-j} \odot \tanh(\mathbf{C}_{t-j})$ ▷ Update the hidden state, eq. (6)
end for
$\hat{\mathbf{X}}_t \leftarrow \mathbf{W}_{\text{out}} \cdot \mathbf{H}_{t-1} + \mathbf{b}_{\text{out}}$ ▷ Produce prediction via prediction layer, eq. (7)
Return $\hat{\mathbf{X}}_t$

Sources: the authors.

3.3 The LSTM multiplier response (LMR)

Once the multi-step LSTM has been properly trained, it can be represented concisely by the function $\mathbf{f}_{\text{LSTM}}$, which takes as input the P lags of the $\mathbf{x}_t \in \mathbb{R}^{1\times k}$ to predict its value, $\hat{\mathbf{x}}_t$:

$$\hat{\mathbf{x}}_t = \mathbf{f}\left(\mathbf{x}_{[t-P:t-1]}\right), \tag{8}$$

where $\mathbf{x}_{[t-P:t-1]} \in \mathbb{R}^{P\times k}$ is defined as:

$$\mathbf{x}_{[t-P:t-1]} = \begin{bmatrix} \mathbf{x}_{t-P} \\ \mathbf{x}_{t-P+1} \\ \vdots \\ \mathbf{x}_{t-2} \\ \mathbf{x}_{t-1} \end{bmatrix} = \begin{bmatrix} x^1_{t-P} & x^2_{t-P} & \cdots & x^k_{t-P} \\ x^1_{t-P+1} & x^2_{t-p+1} & \cdots & x^k_{t-p+1} \\ \vdots & \vdots & \cdots & \vdots \\ x^1_{t-2} & x^2_{t-2} & \cdots & x^k_{t-2} \\ x^1_{t-1} & x^2_{t-1} & \cdots & x^k_{t-1} \end{bmatrix}.$$

Starting at period t, equation (8) serves to generate the N-step ahead forecasts recursively. Assuming $N > P$, the recursive forecasts are computed in a closed loop, using the LSTM's own predictions:

$$\begin{aligned} \hat{\mathbf{x}}_{t+1} &= \mathbf{f}\left([\mathbf{x}_{t-P}, \ldots; \mathbf{x}_t]^T\right), \\ \hat{\mathbf{x}}_{t+2} &= \mathbf{f}\left([\mathbf{x}_{t-P+1}, \ldots; \hat{\mathbf{x}}_{t+1}]^T\right), \\ &\vdots \qquad\qquad \vdots \\ \hat{\mathbf{x}}_{t+N} &= \mathbf{f}\left([\hat{\mathbf{x}}_{t-P+N}, \ldots; \hat{\mathbf{x}}_{t-2+N}, \hat{\mathbf{x}}_{t-1+N}]^T\right). \end{aligned} \tag{9}$$

Notice that in each step, the LSTM model will generate not only the next period forecast of the system variables, $\mathbf{x}_{t+j},\ j = 1, \ldots, N$, but also update the value of the memory cell and the hidden state. Since these updated values are not necessarily consistent across successive forecasts, the recursive forecasts always reset the memory cell and the hidden state in the $t+j-P,\ j = 1, \ldots, N$ periods to the zero vector.

The set of equations (9) serve as the basis for defining the LSTM multiplier. First, following the line of reasoning pursued when defining the impulse response functions in VAR systems, set the expected value of the system variables equal to their LSTM forecast:

$$\mathbb{E}[\hat{\mathbf{x}}_{t+s}] = \mathbf{f}(\mathbf{z}_{[t+s-P:t+s-1]}), \tag{10}$$

where

$$\mathbf{z}_{[t+s-p:t+s-1]} = \begin{cases} \hat{\mathbf{x}}_{[t+s-P:t+s-1]}, & \text{if } s > P \\ [\mathbf{x}_{[t+s-P:t]}; \hat{\mathbf{x}}_{[t+1:t+s-1]}]^T, & \text{otherwise,} \end{cases} \tag{11}$$

and let $\mathbf{d}_t \in \mathbb{R}^{1\times k}$ be a vector representing the shock applied to the system in period t. In the case of an additive shock, the shocked values of the system variables in period t are given by $\mathbf{x}_t^d = \mathbf{x}_t + \mathbf{d}_t$; and in the case of a multiplicative shock, $\mathbf{x}_t^d = \mathbf{x}_t \odot \mathbf{d}_t$. Applying (9) generates the forecasts $\mathbf{x}_{t+j}^d,\ j = 1, \ldots, N$ under the shock conditions, and equations (10) and (11) serve to find the conditional expected value $\mathbb{E}[\mathbf{x}_{t+j}^d]$ once $\mathbf{x}_t^d$ is replaced for $\mathbf{x}_t$.

Given the shock $\mathbf{d}_t$, the LSTM multiplier s periods ahead, $\mathbf{LSTMm}\,(\mathbf{x}_t \mid \mathbf{d}_t)$ is defined as:

$$\mathbf{LSTMm}\,(t+s \mid \mathbf{d}_t) = \mathbb{E}[\hat{\mathbf{x}}_{t+s}^d] - \mathbb{E}[\hat{\mathbf{x}}_{t+s}], \tag{12}$$

and by letting s taking values in $[1, N]$, where N is the last period desired in the analysis, we generate the LSTM equivalent of a VAR impulse response analysis, or the LSTM multiplier response (LMR). Note that the LMR response reflects the LSTM internal representation of the variables interdependencies.

Contrary to the impulse response in the standard linear VAR, which depends only on its coefficients that are constant over time, the LMR depends on the past history of the system over the last P periods, including the current one when the shock occurs, i.e. $\mathbf{x}_{[t-P:t]}$. This dependence on historical context enables the LSTM to recognize that, given the same shock, the impact on the system may differ depending on the prevailing economic regime or recent trajectory—such as whether the economy is in a recession or expansion. As a result, the shape of the LMR reflects these regime-dependent variations, capturing the notion that a negative shock might be more damaging during a recession than in a period of growth.

Another significant advantage of the LMR over the VAR-based impulse response is that no structural assumptions about the variable ordering, e.g. causal relationships or error term correlation, are needed. In the computation of the LMR the shock is applied directly to the variable of interest rather than to the error term, as is the case in the VAR. Hence, as opposite to a structural VAR, it is unnecessary to orthogonalize the shocks in the LSTM. The absence of structural identification combined with dynamics conditional on the lagged and current values of the system variables suggest that the LMR is similar to performing scenario analysis. The advantage is gained, however, at the expense of ignoring economic causality.

3.4 LSTM training: hyperparameters and parameter selection method

As other deep learning models and machine learning models more generally the behavior and performance of LSTMs is determined partly by hyperparameters, whose values are configured before the model is trained on the data. Unlike model parameters such as the weights and biases, the hyperparameter values are not learned by the model itself. Some of these hyperparameters include the number of lags (time steps), the dimension of the hidden state (number of neurons), the number of layers in the LSTM, and the dropout rate. This last hyperparameter forces the LSTM to ignore the information provided by a subset of the neurons in the hidden states, which helps to prevent the model to overfit the data. Other important hyperparameters are the number of epochs, or number of training iterations over the data sample, which if inadequate, could cause the model to underfit or overfit the data; and the learning rate, which if too high or too low, might not allow the algorithm to optimize the weight and bias parameters.

Selecting the best possible hyperparameter values is known as hyperparameter tuning. The tuning process aims to find the combination of hyperparameters values that performs the best according to some metric, such as minimizing a target loss function based on the model's one-step ahead predictions. The loss evaluation is based on a five-fold time series cross validation to allow for the presence of serial correlation in the time series data. The cross-validation requires the dataset to be divided first into a training dataset and a testing dataset. The training dataset is partitioned again into five sequential folds. In the first cross-validation round, the first fold is used as the initial training set and the second as the validation set; in the second round, the first two folds are combined to form the training set and the third fold is used as the validation set; and so on. During the cross-validation exercise, it is assumed that the blocks of P observations are i.i.d. and that predictions one-step ahead depend only on the P data block.

The loss function is set equal to the mean squared error (MSE), the average squared deviation between the LSTM model's one-step ahead predictions and the observed values in the validation set. The hyperparameter tuning is performed using Bayesian Optimization (BO) (Snoek, Larochelle, and Adams 2012). Unlike grid search or random search, which explore the hyperparameter space in a less informed manner, BO leverages a probabilistic model to guide its search. The core idea is to build a "surrogate model" that approximates the true loss function based on past evaluations of different hyperparameter combinations. This surrogate model provides both a prediction of the loss function's value and an estimate of the uncertainty around that prediction. An "acquisition function" then uses this information to determine the next set of hyperparameters to evaluate, balancing exploration (trying new, uncertain regions of the hyperparameter space) and exploitation (focusing on areas predicted to yield high performance). By selecting promising hyperparameter combinations, BO is able to find the optimal settings with significantly fewer evaluations compared to other methods.

4 LSTM empirical implementation

4.1 Data

Two different datasets, both comprising U.S. variables, serve to estimate the LSTM and VAR models. The selection of variables is guided by Sims (1980), for the first dataset, and Sims (1992), for the second dataset, which are two seminal papers that established the VAR methodology as a cornerstone of macroeconomic analysis. The final choice of variables differ slightly from those in the references due to data availability issues. The two datasets differ in the sampling frequency since other studies, some reviewed in Section 2, have found that for some datasets LSTMs might not necessarily perform better than other time series models across all sampling frequencies. All the data are sourced from Haver Analytics.

The first dataset is sampled at a quarterly frequency and covers the period 1985Q1 - 2024 Q4. The variables included are the U.S. money supply, as measured by $M2$, the unemployment rate (U), the wage rate (W), and the import price index (IMP), the gross domestic product ($RGDP$), and the personal consumption expenditure price index (PCE). The second dataset, sampled monthly, includes the effective fed funds rate (FF), money supply ($M2$), the personal consumption expenditure price index (PCE), industrial production (IP), the dollar index (DXY), and the CRB Spot Commodity Price Index, Raw Industrials ($PZRAW$). The sample period is January 1985 - January 2025. In both datasets the variables are transformed to year-on-year growth rates except for the federal funds rate (FF), which is included in levels. All series are subsequently normalized using Z-scores to ensure comparability across variables with different scales.

4.2 Bayesian optimization of the LSTM: search space and hyperparameter tuning

The range of hyperparameter values the BO uses to minimize the MSE of the LSTM one-step ahead predictions are:

- **Hidden size, H, (number of neurons in LSTM):** $H \in [12, 600],\ H \in \mathbb{Z}$.
- **Number of LSTM layers, N:** $N \in [1, 7],\ N \in \mathbb{Z}$.
- **Dropout rate between LSTM layers, d:** $d \in [0, 0.7],\ d \in \mathbb{R}$.
- **Learning rate, η:** $\eta \in [0.001, 0.1],\ \eta \in \mathbb{R}$.
- **Number of epochs, E:** $E \in [300, 1000],\ E \in \mathbb{Z}$.

The values of two other hyperparameters are set outside the BO procedure. The first hyperparameter is the **sequence length**, P, which is also the number of lags in the LSTM was determined by fitting a VAR to the data and selecting the lag order that minimized the Akaike Information Criterion (AIC). The values are set equal to 8 quarters for the first dataset and 14 months for the second dataset. The second hyperparameter is the **batch size**, or the number of examples used during the training process of the model.[3]. Due to computational limitation, we set the batch size to 64.

The partitioning of the data into a training subsample, used for training and validation, and the testing subsample, which serves to evaluate the out-of-sample performance of the model, reduces potential information leakage between the two subsamples. The hyperparameter tuning is performed

3. Please refer to the discussion in the LSTM block section, Section 3.1 For further details, see Goodfellow, Bengio, and Courville (2016))

using data points observed until December 2015. Observations beyond this date are used to test the out-of-sample performance of the LSTM against the VAR model recursively based on one-step ahead predictions in the pre-Covid and COVID-19 and post-Covid samples, as well as for the full sample period.

5 Results

5.1 Hyperparameter tuning

Figure 3 presents the BO workflow for hyperparameter tuning of the LSTM models fitted to the quarterly and monthly datasets. The process began with 50 random evaluations to initialize the Gaussian Process surrogate model, followed by 50 optimization rounds guided by expected improvement (Snoek, Larochelle, and Adams 2012). As shown, the algorithm rapidly identified high-performance regions after the exploratory phase, with subsequent evaluations concentrating near optimal configurations. Table 1 shows the final set of hyperparameter values.

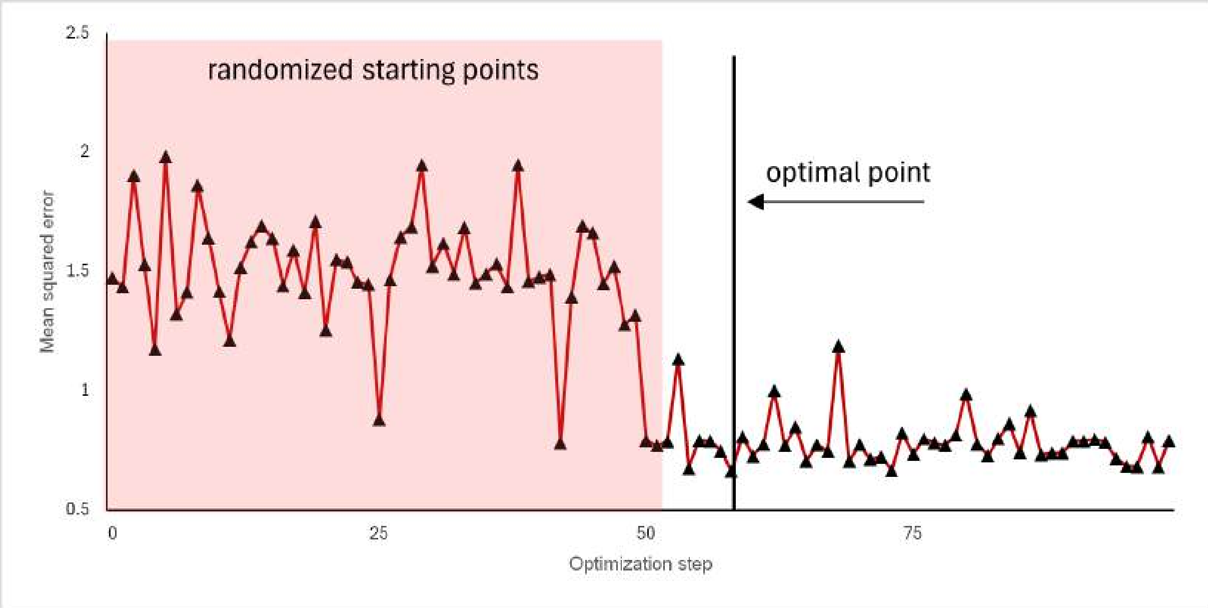
Table 1: LSTM models, hyperparameter values

Hyperparameter	Quarterly model	Monthly model
Sequence length, L	8	14
Number of layers, N	1	1
Hidden cell size, H	84	57
Training epochs, E	800	300
Learning rate, η	0.0032	0.0354

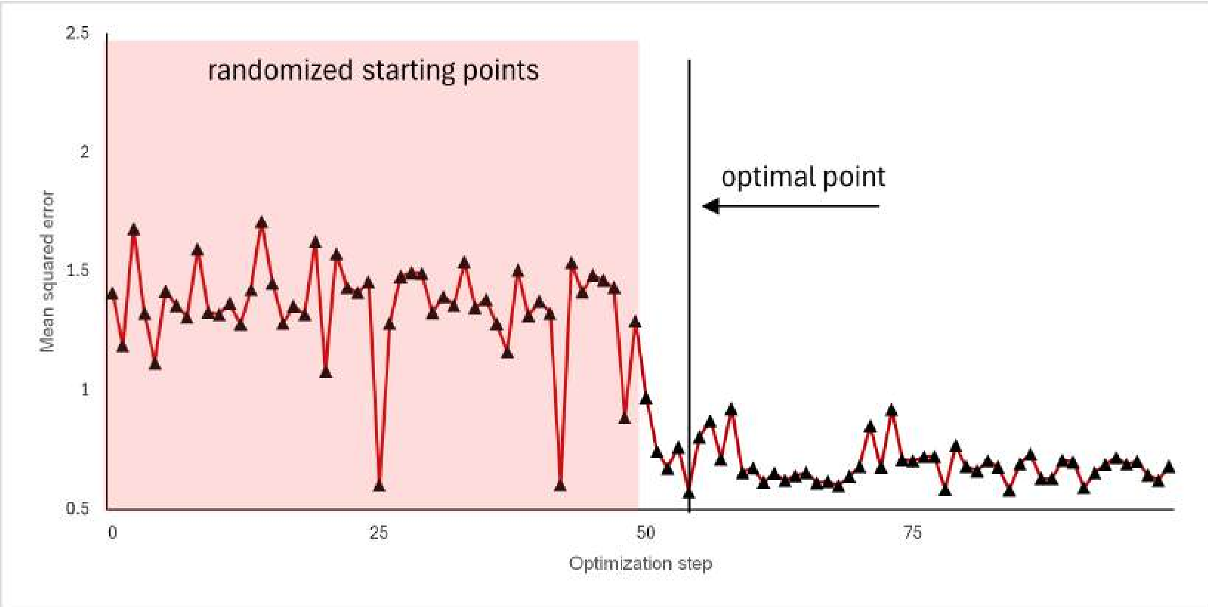
Notes: sequence length determined outside BO workflow and based on AIC in reduced VAR models; dropout rate is irrelevant in one layer LSTMs. Sources: the authors.

Figure 3: Bayesian optimization, mean squared error (MSE)

(a) Quarterly LSTM model



(b) Monthly LSTM model



Sources: the authors.

5.2 Out-of-sample performance vis-à-vis reduced VAR models

Once the optimal hyperparameter values are determined using BO, one-step ahead predictions are used to evaluate the out-of-sample performance of the quarterly and monthly LSTM models, and compared them with those of standard reduced form VARs fitted to the same datasets. The results are summarised in Table 2, which shows the root mean squared errors (RMSE) of the four models over three different out-of-sample periods. The first one is the pre-COVID period from 2015 until 2019; the second period is the COVID and post-COVID period, from 2020 until 2025; and the third one is the full sample period from 2015 until 2024 (quarterly data) or 2025 (monthly data). Figures 4 and 5 illustrate these results.

There are significant differences in the forecasting performance of the LSTMs and the VAR models across different time periods and sampling frequencies. At the quarterly frequency, the LSTM consistently outperformed the VAR during the pre-COVID period, with the VAR's RMSE values ranging from 50 to 150 percent higher than the LSTM's RMSEs for almost all variables. The exception is the wage rate, W, for which the VAR performs better. Crucially, during the highly volatile COVID and post-COVID period (2020–24), the LSTM's advantage increases dramatically—its ability to capture nonlinear patterns proves far superior to the VAR's linear formulation, a finding in line with those of Cao, Li, and Li (2019). For example, in the case of the unemployment rate, U, the VAR's RMSE is 2.5 times higher than the LSTM's RMSE. This suggests that LSTMs could excel during periods in which structural shocks induce regime changes as the LSTM gate structure might be able to capture the changes.

Based on the quarterly models' results, it is expected that the monthly LSTM would outperform the corresponding VAR, but this is not the case. At the monthly frequency, the LSTM and VAR models exhibit similar predictive performance with comparably low RMSE values. The monthly LSTM, however, consistently underperforms the VAR across all variables and time periods but only by a small margin (Figure 5 and Table 2). Arguably, at higher frequencies, non-linear dependencies can be well approximated by linear approximations, which are more robust to the presence of noise. Further refinements to the LSTM architecture—such as increasing the number of input time-steps or incorporating additional consolidation layers— might potentially enhance forecasting accuracy by minimizing the noise effects. Consistent patterns are observed when using lesser outlier-sensitive metrics such as the Mean Absolute Error (MAE), as reported in Table 3.

We would like to note that these results are consistent with findings from other studies, which show that LSTMs sometimes outperform other time series models, but sometimes do not (Makridakis, Spiliotis, and Assimakopoulos 2018). This underscores the importance of trying different models and model specifications (Hewamalage, Bergmeir, and Bandara 2021).

Table 2: LSTMs and VARs: Out-of-sample forecast performance, RMSE

(a) Quarterly models

	Variable	LSTM	VAR
Pre-COVID 2015–19	IMP	0.345	0.460
	M2	0.150	0.224
	PCE	0.224	0.316
	RGDP	0.214	0.338
	U	0.092	0.155
	W	0.224	0.201
COVID and post-COVID 2020–24	IMP	0.840	4.140
	M2	1.290	2.210
	PCE	0.576	2.960
	RGDP	1.840	2.710
	U	2.140	5.370
	W	0.581	1.130
Full test sample 2015–24	IMP	0.667	3.100
	M2	0.968	1.650
	PCE	0.455	2.210
	RGDP	1.380	2.030
	U	1.590	4.010
	W	0.458	0.856

(b) Monthly models

	Variable	LSTM	VAR
Pre-COVID 2015–19	DXY	0.288	0.229
	FF	0.093	0.035
	IP	0.218	0.182
	M2	0.102	0.093
	PCE	0.133	0.138
	PZRAW	0.225	0.166
COVID and post-COVID 2020–25	DXY	0.405	0.453
	FF	0.143	0.083
	IP	0.778	0.797
	M2	0.537	0.357
	PCE	0.318	0.257
	PZRAW	0.410	0.263
Full test sample 2015–25	DXY	0.358	0.371
	FF	0.123	0.067
	IP	0.600	0.608
	M2	0.407	0.274
	PCE	0.254	0.213
	PZRAW	0.341	0.225

Notes: All models estimated using pre-2015 data; out-of-sample performance based on one-step ahead forecasts. Reported results are based on rescaled targets, where all dependent variables were normalised using Z-scores prior to model estimation to improve comparability across series and metrics. Notation: IMP, U.S. import price index; $M2$, U.S. money supply; PCE, U.S. personal consumption expenditure index; $RGDP$, U.S. real GDP; U, U.S. unemployment rate; W, U.S. wage rate; DXY, U.S. dollar index; FF, U.S. federal funds rate; IP, U.S. industrial production index; $PZRAW$, CRB spot commodity price index, raw industrials. Sources: the authors.

Table 3: LSTMs and VARs: Out-of-sample forecast performance, MAE

(a) Quarterly models

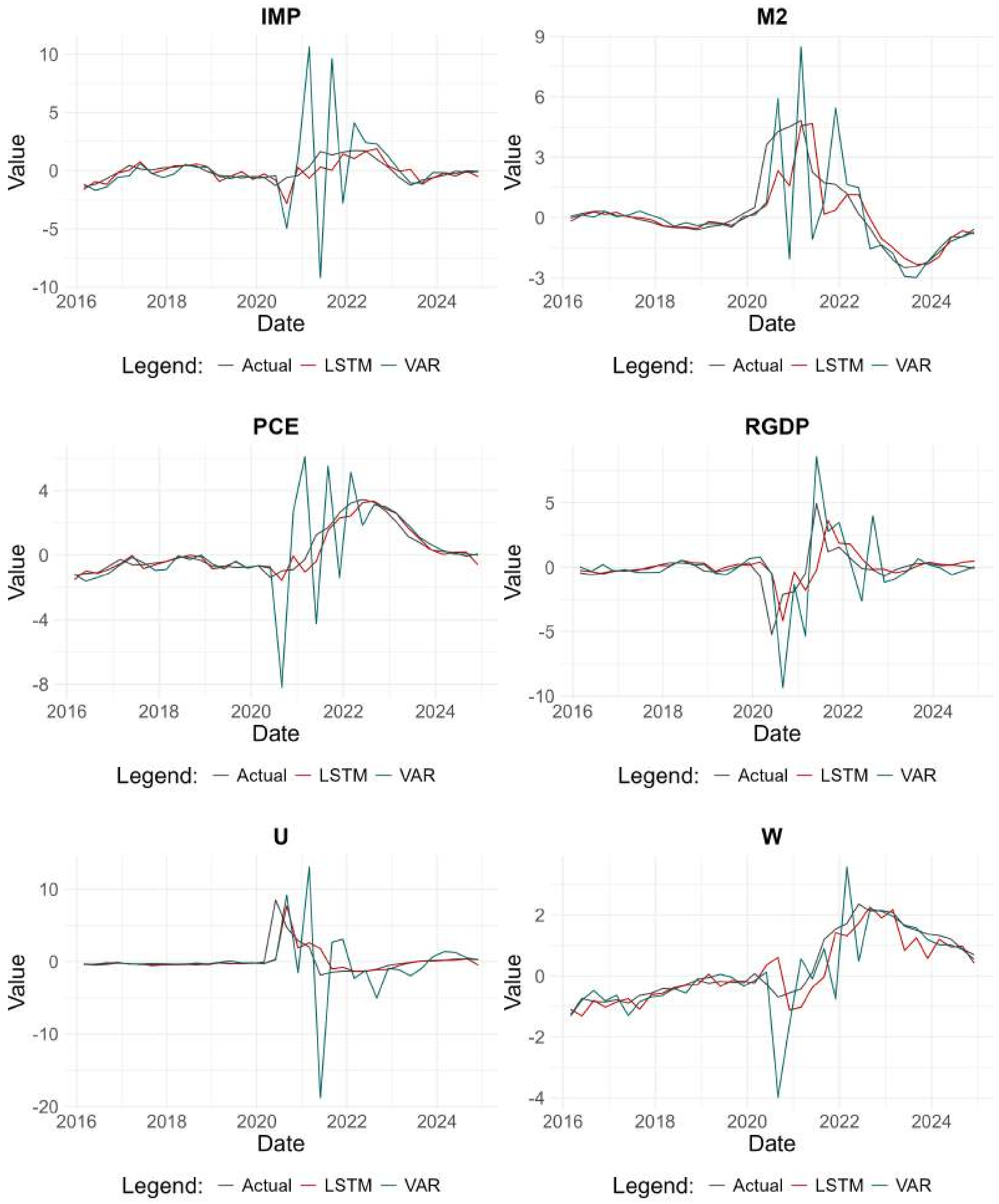
	Variable	LSTM	VAR
Pre-COVID 2015–19	IMP	0.287	0.384
	M2	0.123	0.185
	PCE	0.177	0.276
	RGDP	0.170	0.284
	U	0.081	0.115
	W	0.165	0.165
COVID and post-COVID 2020–24	IMP	0.626	2.400
	M2	0.882	1.400
	PCE	0.427	1.870
	RGDP	1.150	1.830
	U	0.978	3.290
	W	0.467	0.682
Full test sample 2015–24	IMP	0.475	1.500
	M2	0.544	0.858
	PCE	0.316	1.160
	RGDP	0.715	1.140
	U	0.579	1.880
	W	0.333	0.452

(b) Monthly models

	Variable	LSTM	VAR
Pre-COVID 2015–19	DXY	0.223	0.180
	FF	0.069	0.028
	IP	0.172	0.137
	M2	0.086	0.074
	PCE	0.102	0.098
	PZRAW	0.177	0.129
COVID and post-COVID 2020–25	DXY	0.312	0.317
	FF	0.108	0.056
	IP	0.451	0.479
	M2	0.314	0.226
	PCE	0.247	0.202
	PZRAW	0.328	0.207
Full test sample 2015–25	DXY	0.273	0.257
	FF	0.091	0.044
	IP	0.328	0.329
	M2	0.214	0.159
	PCE	0.183	0.156
	PZRAW	0.261	0.173

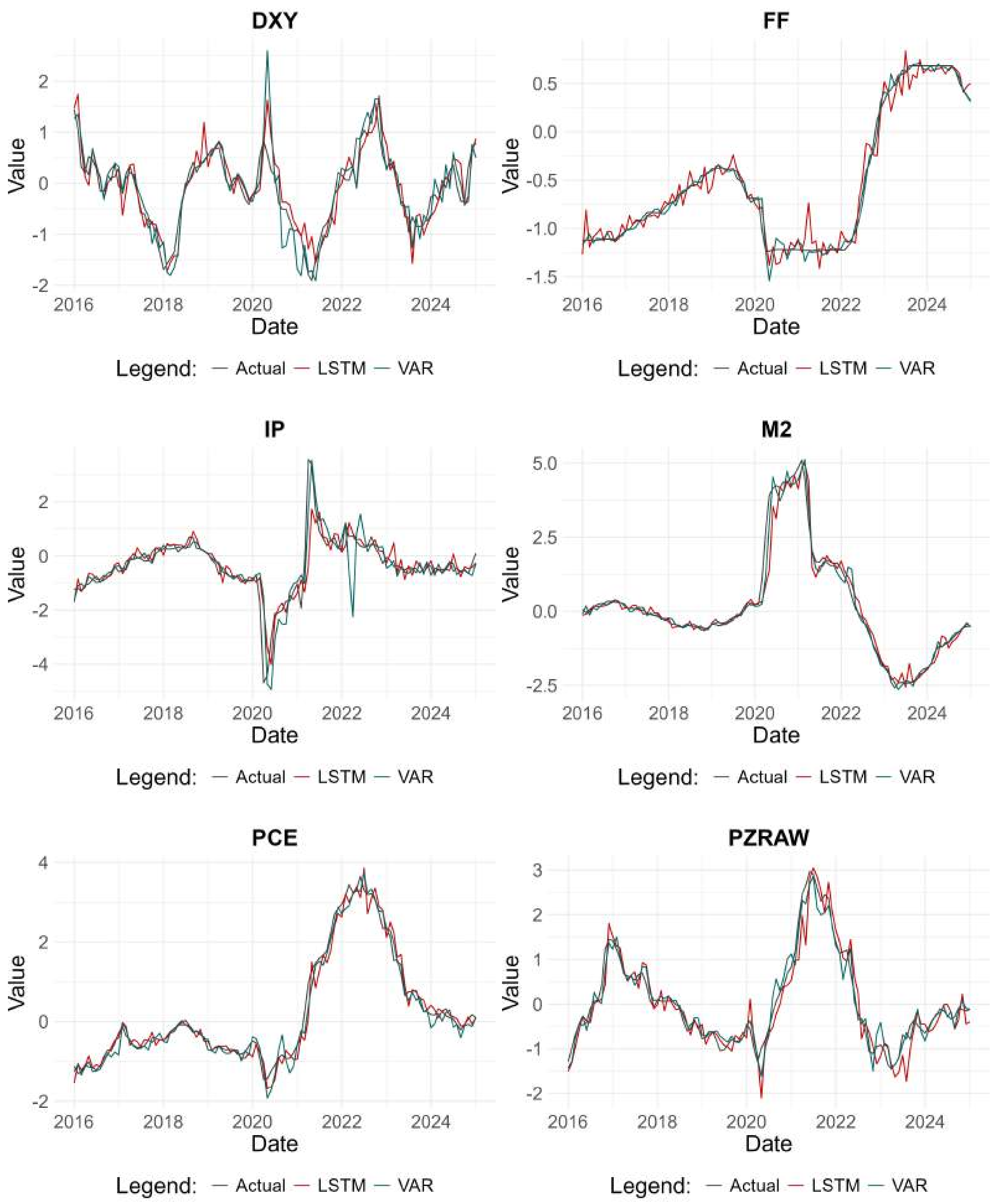
Notes: All models estimated using pre-2015 data; out-of-sample performance based on one-step ahead forecasts. Notation: IMP, U.S. import price index; $M2$, U.S. money supply; PCE, U.S. personal consumption expenditure index; $RGDP$, U.S. real GDP; U, U.S. unemployment rate; W, U.S. wage rate; DXY, U.S. dollar index; FF, U.S. federal funds rate; IP, U.S. industrial production index; $PZRAW$, CRB spot commodity price index, raw industrials. Sources: the authors.

Figure 4: Out-of-sample forecasts: quarterly LSTM and VAR models



Notes: All models estimated using pre-2015 data; out-of-sample performance based on one-step ahead forecasts. Notation: IMP, U.S. import price index; $M2$, U.S. money supply; PCE, U.S. personal consumption expenditure index; $RGDP$, U.S. real GDP; U, U.S. unemployment rate; W, U.S. wage rate. Sources: the authors.

Figure 5: Out-of-sample forecasts: monthly LSTM and VAR models



Notes: All models estimated using pre-2015 data; out-of-sample performance based on one-step ahead forecasts. Notation: DXY, U.S. dollar index; FF, U.S. federal funds rate; IP, U.S. industrial production index; $M2$, U.S. money supply; PCE, U.S. personal consumption expenditure index; $PZRAW$, CRB spot commodity price index, raw industrials. Sources: Haver Analytics and the authors.

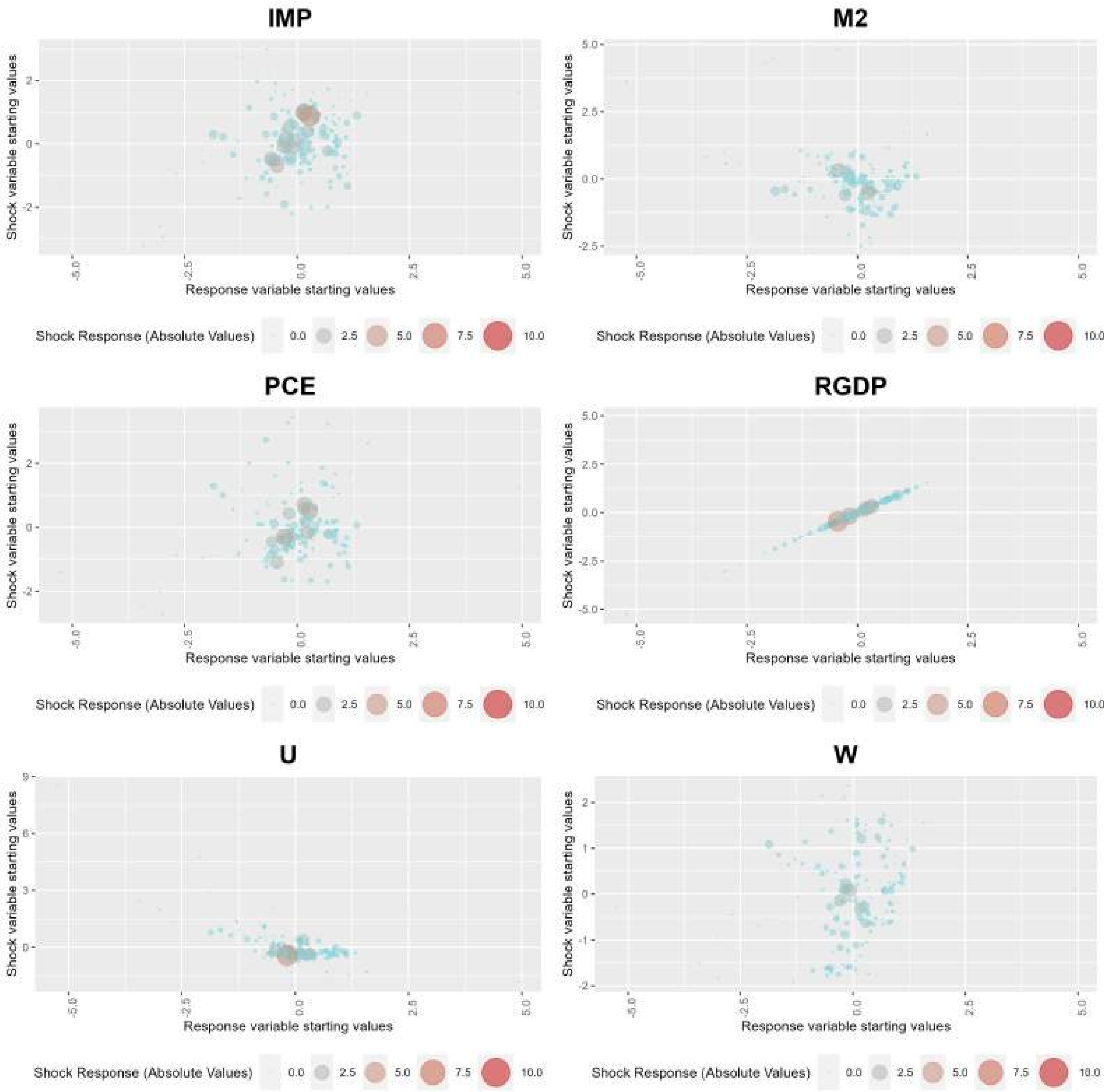
5.3 LSTM multiplier response

To examine the LMR for a single variable, x_i, we apply an additive shock $\mathbf{d}_t^i = [0, \ldots, \sigma_i, \ldots, 0]$, where σ_i is the standard deviation of x_i. Afterwards, we compute the multipliers recursively using equation 12 over a 12-period horizon (12 quarters and 12 months for the quarterly and monthly models respectively). For any given horizon, the values of the multipliers depend on the current and lagged values of the variables at the time of the shock. We illustrate this situation by analyzing the specific case of real GDP shocks in the quarterly model. Figure 6 plots the values of the GDP LSTM multiplier 12 quarters, calculated using historical data and the prevailing values of GDP and the other variables at the time of the shock. Compared with a standard VAR, the variation displayed by the multipliers indicates that the shock transmission mechanism is highly dependent on the initial conditions. Hence, VAR-based policy guidance could underestimate or overestimate policy responses. This is also the case for other variables' multipliers, as the figures in the Appendix show.

Accordingly, the shape of the LRM will also depend on the prevailing state of the system at the time the shock occurs. Instead of analyzing each individual response across the entire data sample we focus on the median response and its interquartile range to capture the typical behavior and variability of the response. We examine the GDP LMR in the quarterly model, shown in Figure 7. From the median LMRs, the impact of the GDP shock on the real economy fades by the end of the sixth quarter. Several LMRs align with economic intuition: import prices (IMP) rise as higher GDP drives import demand; higher income leads to increased personal consumption expenditure (PCE); economic activity accelerates, with growth momentum gradually declining ($RGDP$); and businesses respond to increased demand for goods and services by hiring more workers, lowering the unemployment rate (U).

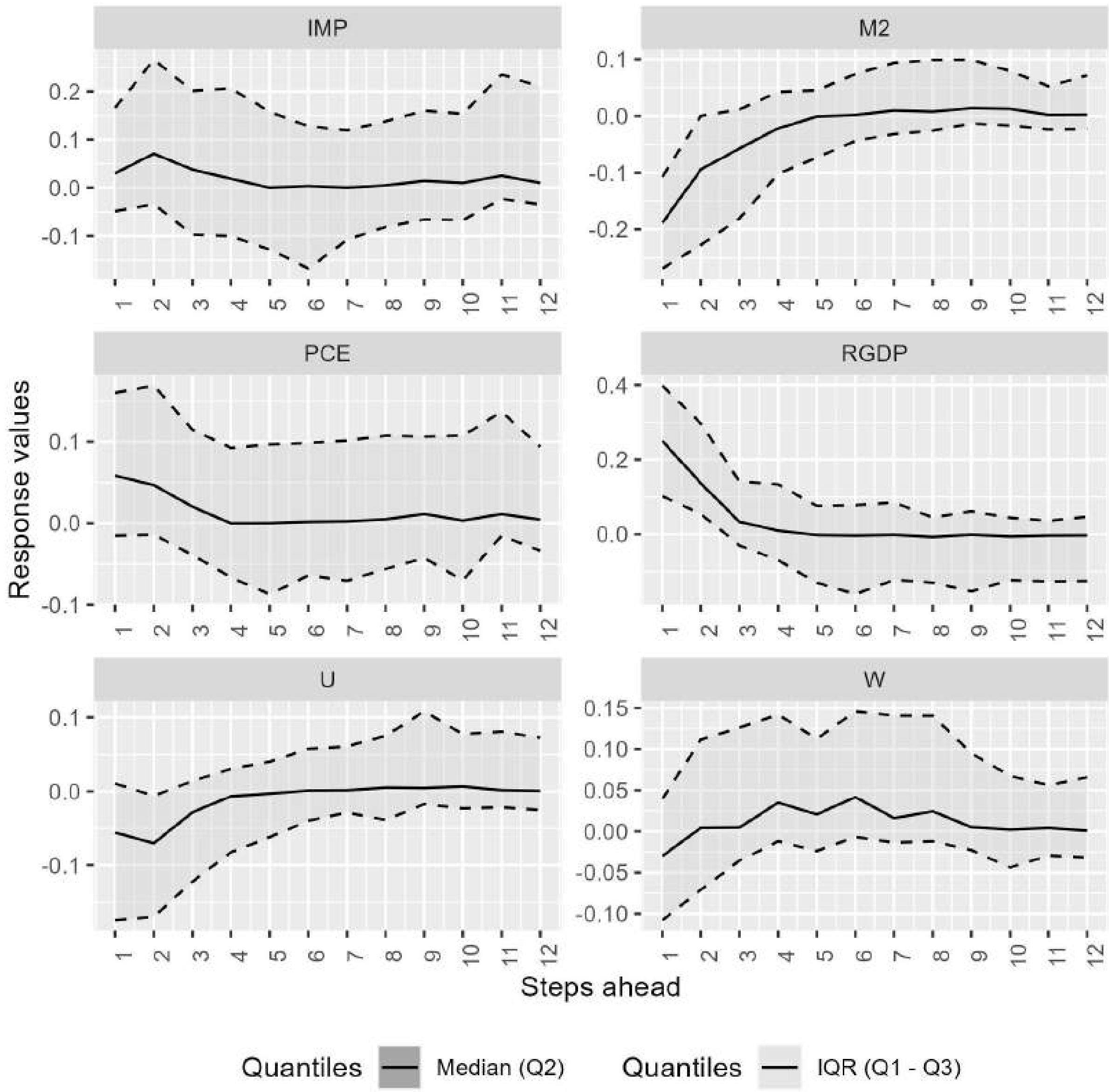
The patterns observed in two median LMRs are seemingly counterintuitive. The first concerns the M2 LMR, which indicates a decline in the money supply following a GDP shock. This stands in stark contrast to a VAR impulse response, which typically shows an increase in M2 under similar circumstances. At least two potential explanations exist for this unexpected response pattern. First, a positive GDP shock often signals economic expansion, leading to increased aggregate demand and potential inflationary pressures. In response, central banks may implement tighter monetary policies to prevent economic overheating. Second, economic agents might anticipate inflation following a positive GDP shock, leading them to hoard money and reduce its velocity. To counteract this behavior and maintain nominal GDP stability, a central bank could opt to reduce the money supply.

Figure 6: LSTM multiplier: real GDP shock ($RGDP$), 12 quarters ahead, quarterly model



Notes: The figure shows the impact of a 1 standard deviation shock to real GDP on other variables 12 quarters ahead, as measured by the corresponding LSTM multiplier (equation 12). Both axes show the Z-score values of the variables. Notation: IMP, U.S. import price index; $M2$, U.S. money supply; PCE, U.S. personal consumption expenditure index; $RGDP$, U.S. real GDP; U, U.S. unemployment rate; W, U.S. wage rate. Source: Haver Analytics and the authors.

Figure 7: LSTM response (LMR): 1-standard deviation shock to real GDP ($RGDP$)



Note: Response values measured as Z-scores. Notation:IMP, U.S. import price index; $M2$, U.S. money supply; PCE, U.S. personal consumption expenditure index; $RGDP$, U.S. real GDP; U, U.S. unemployment rate; W, U.S. wage rate. Sources: Haver Analytics and the authors.

The second observation concerns the wage dynamics in the LMR. Contrary to common intuition and the VAR predictions, which indicate that wages increase following a GDP shock, the LMR reveals a different pattern: wages initially decline before subsequently rising. This unexpected behaviour may be attributed to an increase in labour supply, as heightened economic activity encourages more individuals to enter the workforce, potentially causing a temporary downward pressure on wages. Simultaneously, the expected decline in the money supply, as discussed earlier, could constrain firms' capacity to raise wages. Over time, as the economy adjusts and both labour demand and money supply catch up with the expanded workforce, wages begin to trend upward. These nuanced patterns captured by the LMR for both money supply and wage dynamics offer a more complex view of economic responses to shocks than traditional models. We hypothesise that these scenarios, observed in the historical sample, may occur more frequently than previously thought, underscoring the complexity of monetary and labour market dynamics and the potential limitations of traditional economic models in capturing such responses.[4]

6 Conclusions

Through a detailed examination of the LSTM architecture, encompassing the LSTM block and multi-step framework, this study establishes a robust process for implementing multivariate time series LSTMs and analyzing shock propagation. Key to this process is rigorous hyperparameter tuning via Bayesian optimization, ensuring optimal model performance. Empirical results derived from U.S. macroeconomic data indicate that quarterly LSTMs can exhibit superior forecasting accuracy compared to VAR models, particularly during periods of heightened volatility like the COVID-19 pandemic.

This paper also introduces the LSTM multiplier response function as a novel approach to analyzing shock propagation within multivariate economic systems. The analysis demonstrates that LSTM networks offer key advantages over traditional linear VAR models, notably in their capacity to capture nonlinear dynamics while accounting for the state of the system at the time of the shock realization. The methodology involves applying shocks directly to variables of interest, removing the need for establishing causality or orthogonalizing the system as required by VAR approaches.

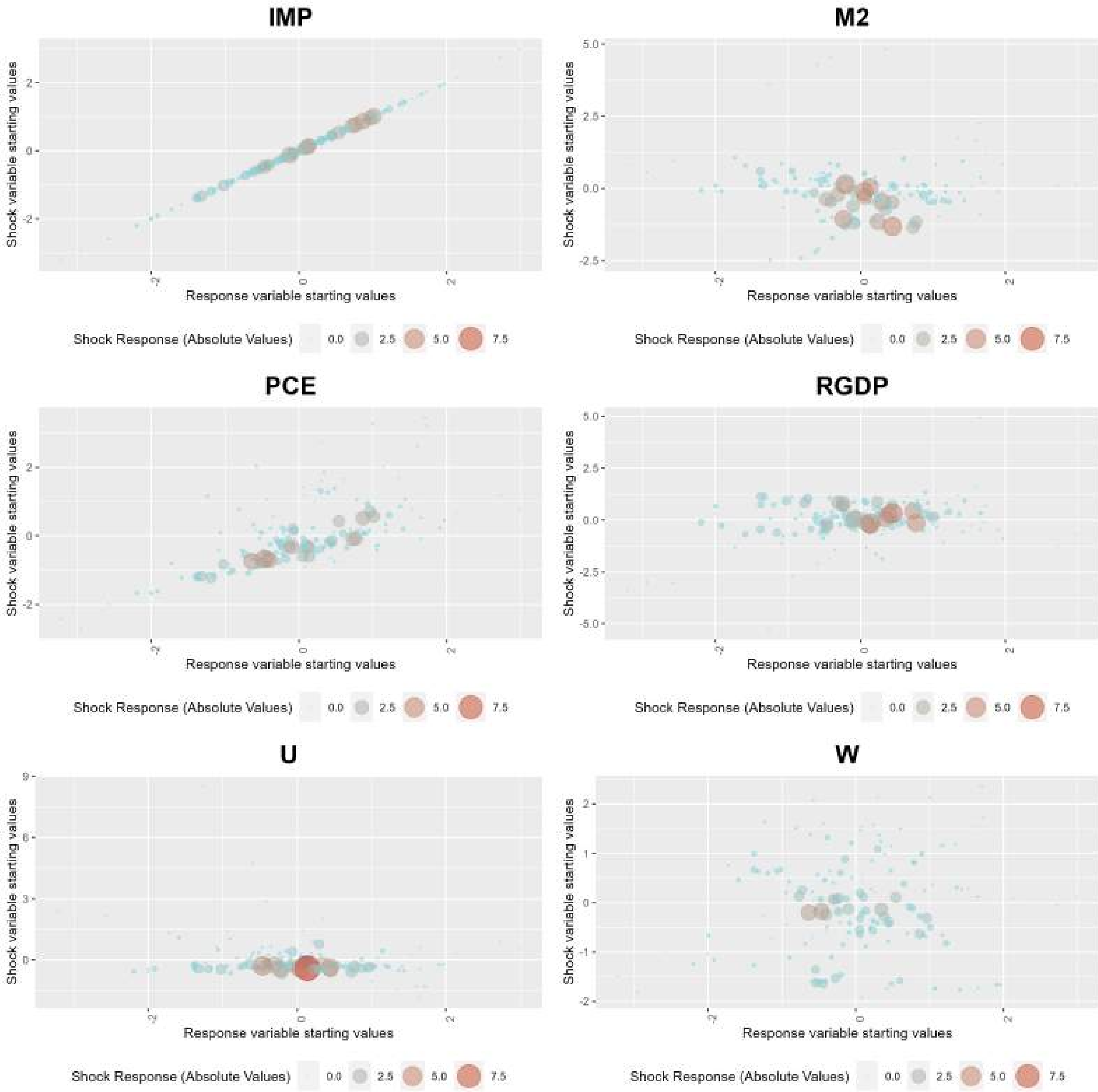
The LSTM multiplier response function, designed to parallel VAR impulse responses, displays similar qualitative characteristics and could potentially capture the shock propagation dynamics with greater precision due to LSTM's enhanced forecasting capabilities. Our analysis, however, suggests that the LSTM advantage erodes with higher frequency data, as linear models may handle noise more effectively. These findings underscore the potential of LSTM networks as a tool for understanding economic shock transmission, while also highlighting their limitations. Future research

4. The appendix presents additional figures illustrating the LSTMm and LMR for the variables in the quarterly model since it outperforms its VAR counterpart. Results for the monthly model are available upon request.

may benefit from exploring hybrid models or frequency-specific LSTM architectures, as suggested by Sezer et al. (2020).

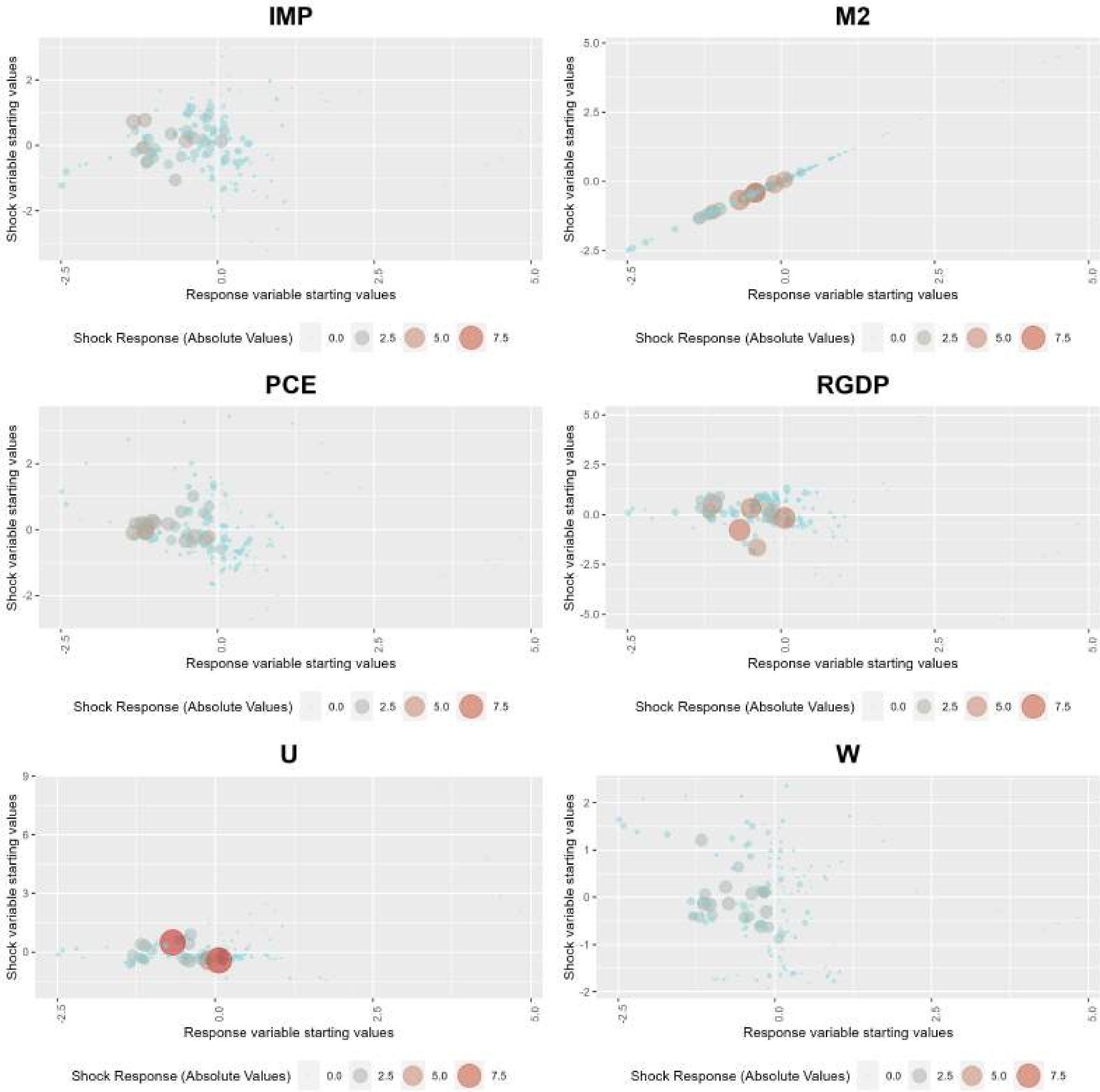
Appendix - additional figures

Figure A.1: LSTM multiplier: import price shock (IMP), 12 quarters ahead, quarterly model



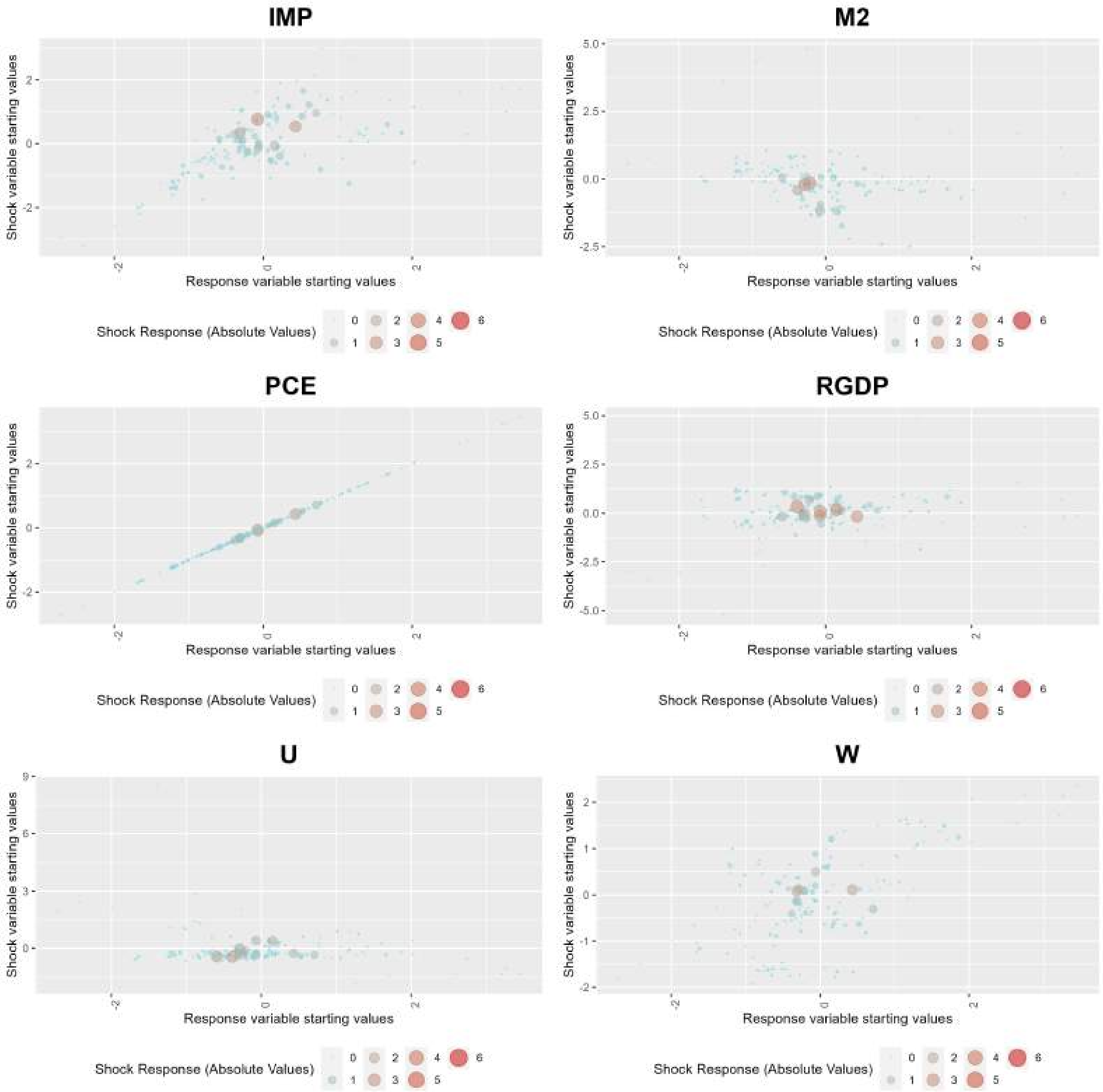
Notes: The figure shows the impact of a 1 standard deviation shock to import prices on other variables 12 quarters ahead, as measured by the corresponding LSTM multiplier (equation 12). Both axes show the Z-score values of the variables. Notation: IMP, U.S. import price index; $M2$, U.S. money supply; PCE, U.S. personal consumption expenditure index; $RGDP$, U.S. real GDP; U, U.S. unemployment rate; W, U.S. wage rate. Sources: Haver Analytics and the authors.

Figure A.2: LSTM multiplier: money supply shock ($M2$), 12 quarters ahead, quarterly model



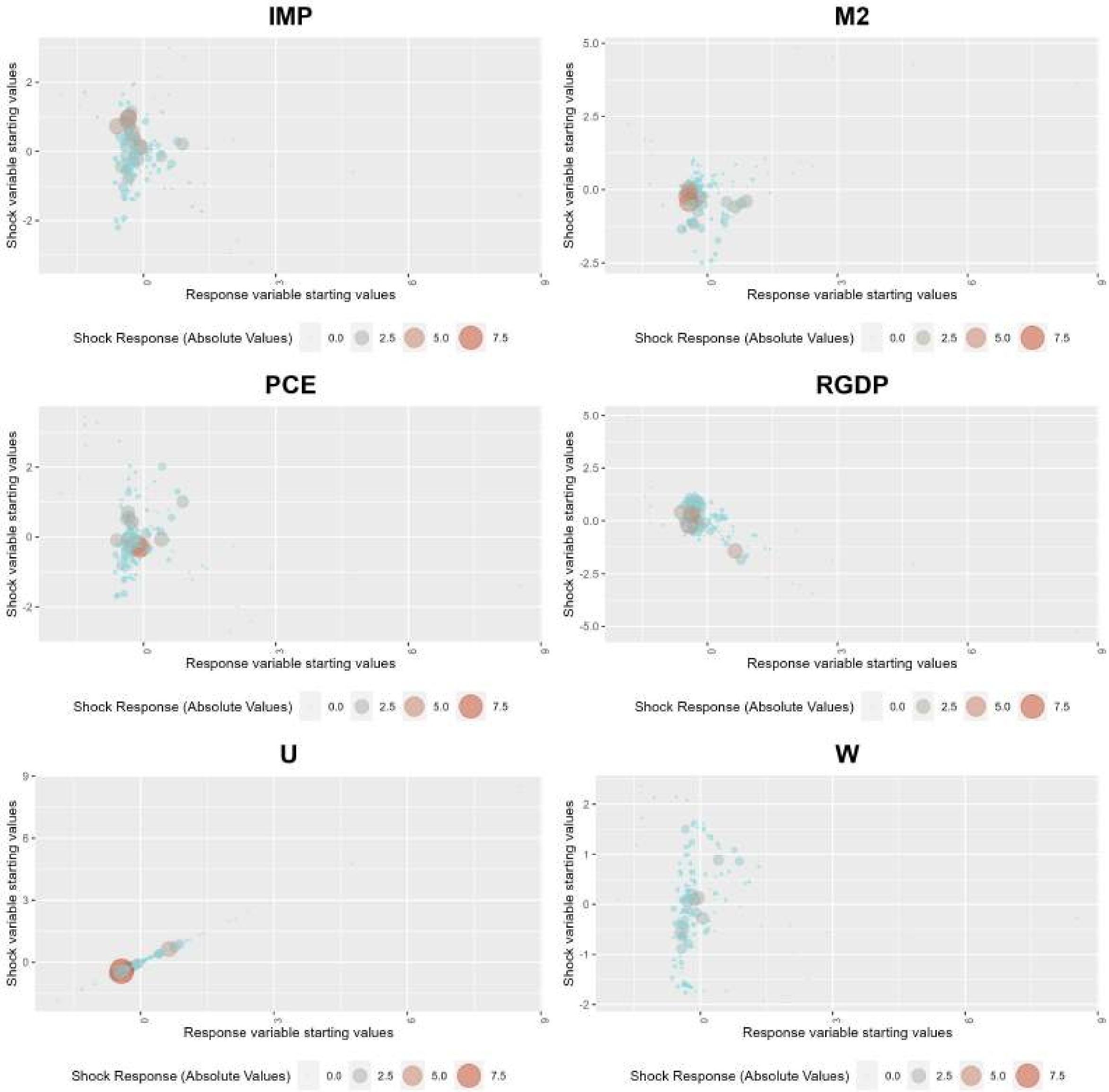
Notes: The figure shows the impact of a 1 standard deviation shock to the money supply on other variables 12 quarters ahead, as measured by the corresponding LSTM multiplier (equation 12). Both axes show the Z-score values of the variables. Notation: IMP, U.S. import price index; $M2$, U.S. money supply; PCE, U.S. personal consumption expenditure index; $RGDP$, U.S. real GDP; U, U.S. unemployment rate; W, U.S. wage rate. Sources: Haver Analytics and the authors.

Figure A.3: LSTM multiplier: personal consumption expenditures shock (PCE), 12 quarters ahead, quarterly model



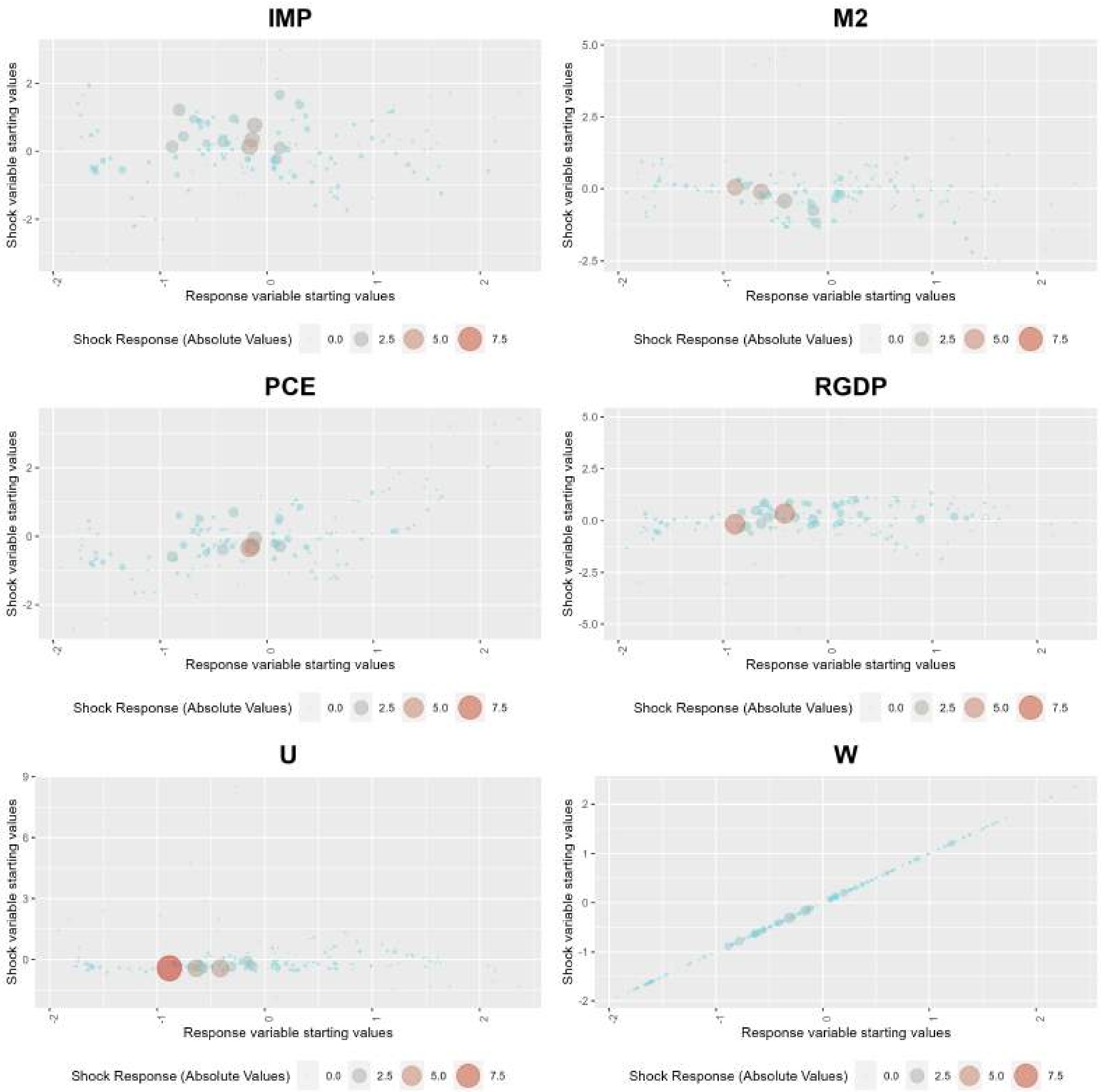
Notes: The figure shows the impact of a 1 standard deviation shock to personal consumption expenditures on other variables 12 quarters ahead, as measured by the corresponding LSTM multiplier (equation 12). Both axes show the Z-score values of the variables. Notation: IMP, U.S. import price index; $M2$, U.S. money supply; PCE, U.S. personal consumption expenditure index; $RGDP$, U.S. real GDP; U, U.S. unemployment rate; W, U.S. wage rate. Sources: Haver Analytics and the authors.

Figure A.4: LSTM multiplier: unemployment shock (U), 12 quarters ahead, quarterly model



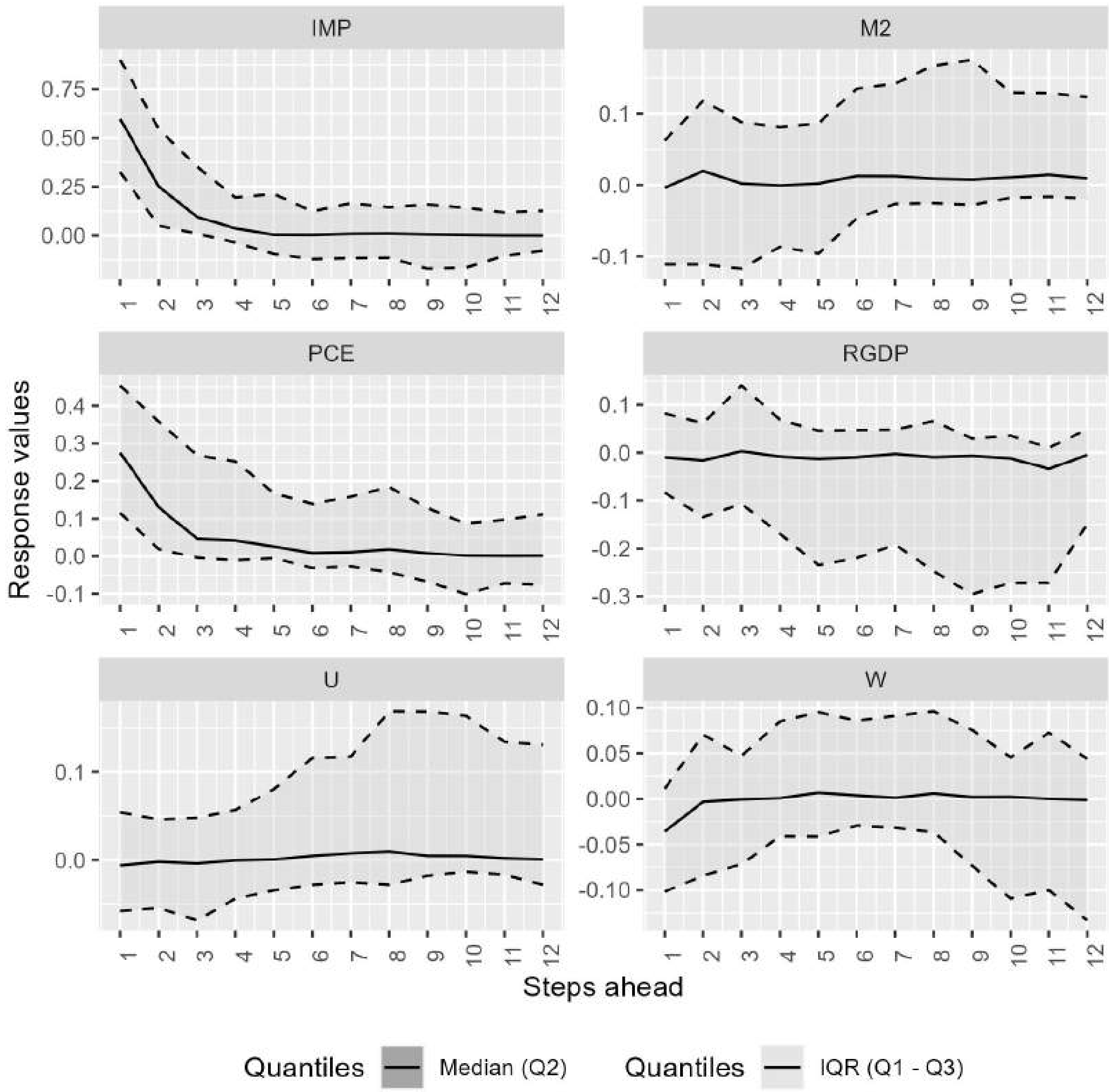
Notes: The figure shows the impact of a 1 standard deviation shock to unemployment on other variables 12 quarters ahead, as measured by the corresponding LSTM multiplier (equation 12). Both axes show the Z-score values of the variables. Notation: IMP, U.S. import price index; $M2$, U.S. money supply; PCE, U.S. personal consumption expenditure index; $RGDP$, U.S. real GDP; U, U.S. unemployment rate; W, U.S. wage rate. Sources: Haver Analytics and the authors.

Figure A.5: LSTM multiplier: wage shock (W), 12 quarters ahead, quarterly model



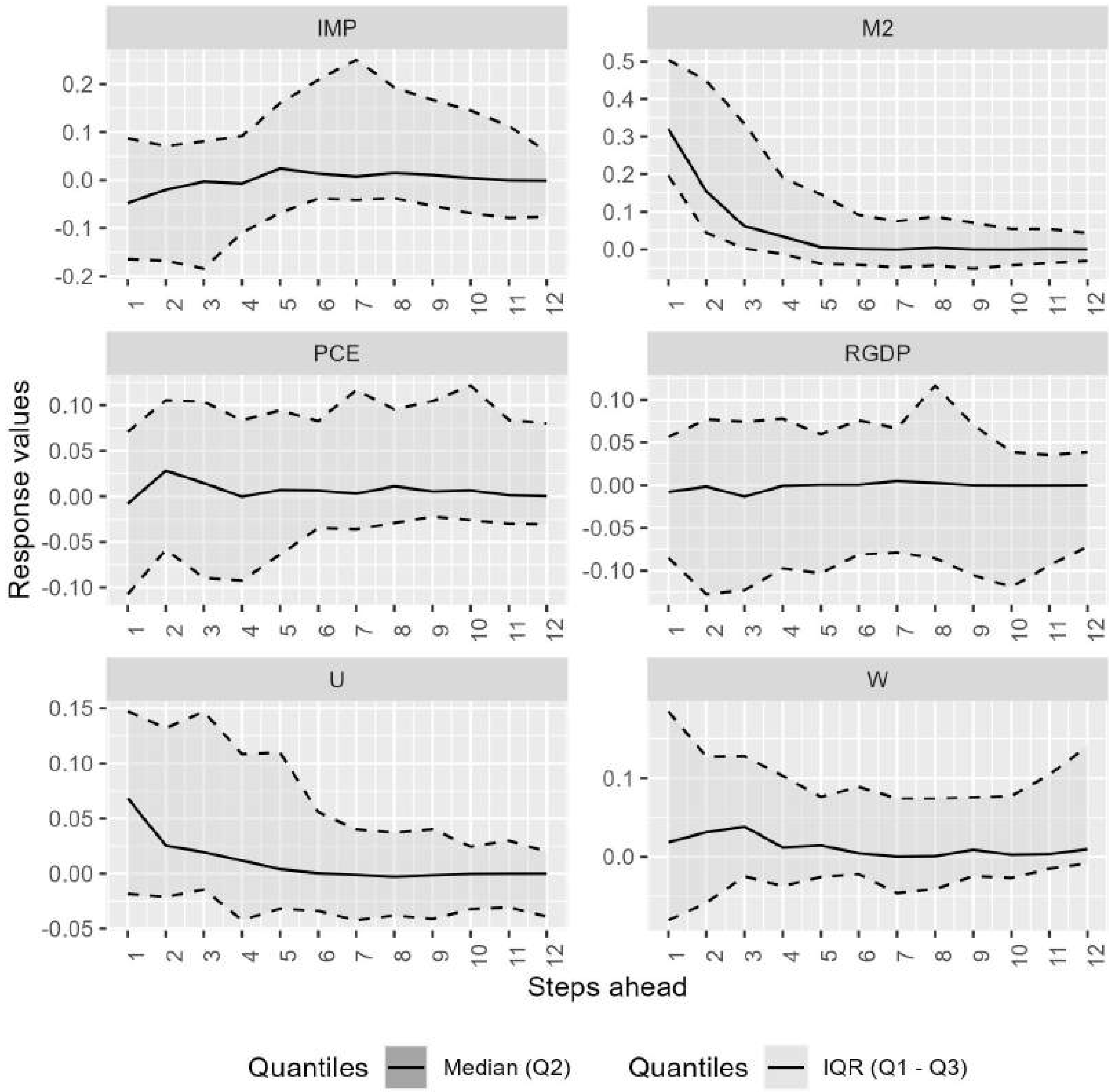
Notes: The figure shows the impact of a 1 standard deviation shock to wages on other variables 12 quarters ahead, as measured by the corresponding LSTM multiplier (equation 12). Both axes show the Z-score values of the variables. Notation: IMP, U.S. import price index; $M2$, U.S. money supply; PCE, U.S. personal consumption expenditure index; $RGDP$, U.S. real GDP; U, U.S. unemployment rate; W, U.S. wage rate. Source: Haver Analytics and the authors.

Figure A.6: LSTM response (LMR): 1-standard deviation shock to import prices (IMP)



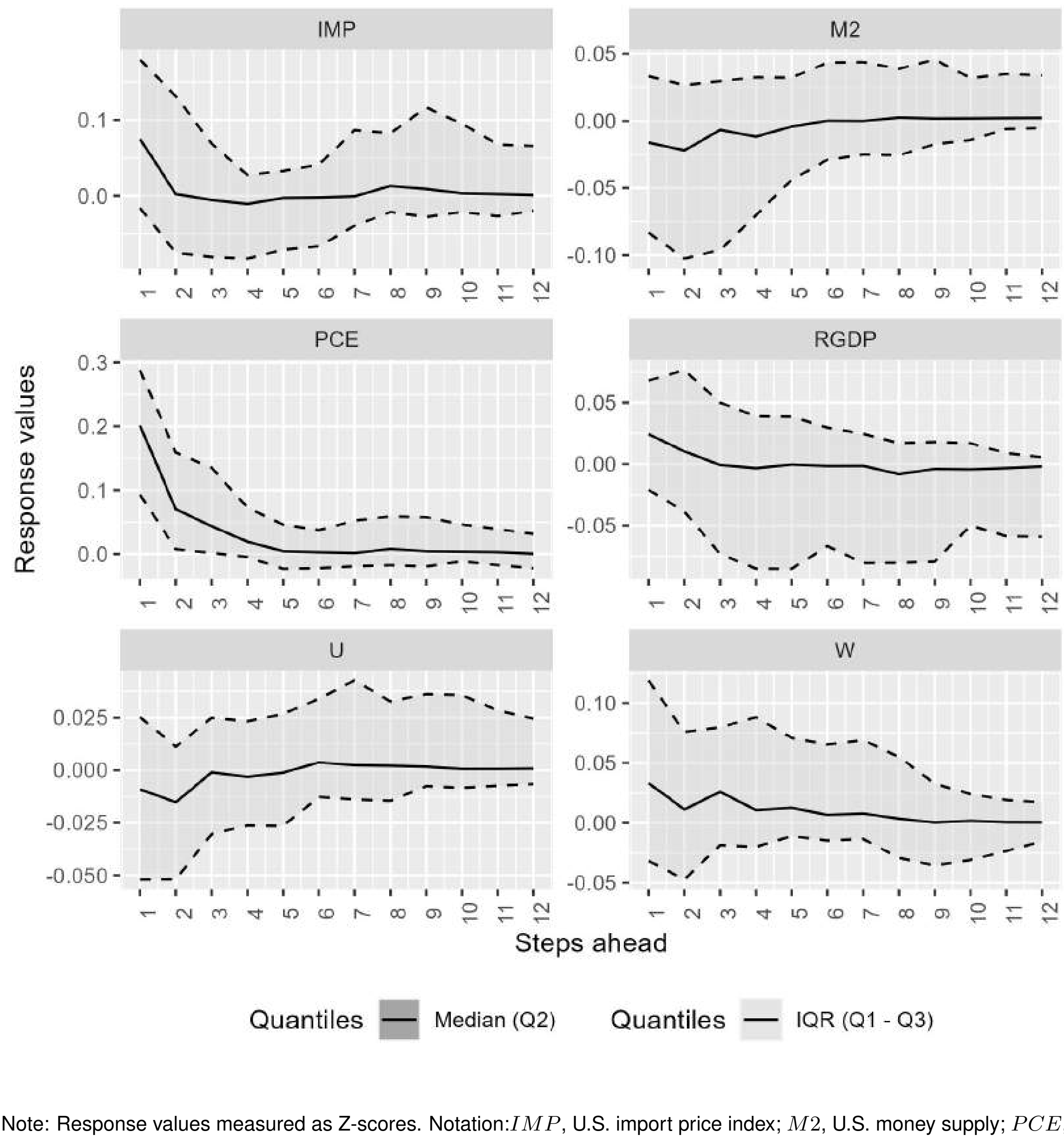
Note: Response values measured as Z-scores. Notation:IMP, U.S. import price index; $M2$, U.S. money supply; PCE, U.S. personal consumption expenditure index; $RGDP$, U.S. real GDP; U, U.S. unemployment rate; W, U.S. wage rate. Sources: Haver Analytics and the authors.

Figure A.7: LSTM response (LMR): 1-standard deviation shock to money supply ($M2$)



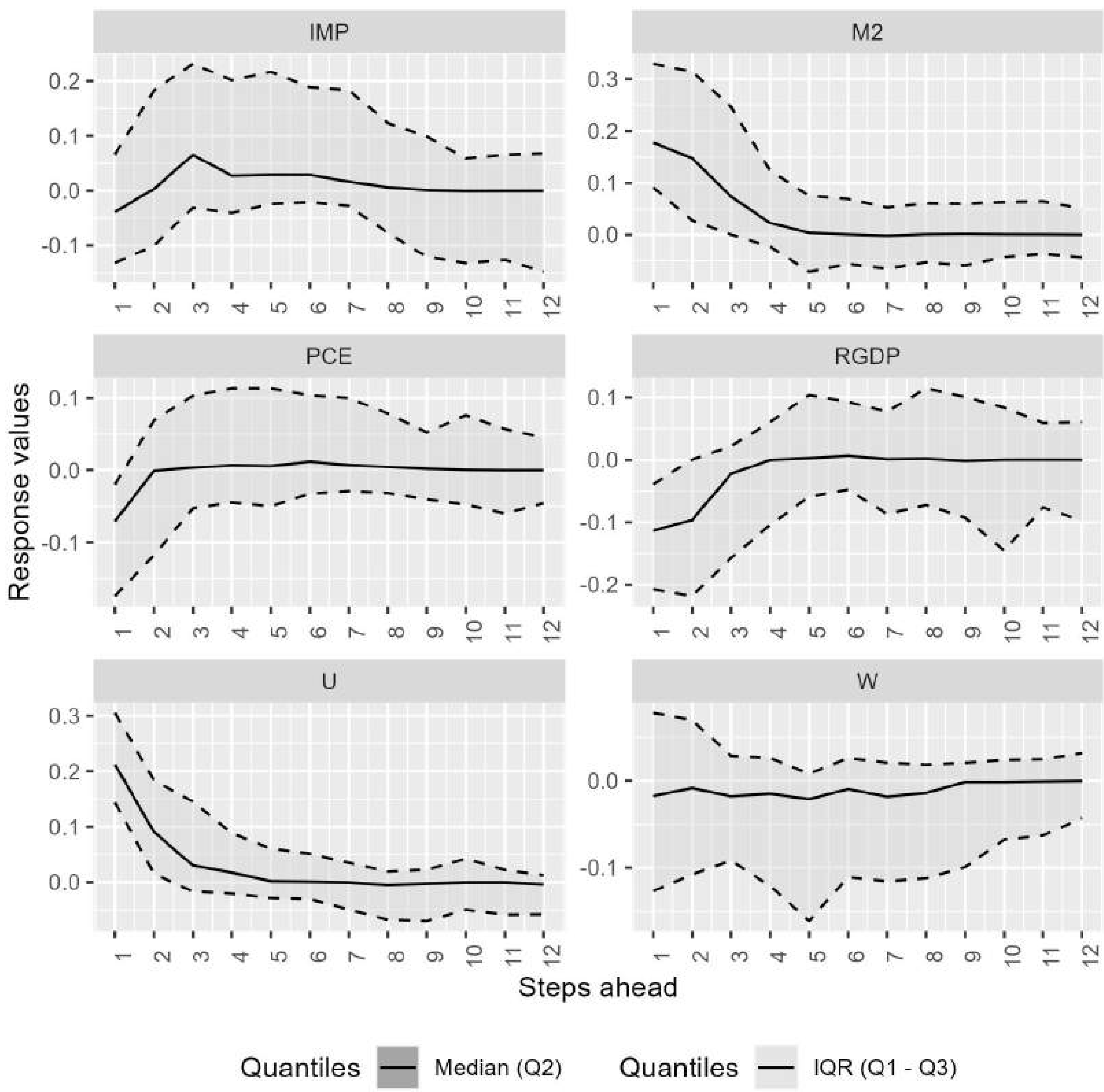
Note: Response values measured as Z-scores. Notation:IMP, U.S. import price index; $M2$, U.S. money supply; PCE, U.S. personal consumption expenditure index; $RGDP$, U.S. real GDP; U, U.S. unemployment rate; W, U.S. wage rate. Sources: Haver Analytics and the authors.

Figure A.8: LSTM response (LMR): 1-standard deviation shock to personal consumption expenditures (PCE)



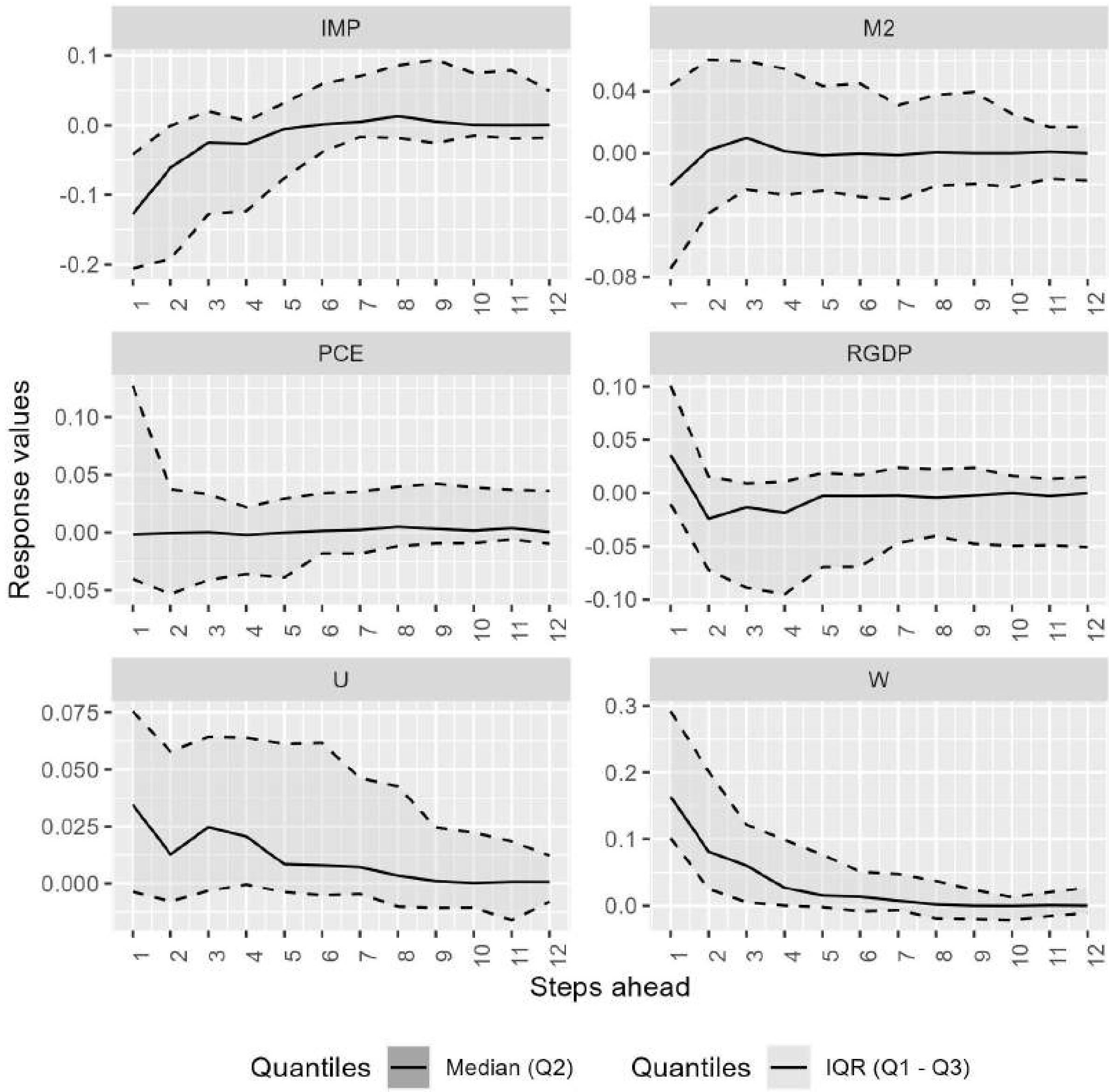
Note: Response values measured as Z-scores. Notation:IMP, U.S. import price index; $M2$, U.S. money supply; PCE, U.S. personal consumption expenditure index; $RGDP$, U.S. real GDP; U, U.S. unemployment rate; W, U.S. wage rate. Sources: Haver Analytics and the authors.

Figure A.9: LSTM response (LMR): 1-standard deviation shock to unemployment (U)



Note: Response values measured as Z-scores. Notation:IMP, U.S. import price index; $M2$, U.S. money supply; PCE, U.S. personal consumption expenditure index; $RGDP$, U.S. real GDP; U, U.S. unemployment rate; W, U.S. wage rate. Sources: Haver Analytics and the authors.

Figure A.10: LSTM response (LMR): 1-standard deviation shock to wages (W)



Note: Response values measured as Z-scores. Notation:IMP, U.S. import price index; $M2$, U.S. money supply; PCE, U.S. personal consumption expenditure index; $RGDP$, U.S. real GDP; U, U.S. unemployment rate; W, U.S. wage rate. Sources: Haver Analytics and the authors.

References

Bollerslev, Tim. 1986. "Generalized Autoregressive Conditional Heteroskedasticity." *Journal of Econometrics* 31 (3): 307–327. https://doi.org/10.1016/0304-4076(86)90063-1.

Buczynski, Mateusz, Marcin Chlebus, Katarzyna Kopczewska, and Marcin Zajenkowski. 2023. "Financial time series models - comprehensive review of deep learning approaches and practical recommendations." *Engineering Proceedings* 39 (79). https://doi.org/10.3390/engproc2023039079.

Cao, Jianlong, Zhi Li, and Jingwei Li. 2019. "Financial time series forecasting model based on CEEMDAN and LSTM." *Physica A: Statistical Mechanics and its Applications* 519:127–139.

Chan-Lau, Jorge Antonio, and Toah Long Quach. 2023. Cryptocurrencies and Banking Sector Connectedness: Don't Worry, Be Happy . . . for Now. AMRO working paper 23-04. https://dx.doi.org/10.2139/ssrn.4720021.

Dell, Melissa. 2025. "Deep Learning for Economists." *Journal of Economic Literature* 63 (1): 5–58. https://doi.org/DOI:10.1257/jel.20241733.

Engle, Robert F. 1982. "Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation." *Econometrica* 50 (4): 987–1007. https://doi.org/10.2307/1912773.

Fernández-Villaverde, Jesús, Galo Nuño, and Jesse Perla. 2024. *Taming the Curse of Dimensionality: Quantitative Economics with Deep Learning.* https://www.nber.org/papers/w33117.

Fischer, Thomas, and Christopher Krauss. 2018. "Deep learning with long short-term memory networks for financial market predictions." *European Journal of Operational Research* 270 (2): 654–669. https://doi.org/10.1016/j.ejor.2017.11.054.

Furizal, Furizal, Asdelina Ritonga, Alfian Ma'arif, and Iswanto Suwarno. 2024. "Stock price forecasting with multivariate time series long-short term memory: a deep learning approach." *Journal of Robotics and Control* 5 (5). https://doi.org/10.18196/jrc.v5i5.22460.

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning.* Cambridge, MA: MIT Press. ISBN: 9780262035613. https://www.deeplearningbook.org/.

Greff, Klaus, Jan Koutnik, Bas R. Steunebrink, and Jürgen Schmidhuber. 2017. "LSTM: a search space odyssey." *IEEE Transactions on Neural Networks and Learning Systems* 28 (10): 2222–2232. https://doi.org/10.1109/TNNLS.2016.2582924.

Guo, Ying, et al. 2023. "Deep learning in economics: A systematic and critical review." *Artificial Intelligence Review* 56:9497–9539. https://doi.org/10.1007/s10462-022-10272-8.

Hamiane, Sana, Yousseff Ghanou, Hamid Khalifi, and Meryam Telmem. 2024. "Comparative analysis of LSTM, ARIMA, and hybrid models for forecasting future GDP." *Ingénierie des Systèmes d'Information* 29 (3): 853–861. https://doi.org/10.18280/isi.290306.

Hamiane, Sana, Hamid Khalifi, Youssef Ghanou, and Gabriella Casalino. 2023. "Forecasting the Gross Domestic Product using LSTM and ARIMA." In *2023 IEEE International Conference on Technology Management, Operations and Decisions (ICTMOD),* 1–6. https://doi.org/10.1109/ICTMOD59086.2023.10438159.

Hamilton, James D. 1989. "A New Approach to the Economic Analysis of Nonstationary Time Series and the Business Cycle." *Econometrica* 57 (2): 357–384. https://doi.org/10.2307/1912559.

Hewamalage, Hansika, Christoph Bergmeir, and Kasun Bandara. 2021. "Recurrent neural networks for time series forecasting: Current status and future directions." *International Journal of Forecasting* 37 (1): 388–427.

Hochreiter, Sepp, and Jürgen Schmidhuber. 1997. "Long Short-Term Memory." *Neural Computation* 9 (8): 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735.

Hollis, Thomas, Antoine Viscardi, and Seung Eun Yi. 2018. A Comparison of LSTMs and Attention Mechanisms for Forecasting Financial Time Series. Working paper. http://arxiv.org/abs/1812.07699.

Hopp, Daniel. 2022. "Economic nowcasting with long short-term memory artificial neural networks (LSTM)." *Journal of Official Statistics* 38 (3): 847–873. https://doi.org/https://doi.org/10.2478/jos-2022-0.

Ji, Yang, Ying Sun, Yuting Zhang, Zhigaoyuan Wang, Yuanxin Zhuang, Zheng Gong, Dazhong Shen, Chuan Qin, Hengshu Zhu, and Hui Xiong. 2025. A Comprehensive Survey on Self-Interpretable Neural Networks. Technical report. arXiv: 2501.15638.

Jordà, Òscar. 2005. "Estimation and inference of impulse responses by local projections." *American Economic Review* 95 (1): 161–182. https://doi.org/10.1257/0002828053828518.

Ju, Jie, and Fang-Ai Liu. 2021. "Multivariate time series data prediction based on ATT-LSTM network." *Applied Sciences* 11 (9373).

Kobiela, Dariusz, Dawid Krefta, Weronika Krol, and Pawel Weichbroth. 2022. "ARIMA vs LSTM on NASDAQ stock exchange data." *Procedia Computer Science* 207:3836–3845. https://doi.org/10.1016/j.procs.2022.09.445.

Kong, Yaxuan, Zepu Wang, Yuqi Nie, Tian Zhou, Stefan Zohren, Yuxuan Liang, Peng Sun, and Qingsong Wen. 2024. Unlocking the power of LSTM for long term time series forecasting. Working paper. https://arxiv.org/abs/2408.10006v1.

Korinek, Anton. 2023. "Generative AI for Economic Research: Use Cases and Implications for Economists." *Journal of Economic Literature* 61 (4): 1281–1371. https://doi.org/10.1257/jel.20231736.

Kraus, Maurice, Felix Divo, Devendra Singh Dhami, and Kristian Kersting. 2024. xLSTM-Mixer: Multivariate Time Series Forecasting by Mixing via Scalar Memories. Working paper. https://arxiv.org/abs/2410.16928.

Kumarappan, Jayaraman, Elakkiya Rajasekar, Subramaniyaswamy Variavasundaram, Ketan Kotecha, and Ambarish Kulkarni. 2024. "Federated Learning Enhanced MLP–LSTM Modeling in an Integrated Deep Learning Pipeline for Stock Market Prediction." *International Journal of Computational Intelligence Systems* 17:267. https://doi.org/10.1007/s44196-024-00680-9.

Lakshmi Narayanaa, T, R R Skandarsini, S. Jhansi Ida, S. Rathana Sabapathy, and P Nanthitha. 2023. "Inflation Prediction: A Comparative Study of ARIMA and LSTM Models Across Different Temporal Resolutions." In *2023 3rd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA),* 1390–1395. https://doi.org/10.1109/ICIMIA60377.2023.10425970.

Lashina, Maria, and Sergey Grishunin. 2023. "Comparison of forecasting power of statistical models for GDP growth under conditiosn of permanent crises for application in strategic risk controlling." *Procedia Computer Science* 221:442–449. https://doi.org/10.1016/j.procs.2023.07.059.

LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. 2015. "Deep learning." *Nature* 521 (7553): 436–444. https://doi.org/10.1038/nature14539.

Lim, Bryan, and Stefan Zohren. 2021. "Time-series forecasting with deep learning: A survey." *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 379 (2194): 20200209. https://doi.org/10.1098/rsta.2020.0209.

Liu, Sixing, and Quanxiang Lan. 2025. "LSTM economic forecasting model based on Adam optimization." *ICCSMT '24: Proceeding of the 2024 5th International Conference on Computer Science and Management Technology,* 1408–1413.

Makridakis, Spiros, Evangelos Spiliotis, and Vassilio Assimakopoulos. 2018. "The M4 competition: results, findings, conclusions and way forward." *International Journal of Forecasting* 34 (4): 802–808. https://doi.org/10.1016/j.ijforecast.2018.06.001.

Molnar, Christoph. 2025. *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable.* 3rd. Available online: https://christophm.github.io/interpretable-ml-book/. Self-published.

Ormaniec, Weronika, Marcin Pitera, Sajad Safarveisi, and Thorsten Schmidt. 2022. Estimating value at risk: LSTM vs. GARCH. Working paper. https://doi.org/https://arxiv.org/abs/2207.10539.

Paranhos, Livia. 2025. “Predicting inflation with recurrent neural networks.” *International Journal of Forecasting,* https://doi.org/10.1016/j.ijforecast.2024.07.010.

Park, Sanjing, and Jae-Suk Yang. 2022. “Interpretable deep learning LSTM model for intelligent economic decision-making.” *Knowledge-Based Systems* 248 (108907). https://doi.org/10.1016/j.knosys.2022.108907.

Pilla, Prashant Reddy, and Raji Mekonen. 2025. Forecasting SP 500 Using LSTM Models. Working paper. https://doi.org/10.5281/ZENODO.14759118.

Salem, Abdorfwf A Mohamed, Amaal Jummah, and Abdullah Albourawi. 2024. “Predictive models for interest rate forecasting using machine learning: a comparative analysis and practical applications.” *Brilliance: Research of Artificial Intelligence* 4 (2). https://doi.org/10.47709/brilliance.v4i2.4983.

Serin, Selim, and Gülder Kemalbay. 2024. “LSTM-based multivariate deep neural networks for stock price forecasting.” In *Güncel Ekonometrik ve İstatistiksel Uygulamalar ile Akademik Çalışmalar.* Özgür Yayınları.

Sezer, Omer Berat, Mehmet Ugur Gudelek, and Ahmet Murat Ozbayoglu. 2020. “Financial time series forecasting with deep learning: A systematic literature review: 2005–2019.” *Applied Soft Computing* 90:106181. https://doi.org/10.1016/j.asoc.2020.106181. https://www.sciencedirect.com/science/article/pii/S1568494620300960.

Shamsi, Jawwad A., et al. 2021. “LSTM based stock prediction using weighted and categorized financial news.” *PLOS ONE* 16 (3): e0246674. https://doi.org/10.1371/journal.pone.0282234.

Shi, Rui (Aruhan). 2023. “Deep reinforcement learning and macroeconomic modelling.” Unpublished. http://webcat.warwick.ac.uk/record=b3988261.

Siami-Namini, Sima, Neda Tavakoli, and Akbar Siami Namin. 2018. “A Comparison of ARIMA and LSTM in Forecasting Time Series.” In *17th IEEE International Conference on Machine Learning and Applications (ICMLA),* 1394–1401. IEEE.

Sims, Christopher A. 1980. “Macroeconomics and Reality.” *Econometrica* 48 (1): 1–48. https://doi.org/10.2307/1912017.

———. 1992. “Interpreting the Macroeconomic Time Series Facts: The Effects of Monetary Policy.” *European Economic Review* 36 (5): 975–1000. https://doi.org/10.1016/0014-2921(92)90041-T.

Sivakumar, Guhan. 2025. HMM-LSTM Fusion Model for Economic Forecasting. Working paper. https://arxiv.org/abs/2501.02002.

Snoek, Jasper, Hugo Larochelle, and Ryan P. Adams. 2012. "Practical Bayesian optimization of machine learning algorithms." In *: Proceedings of the 26th International Conference on Advances in Neural Information Processing Systems (NeurIPS'12),* edited by F. Pereira, C. Burges, L. Bottou, and K. Weinberger, 2:2951–2959. https://dl.acm.org/doi/10.5555/2999325.2999464.

Stock, James H., and Mark W. Watson. 2001. "Vector Autoregressions." *Journal of Economic Perspectives* 15 (4): 101–115. https://doi.org/10.1257/jep.15.4.101.

Xie, Huaqing, Xingcheng Xu, Fangjia Yan, Xun Qian, and Yanqing Yang. 2024. Deep Learning for Multi-Country GDP Prediction: A Study of Model Performance and Data Impact. Working paper. https://arxiv.org/abs/2409.02551.

Xu, Ke, Yu Cheng, Shiqing Long, Junjie Guo, Jue Xiao, and Mengfang Sun. 2024. Advancing Financial Risk Prediction Through Optimized LSTM Model Performance and Comparative Analysis. Working paper. https://arxiv.org/abs/2405.20603.

Zhang, Aston, Zachary C. Lipton, Mu Li, and Alexander J. Smola. 2023. *Dive into Deep Learning.* Https://D2L.ai. Cambridge University Press.

Zhang, Junhuan, Jiaqi Wen, and Zhen Yang. 2022. "China's GDP forecasting using long short term memory recurrent neural network and hidden markov model." *PLOS One* 17 (6): e0269529. https://doi.org/10.1371/journal.pone.0269529.

Zhang, Xuan, Xun Liang, Aakas Zhiyuli, Shusen Zhang, Rui Xu, and Bo Wu. 2019. "AT-LSTM: and attention-based LSTM model for financial time series prediction." *IOP Conference Series: Materials Science and Engineering* 569 (5). https://doi.org/10.1088/1757-899X/569/5/05237.

Zhang, Yu, Peter Tiňo, Aleš Leonardis, and Ke Tang. 2021. "A Survey on Neural Network Interpretability." *IEEE Transactions on Emerging Topics in Computational Intelligence* 5 (5): 726–742. https://doi.org/10.1109/TETCI.2021.3100641. arXiv: 2012.14261.

Zhao, Jiayao. 2024. "The Macroeconomic Prediction Model based on LSTM Improved Algorithm." In *2024 International Conference on Integrated Circuits and Communication Systems (ICICACS),* 1–7. https://doi.org/10.1109/ICICACS60521.2024.10498821.

[This page is intentionally left blank]

ASEAN +3 MACROECONOMIC RESEARCH OFFICE

Address: 10 Shenton Way, #15-08
MAS Building, Singapore 079117
Website: www.amro-asia.org
Tel: +65 6323 9844
Email: enquiry@amro-asia.org
LinkedIn | Twitter | Facebook | YouTube