US GDP Forecasts Without US Data? Why Pooled Cross Country Data Improves US GDP Forecasting and Pooled Data Without The US Improves Forecasts Even More

Cameron Fen and Samir Undavia*

May 6, 2024

Economic forecasting is both an essential exercise and a way to discipline the construction of models. We show that pooling data from many countries in a cross-sectional dimension to macroeconomic data can statistically significantly improve the generalization ability of structural, reduced-from, and machine learning models in forecasting GDP. This procedure reduces root mean squared error (RMSE) by 12 percent across horizons for certain reduced form models and by 24 percent across horizons for structural DSGE models. A central result in this paper is that, in contrast with theory, DSGE or reduced from models forecast better, or at least as well, when estimated on more recent time series across a cross-section of countries rather than data that extends further back in time but only from the country of interest. Finally, given the comparative advantage of "nonparametric" Machine Learning forecasting models in a data-rich regime, we demonstrate that our Recurrent Neural Network (RNN) model and Automated Machine Learning (AutoML) approach outperform all baseline Economic models and even the Survey of Professional Forecasters (Philadelphia Federal Reserve, 1968) at long term horizons when using pooled macroeconomic data. Robustness checks indicate that machine learning techniques are reproducible, numerically stable, and generalize across models.

JEL Codes: E27, C45, C53, C32
Keywords: Neural Networks, Deep Learning, Policy-Invariance, GDP Forecasting, Pooled Data

*Corresponding author Cameron Fen is a PhD student at the University of Michigan, Ann Arbor, MI, 48104 (E-mail: camfen@umich.edu. Website: cameronfen.github.io.). Samir Undavia is an ML and NLP engineer (E-mail: samir@undavia.com). The authors thank Xavier Martin Bautista, Thorsten Drautzburg, Florian Gunsilius, Jeffrey Lin, Daniil Manaenkov, Matthew Shapiro, Eric Jang, John Leahy, Toni Whited, Mark Newman and the member participants of the European Winter Meeting of the Econometric Society, the ASSA conference, the Midwestern Economic Association conference, EconWorld Conference, and the EcoMod Conference for helpful discussion and/or comments. We would also like to thank Nimarta Kumari for her research assistance in assembling the DSGE panel data set and AI Capital Management for computational resources. All errors are our own.

I. Introduction

A central problem in both reduced-form and Structural Macroeconomics is the dearth of underlying data. For example, GDP is a quarterly dataset that only extends back to the late 1940s, around 300 timesteps. Thus generalization and external validity of these models are a pertinent problem. In forecasting, this approach is partially addressed by using simple linear models. In structural macroeconomics, the use of micro-founded parameters and Bayesian estimation attempts to improve generalization to limited effect. More flexible and nonparametric models would likely produce more accurate forecasts, but with limited data, this avenue is not available. However, pooling data across many different countries allows economists to forecast and even estimate larger structural models which have both better external validity and forecasting when predicting GDP, without having to compromise on internal validity or model design.

This paper sets out to answer a series of questions stemming from the main question: If and how does pooled data from across many (50+) countries improve the GDP forecasting performance of linear, structural, and machine learning models? A bias and variance trade-off is occurring. On the one hand, GDP data is correlated across countries and many structural and machine learning models have many parameters making them prone to overfitting. On the other hand, countries are different and pooling may lead to bias.

Focusing on US GDP prediction, we use a US-only data-set (211 timesteps), a world data set containing economic data from the US and 49 other countries (2581 timestep-countries in total), and a country out-of-sample data set that is the world dataset without US data (2370 timestep-countries). In most cases, the model estimated on world data outperforms the model estimated on the US-only data set in out-of-sample forecasting of US GDP. These results also hold across less flexible and smaller models like Autoregressive (AR) or Vector Autoregressive (VAR) models, but the improvement is more limited.

More interestingly, models estimated on the world data set ex-US, often outperforms the world data even though world data contains additional US data, directly relevant to US GDP forecasting. This result suggests data pooling many countries across space, may be more informative than one

country extending across a longer gap in time. For example 50 years in the past. A central result of this paper contained in part of results Section VI.B. and part of the Section VI.A. suggests US economic data in the 1970s seems to bias US GDP forecasts in the 2020s more than, for example, French economic data in 2010s. This empirical result is in direct contrast to the theory that suggests estimating pooled countries' data will lead to parameter estimates that don't correspond to economic primitives and will lead to models that are more mismatched concerning the true structural model (Pesaran and Smith, 1995).

Finally, we run a horserace among all models. Model improvement indicates that while these approaches are competitive in the low data regime, machine learning methods consistently outperform baseline economic models – VAR(1), VAR(4), AR(2), Smets-Wouters (Smets and Wouters, 2007), and Factor models – in the high data regime. Over most horizons, our ML models approach SPF median forecast performance, albeit evaluated on 2020 vintage data (see Appendix C), resulting in outperformance over SPF benchmark at 5 quarters ahead.

The paper proceeds as follows: Section II. reviews the literature on forecasting and Recurrent Neural Networks (RNNs) and describes how our paper merges these two fields; Section III. discusses Feed-Forward Neural Networks, linear state-space models, and gated recurrent units (Cho et al., 2014); Section III.F. briefly mentions our model architecture; Section IV. discusses the benchmark Economic models and the SPF (Philadelphia Federal Reserve, 1968) that we compare our model to; Section V. describes the data; Section VI. and Appendix I.2 provide the main results and robustness checks, and Section VII. concludes the paper.

II. Literature Review

This paper connects multiple strands of literature: Machine Learning, time-series econometrics, and panel Macroeconomic analysis. I will focus on the economic literature here and discuss the machine learning methods in the methods section.

II.A. Economic Models

One advancement in forecasting stems from the greater adoption of structural or pseudo-structural time series models like the Smets-Wouters DSGE models (Smets and Wouters, 2007). While DSGE forecasting is widely used in the literature, it is competitive with, but often no better than, a simple AR(2), with more bespoke DSGE models performing poorer (Edge, Kiley and Laforte, 2010). However, the use of DSGE models for counterfactual analysis is an important and unique benefit of these models. The final economic baseline is the Factor Model (Stock and Watson, 2002*a*), which attempts to use a large cross-section of data resulting in a more comprehensive picture of the economy to perform forecasting.

Our paper uses tools from forecast evaluation –West (1996), Pesaran and Timmermann (1992), and Diebold and Mariano (2002) – as well as model averaging – Koop, Leon-Gonzalez and Strachan (2012), Timmermann (2006), and Wright (2008). Details on all these models and our implementation can be found in Appendix D.

Moving to structural economics, there is little literature on panel data and dynamic general equilibrium models (Breitung, 2015). Most of it focuses on the use of panel data to better identify the effects of interest across countries. There is also literature looking at specific panel models applied to Macroeconomics like dynamic panel models – Doran and Schmidt (2006), Bai and Ng (2010), and Diebold, Rudebusch and Aruoba (2006).

III. Methods: Machine Learning Models

We use two main Machine Learning methods. First, we use a recurrent neural network (RNN) model, mainly the Gated Recurrent Unit (GRU) (Cho et al., 2014), but also Long Short-Term Memory Networks (LSTMs) (Hochreiter and Schmidhuber, 1997*a*) as a robustness check. Second, we apply AutoML (Hutter et al., 2014), an automated algorithm that estimates and compares many machine learning models.

III.A. Automated Machine Learning (AutoML)

Automated Machine Learning (AutoML) software is designed to provide end-to-end solutions for Machine Learning problems by training and evaluating many different models and ultimately returning the best models in order of performance. We just used the default H2o AutoML library's collection of models for our estimation routine. The approach automates the process of picking and training a host of models and produces a top-performance leaderboard. To provide a proxy for the performance of a good and flexible Machine Learning model, we tested the open-source Automated Machine Learning (AutoML) software H2O[1]. We created a pipeline for each prediction horizon, trained the model using our international cross-sectional data, evaluated on US GDP validation data, and lastly, predicted using our US GDP data test set. In contrast with our own custom model, setting up H2O and training on our dataset was almost entirely automated. A consequence is that AutoML is automated makes unintentional p-hacking difficult.

From predicting one quarter to five quarters ahead, the AutoML software picked the model for each horizon: XGBoost, gradient boosting machine, gradient boosting machine again, distributed random forest, and deep learning, respectively. We noticed that the software generally picked deep learning models for the quarters that were further away while picking gradient-boosting techniques for closer quarters. Ultimately, AutoML had strong results and can be applied to other prediction problems in economics.

Additionally, because the AutoML selects a different model for a given horizon and data set size, we also estimated a GRU RNN on both the reduced USA dataset and the pooled world data set. This allows us to show the effect of the increase in data size holding the model architecture fixed. The RNN also has the advantage of not being a model considered by AutoML, which gives broader coverage of the universe of Machine Learning models that are being considered in our paper.

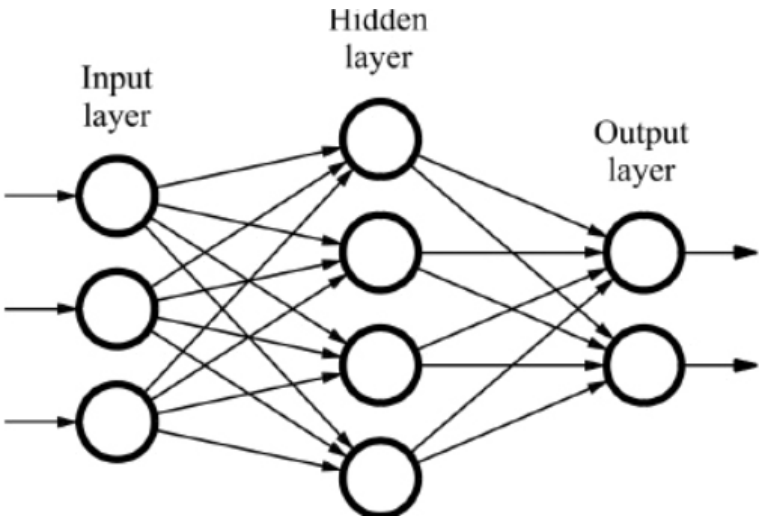

FIGURE I: AN EXAMPLE OF A FEED-FORWARD NEURAL NETWORK

III.B. Neural Network Models

III.B..1 Feed-Forward Neural Network

The feed-forward neural network is the prototypical image associated with deep learning. At its core, a feed-forward neural network is a recursively nested linear regression with nonlinear transforms. For example, assume X_{in} is a vector-valued input to the neural network and X_{out} is the output. In a typical linear regression, $X_{out} = X_{in}\beta_1$. In deep learning parlance, this linear relationship between input and output is one layer. The insight for composing a feed-forward network is to take the output and feed that into another linear regression: $Y = X_{out}\beta_2$, i.e., a higher layer. In Figure I, X_{in} would be the input layer, X_{out} would be the hidden layer and Y would be the output layer. The problem is not all that interesting if X_{out} is a linear equation. If X_{in} is a matrix of dimension timesteps by regressors, X_{out} can be a matrix of dimension timesteps by hidden units. Here in the figure, the dimension of the hidden layer is four, so β_1 has to be a matrix of dimension three by four (regressors by hidden units). Thus, we make X_{out} an input into a multidimensional regression for the second layer, $Y = X_{out}\beta_2$, if the first layer is a vector

1. https://www.h2o.ai/

regression.[2] This can be repeated for as many layers as desired.

Now a composition of two layers will result in: $Y = X_{out}\beta_2 = (X_{in}\beta_1)\beta_2$. A product of two matrices is still another matrix, which means the model is still linear. Clearly, this will hold no matter how many layers are added. However, an early result in the literature showed that if, between every regression, eg $X_{out} = X_{in}\beta_1$, one inserts an almost arbitrary nonlinear link function, this allows a neural network to approximate any continuous function (Hornik, Stinchcombe and White, 1989). For example, inserting a logistic transformation between X_{in} and X_{out}, i.e. $X_{out} = \sigma(X_{in}\beta_1)$, where $\sigma(z) = \frac{1}{1+e^{-z}}$, achieves this objective. One can put these nonlinearities as often as one would like to get something like this: $Y = \sigma(\sigma(X_{in}\beta_{1,1})\beta_2)$. These are the fundamental building blocks of neural networks and allow these models to be universal approximators.

III.C. The Simplest Recurrent Neural Network: A Linear State-Space Model

Without even knowing it, many economists are already familiar with RNNs. The simplest is a Kalman filter-like linear state-space model. The two equations that define the linear state-space model are[3]:

$$\mathbf{s}_t = \mathbf{s}_{t-1}\mathbf{U}^s + \mathbf{y}_{t-1}\mathbf{W}^s + \mathbf{b}^s, \tag{1}$$

$$\mathbf{y}_t = \mathbf{s}_t\mathbf{U}^y + \mathbf{y}_{t-1}\mathbf{W}^y + \mathbf{b}^y \tag{2}$$

In a linear state-space model, the state $\mathbf{s}_i$ is an unobserved variable that allows the model to keep track of the current environment. One uses the state, along with lagged values of the observed variables, to forecast observed variables $\mathbf{y}_i$. For example, for GDP, the state could be either an expansionary period or a recession – a priori, the econometrician does not know. However, one can make an educated guess based on GDP growth. As Machine Learning is more interested in prediction, the state is often estimated with point estimates, which allows the data scientist to

2. Note: this regression is not a vector autoregression as X_{out} is a latent variable
3. We add autoregressive lags to make the model more general.

sidestep the tricky problem of filtering.

III.D. Estimating the Parameters of a Linear State-Space Model on Data

The two equations that define the linear state-space model are

$$Y_t = D * S_t + E * Y_{t-1} + F, \tag{3}$$

$$S_t = A * S_{t-1} + B * Y_{t-1} + C \tag{4}$$

We use Equations (3) and (4) to recursively substitute for the model prediction at a particular period, so the forecast for period 1 then is:

$$\hat{y}_1 = D * (A * 0 + B * Y_0 + C) + E * Y_0 + F \tag{5}$$

and the forecast for period 2 is:

$$\hat{y}_2 = D * (A * (A * 0 + B * Y_0 + C) + B * Y_1 + C) + E * Y_1 + F \tag{6}$$

Hatted variables indicate predictions and unhatted variables correspond to actual data. Additional periods would be solved by iteratively substituting for the state using Equations (3) and (4) for the previous state. To update the parameters matrices A, B, C, D, E, and F, the gradient is derived for each matrix and each parameter is updated via hill climbing. We will illustrate the process of hill climbing by taking the gradient of one parameter, B:

$$\frac{\partial \sum_{\forall t} L(y - \hat{y})}{\partial B} = \frac{\partial L(y_1 - \hat{y}_1)}{\partial B} + \frac{\partial L(y_2 - \hat{y}_2)}{\partial B} \tag{7}$$

Here $L()$ indicates the loss function. Substituting for y_1' and y_2' with Equations (5) and (6) into (7) and using squared error as the loss function, we arrive at an equation with which we can take partial derivatives for with respect to A:

$$\begin{aligned}(8)\quad \frac{\partial}{\partial B}L = &\frac{\partial}{\partial B}\frac{1}{2}(y_1 - D*(A*0+B*Y_0+C)+E*Y_0+F)^2 \\ &+\frac{\partial}{\partial B}\frac{1}{2}(y_2 - D*(A*(A*0+B*Y_0+C)+B*Y_1+C)+E*Y_1+F)^2\end{aligned}$$

Distributing all the B's and taking the derivative of (8) results in $\frac{\partial}{\partial B}L = -(y_1 - D*(A*0+B*Y_0+C)+E*Y_0+F)*D*Y_0-(y_2-D*(A*(A*0+B*Y_0+C)+B*Y_1+C)+E*Y_1+F)*(D*A*Y_0+D*Y_1)$ which provide the gradients for hill climbing. In practice, the derivatives are taken automatically in code.

III.E. Gated Recurrent Units (GRUs)

GRUs (Cho et al., 2014) were introduced to improve upon the performance over previous RNNs that resembled linear state-space models and can deal with the exploding gradient problem.

The problem with linear state-space models is that if one does not apply filtering, the state vector either blows up or goes to a steady state value. This can be seen by recognizing that each additional timestep results in the state vector getting multiplied by $\mathbf{U}^s$ an additional time. Depending on if the eigenvectors of $\mathbf{U}^s$ are greater than or less than one, the states will ultimately explode (go to infinity) or go to a steady state. More sophisticated RNNs, like the GRU (Cho et al., 2014) used in this paper, fix this with the use of gates.

First, we redefine σ as the logistic link function gate:

$$(9)\qquad \sigma(x) = \frac{e^{\beta x}}{1+e^{\beta x}}$$

The idea behind the gate is to allow the model to control the magnitude of the state vector. A simple gated RNN looks like the linear state-space model with an added gate equation:

$$\mathbf{y}_t = \mathbf{h}_t \mathbf{U}^y + E * \mathbf{y}_{t-1} \mathbf{W}^y + \mathbf{b}^y \tag{10}$$

$$\mathbf{z}_t = \sigma(\mathbf{h}_{t-1} \mathbf{U}^h + \mathbf{y}_{t-1} \mathbf{W}^h + \mathbf{b}^h) \tag{11}$$

$$\mathbf{s}_t = \mathbf{h}_{t-1} \mathbf{U}^s + \mathbf{y}_{t-1} \mathbf{W}^s + \mathbf{b}^s \tag{12}$$

$$\mathbf{h}_t = \mathbf{z}_t \odot \mathbf{s}_t \tag{13}$$

The output of $\sigma()$ is a number between zero and one which is element-wise multiplied by $\mathbf{s}_t$, the first draft of the state. The operation $\odot$ indicates element-wise multiplication. Variables are subscripted with the period they are observed in (t or $t-1$). Weight matrices, which are not a function of the inputs, are superscripted with the equation name they feed into. All elements are considered vectors and matrices, and matrix multiplication is implied when no operation is present.

The presence of the gate controls the behavior of the state, which means that even if the eigenvalues of $\mathbf{U}^s$ were greater than one, or equivalently, even if $\mathbf{h}_t$ would explode without the gate, the gate can keep the state bounded. Additionally, the steady-state distribution of the state does not have to converge to a number. The behavior could be periodic, or even chaotic (Zerroug, Terrissa and Faure, 2013). This allows for the modeling of more complex behavior as well as the ability of the state vector to "remember" behavior over longer time periods (Chung et al., 2014).

The equations of the gated recurrent unit are:

$$\mathbf{y}_t = \mathbf{h}_t \mathbf{U}^y + E * \mathbf{y}_{t-1} \mathbf{W}^y + \mathbf{b}^y \tag{14}$$

$$\mathbf{z}_t = \sigma(\mathbf{x}_t \mathbf{U}^z + \mathbf{h}_{t-1} \mathbf{W}^z) \tag{15}$$

$$\mathbf{r}_t = \sigma(\mathbf{x}_t \mathbf{U}^r + \mathbf{h}_{t-1} \mathbf{W}^r) \tag{16}$$

$$\mathbf{s}_t = \tanh(\mathbf{x}_t \mathbf{U}^s + (\mathbf{h}_{t-1} \odot \mathbf{r}_t) \mathbf{W}^s) \tag{17}$$

$$\mathbf{h}_t = (1 - \mathbf{z}_t) \odot \mathbf{s}_t + \mathbf{z}_t \odot \mathbf{h}_{t-1} \tag{18}$$

Tanh is defined as the hyperbolic tangent:

$$\tanh(x) = \frac{e^{2*x} - 1}{e^{2*x} + 1} \tag{19}$$

Like the linear state-space model, the state vector of the gated recurrent unit persists over timesteps in the model. Mapping these equations to Equation (10)-(13), Equation (18) is the measurement equation (analogous to Equation (10)). Equation (15) and (16) are both gates and analogous to Equation (11). Equation (17) is the first draft of the state before the gate $\mathbf{z}_t$ is applied and resembles Equation (12). Equation (18) is the final draft of the state after $\mathbf{z}_t$ is applied and resembles Equation (13).

The RNN is optimized using gradient descent, where the derivative of the loss function with respect to the parameters is calculated via the chain rule/reverse mode differentiation. The gradient descent optimizer algorithm we use is Adam (Kingma and Ba, 2014), which shares similarities with a quasi-Newton approach. See Section G for more information.

III.F. Our Neural Network Model Architecture

Figure II is the picture of our RNN model we use to supplement AutoML based on the Gated Recurrent Unit (GRU) model, described in Section III.E.. The input comes from the top, passes through all the lines and boxes and produces an output from Dense(6) which is forecast for time horizons 1 through 6. Lines between boxes indicate the output of the higher box flows into the input of the lower box. There are Add boxes that add two inputs, Dense boxes which are Feedforward layers (See III.B..1), GRU boxes which are the GRU RNN (See III.E.), Concatenate boxes which combine the covariates from two different outputs, and Batch Normalization discussed in Section F. We use rectified linear unit (Agarap, 2018) activation (see section E) between dense layers. We used the GRU architecture over the more common Long-Short Term Memory (Hochreiter and Schmidhuber, 1997*b*) as the GRU outperformed on validation data. While systematic architectural design was not implemented, basic architectural modifications were applied and evaluated on the validation set to get an ultimate structure.

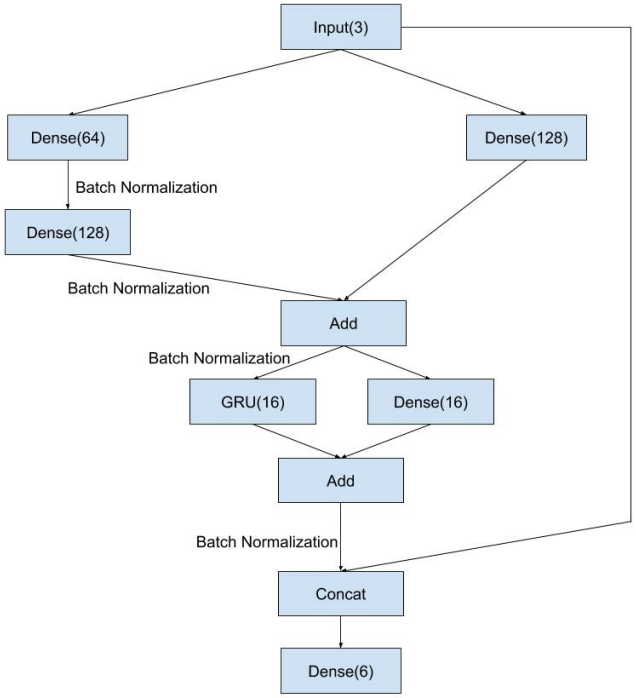

Figure II: Our RNN Model Architecture

Our model contains parallel dense layers between each operation; the layers were originally skip connections (He et al., 2015), but we modified them to allow for learning of linear parameters. The final skip connection concatenates the input with the output of the network so that the neural network would nest a VAR(1) model. The concatenate operation before the final Dense(6), which is a set of linear parameters, multiplies both the output of the GRU alomg with the input. In this final layer, if the parameters evaluating the output of the GRU are 0, and the parameters learned for the input in the final concat agree with the VAR(1) parameters, this model nest a VAR(1). Thus one can initialize parameters in this way to give the GRU model a good starting forecast. Ultimately, our model comprises about 17,000 parameters which explain the comparative outperformance in our data-rich cross-sectional world dataset.

IV. Economic Models

We tested the predictive power of a series of machine learning and traditional macroeconomic models estimated on our panel of countries using our novel data pooling method. We found that the more complex the model, the more our data augmentation helped. The machine learning models tended to be more flexible, but even among economic models, the trend still held. Additionally, we provided comparisons to the Survey of Professional Forecasters (SPF)(Philadelphia Federal Reserve, 1968) median GDP forecast, which is seen as a proxy for state-of-the-art performance. While we forecast the most recent GDP vintage, we do not use older vintages for older data. While this may result in a minor apples-to-oranges comparison with SPF, it is difficult to update vintages for the panel of 50 countries, most of which have documentation written in other languages. A discussion of the Survey of Professional Forecasters and our attempt to evaluate their forecasts is contained in Appendix C. The baseline economic models we used are the AR(2) autoregressive model, a VAR(4)/VAR(1), a Bayesian VAR (BVAR) model (Litterman, 1981), a Factor model (Stock and Watson, 2002*a*), (Stock and Watson, 2002*b*) (Sims, 1980), and the Smets-Wouters 2007 DSGE model (Smets and Wouters, 2007),.

Discussing the models more thoroughly, the formula for the AR(2) model is:

$$GDP_t = \beta_0 + \beta_1 * GDP_{t-1} + \beta_2 * GDP_{t-2}$$

The AR(2) model is an autoregressive model for GDP with 2 lags.

The formula for the VAR(1) model is:

$$GDP_t = \beta_0 + \beta_1 * X_{t-1}$$

where X_{t-1} is a vector of data containing: GDP_{t-1}, $Consumption_{t-1}$, and $Unemployment_{t-1}$. The VAR(4) contains the GDP, Consumption and Unemployment for four time lags rather than one. The VAR(4) was chosen by cross-validation to be the best number of lags for out of sample prediction.

The BVAR was calculated with a Litterman prior (Litterman, 1981), where I regularized around 1 for the lagged GDP term and around zero for all other terms:

$$GDP_t = \beta_0 + \beta_1 * X_{t-1} + \lambda * ||\beta_1[0] - 1, \beta_1[\neq 0]||_i$$

The $||.||_i$ indicates the i norm. I use both $i = 1$ and $i = 2$ and test both Ridge and LASSO regressions.

The Factor Model approach takes a large cross-section of data and uses a technique like principal components analysis to reduce the dimensionality of the problem. In our case, we concatenate five to eight principal components based on the information criteria of the high dimensional data with a lagged value of GDP and regress future GDP. We modified and used the code from FRED-QD as our baseline Factor Model (McCracken and Ng, 2016). Here is the factor equation:

$$GDP_t = \beta_0 + \beta_1 * GDP_{t-1} + \sum_{i=2}^{7} \beta_i * f_{it-1}$$

Here the f_{it-1} are derived by using principal components on a large scale database of quarterly

covariates from FRED-QD. While factor models models were extremely effective at shorter horizons, they were also dependent on a large cross-section of economic data with a long history in a country. In reality, only a few other developed countries have a cross-section of data that would be large enough to permit using these models as effectively as can be used in the United States. A more detailed explanation of these models is contained in Appendix D.

As is standard with economic forecasting, the baseline models were trained in a pseudo-out-of-sample fashion where the training set/validation set expands as the forecast date becomes more recent. However, with our neural network and AutoML, we keep the training set and validation set fixed due to computational costs and programming constraints. We expect that our model will improve if we use a pseudo-out-of-sample approach.

V. Data and Method

Discussing how we tested each class of model – reduced-form, structural, and machine learning – we applied three datasets: a US-only dataset, a world-pooled dataset, and a world-pooled ex-US dataset. This approach extends Lyu, Nie and Yang (2021), from an analysis of cross-country pooling for factor models to an analysis across a wide range of economic models.

We split our data into training, validation, and test sets. The models would be estimated first on only the 200+ US-only timesteps from 1951Q1-2008Q3 containing both a training and validation set and evaluated on a US out-of-sample test set (2008Q4-2020Q1). Then the models would be estimated on the 2300+ data points from 1951Q1-2008Q3 across 50 countries and evaluated on the same US test set See Appendix A for the country list. For the US this data starts 1951Q1. For all the other countries their data starts after 1951Q1, at varying times. The validation consisted of data from 2003-Q4 to 2008-Q3, which was only used for the RNN, and for the LASSO selection. This data was in the training set for all other models. Again US data was the only validation data used.

For both the reduced form and structural models, we also tested the performance for the world pooled data set ex US, so the training data contained no data from the US even though we were

forecasting the same US GDP test set. Interestingly, this often improved forecasting and at worst was statistically indistinguishable from forecasting from the entire world dataset, suggesting the US in 2020 is much more similar to France in 2020, than the US in 1970, and using data outside the US is a substitute that results in little forecasting loss.

AutoML, which was not a sequential model, used k-fold cross-validation on the entire training set, comprised of the remainder of data excluding test or validation sets. Theory suggests issues with this, but empirical tests justify the effectiveness even if the iid assumption of k-fold validation on time series does not entirely hold (Bergmeir, Hyndman and Koo, 2018). Finally, we compare all models, all horizons, pooled and US-only data, and SPF data and have a horserace – demonstrating the best-performing models.

We chose these periods so that both the test set and the validation set would have periods of both expansion and recession based on the US business cycle. Including the 2001 recession in the validation set would leave the model without enough training data, so we split the data of the Great Recession over the test and validation set. The quarter that bore the brunt of the fall of Lehman Brothers and the largest dip in GDP was the first quarter in our test set, 2008-Q4. Two quarters with negative growth preceding this were in the validation set. We estimated all models from a horizon of one quarter ahead to five quarters ahead. The metric of choice for forecast evaluation was RMSE. Additionally, all RMSE evaluations, unless otherwise noted, are on test data after using both training and validation/cross-validation data.

The data used in the reduced form models was consumption, unemployment, and output data, attempting to forecast GDP. US data is US only data from these three covariates and World data comes from our entire cross-section of countries with the same three covariates. The out-of-sample data attempts to predict US GDP, both without any data across countries from 2008Q4 to 2020Q1, and also without any US data in any time-step. In some sense, this is both an out-of-sample from a time-step perspective as well as a country perspective. We use these tests to determine how applicable estimated parameters across countries hold up in a preliminary test of parameter invariance. Additional robustness checks for reduced-form models is contained in Section VI.A..

We sourced cross-country data from Trading Economics via the Quandl platform API [4] as well as GDP data from the World Bank.[5]. We used GDP, consumption, and the unemployment rate as inputs to the reduced form and machine learning models. GDP and consumption were all expressed in growth rates. Unemployment was expressed as a rate as well. We selected these covariates after using LASSO on the validation set to pick covariates most likely to be useful predictors. This data was used both for the reduced-form models and the machine-learning ones. Regarding structural models, since the Smets-Wouter model requires 12 different data series, we used a reduced set of 27 countries where we can find this data. Data came from the Federal Reserve Economic Data (FRED), the World Bank, Eurostat, the Organization for Economic Cooperation (OECD) and the International Monetary Fund (IMF). See the countries in Appendix A. In all cases including the sturtural model, we only analyze GDP forecasting performance, however.

The out-of-sample data was the world-pooled training data with the US removed. Since the main exercise is to forecast US GDP over the test set, this dataset both has no data on the country being forecasted nor the timesteps that are being forecasted.

VI. Results

TWe show the results by model type: Reduced Form, Structural and Machine Learning. Then we will put all models in a horserace along with the SPF.

VI.A. Reduced-Form Models

Figure III shows the forecasting performance of both VAR(1) or AR(2) models at a five forecast horizons: 1-quarter ahead (H1) to 5-quarters ahead (H5). The stars next to the models' name indicate the statistical significance of world data outperformance over the US data using a Diebold-Mariano test (Diebold and Mariano, 2002) at the 1% (***), 5% (**) or 10% (*) level. This format will be followed throughout. Refer to Section V. for a description of the pooled and US data sets

4. https://www.quandl.com/tools/api
5. World Development Indicators, The World Bank

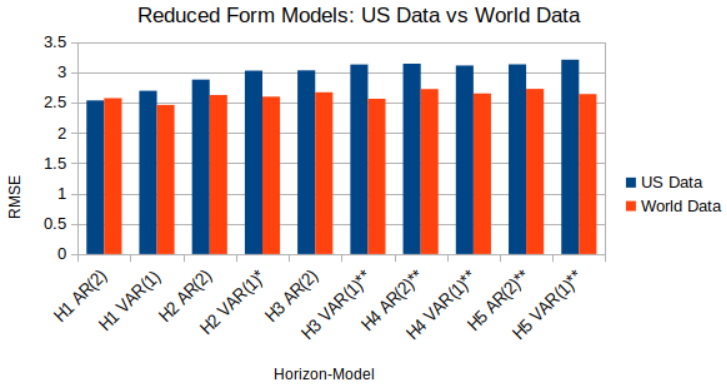

FIGURE III: FORECAST US GDP RMSE FOR LINEAR MODELS (AR(2) AND VAR(1)) USING BOTH US ONLY DATA AND POOLED WORLD PANEL DATA FOR FIVE TIME HORIZONS.

and Section IV. for a description of the models.

Pooling improved the performance of the models in a statistically significant manner, especially at longer horizons. Except for a slight underperformance at one quarter ahead for the AR(2), all other horizon models show outperformance using the country panel data augmentation. The outperformance of the pooled data averages roughly 12% of US RMSE over all horizon-model pairs.

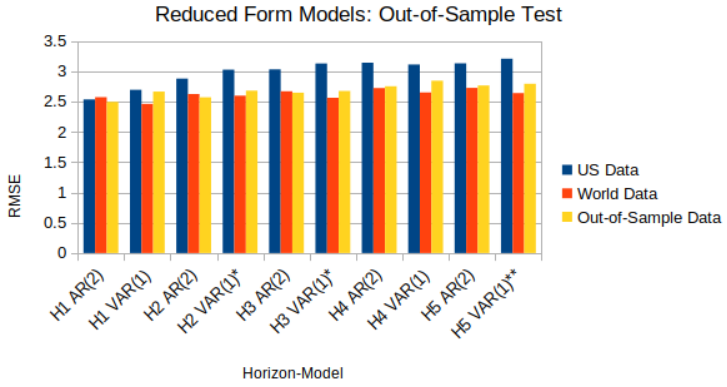

FIGURE IV: FORECAST US GDP RMSE FOR LINEAR MODELS USING US ONLY DATA, POOLED WORLD PANEL DATA, AND US FORECAST WITH WORLD DATA EXCLUDING US DATA

Figure IV shows similar data to Figure III, except the third bar for each horizon-model triplet

shows the RMSE of a model forecasting US GDP that has neither time series data after 2008Q4, nor US data over any period. This test enables us to show that using panel data can lead to models that are policy/country invariant and can generalize even to country data that the model lacked access to. This is labeled the Out-of-Sample data. In all cases, the RMSE of the world ex-US data forecasts was statistically indistinguishable from the RMSE of the model estimated on the full world panel of countries, including the US. Thus we use stars to indicate statistically significant out-of-sample forecast improvement over the US only data baseline. Excluding the H1 AR(2) pair and using only the out-of-sample data led to capturing 79%, on average, of the outperformance of the world panel over US-only data forecasts. It is interesting and suggestive that removing US data, doesn't statistically decrease the accuracy of forecast and suggests that US data is more similar than not with other countries when it comes to forecasting GDP. We explore this country out-of-sample test in Section VI.B. that discusses pooling for structural models and the Appendix (Section **??**)

VI.B. Structural Models

The performance of our structural models demonstrates that this pooling of data likely leads to performance gains across models, including DSGE models that should generalize to out-of-sample data because of their resilience to the Lucas critique. Combined with the results using machine learning models, our results make the case that model outperformance due to pooling helps the ability to generalize and forecast generally across the structural models we tested.

Figure V shows the improved performance moving from US-only data to cross-sectional global data:

The Smets-Wouters parameters that seemed to change the most – moving from US-only data to world data – were the shocks and the moving averages of the shock variable, monetary policy Taylor-rule variables, and variables governing wages and inflation. While the increasing variance of the shocks did not affect expected forecasts due to certainty equivalence, the model is both less confident and closer to correct when using pooled world data. Perhaps it is unsurprising that variables focusing on monetary policy and inflation are different when estimated on world data. Inflation, especially among rich developing countries, along with the monetary response to them,

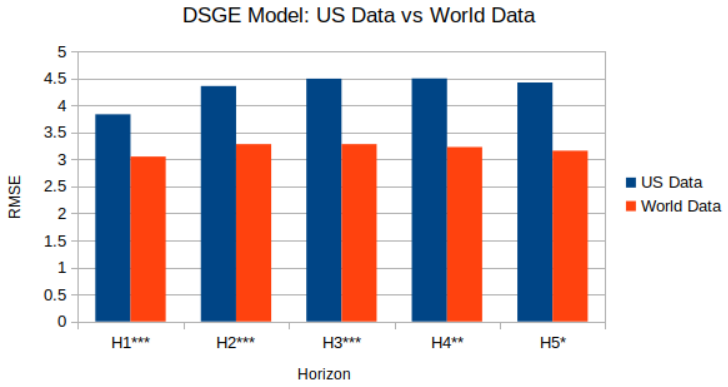

FIGURE V: FORECAST US GDP RMSE FOR DSGE MODELS USING BOTH US ONLY DATA AND POOLED WORLD PANEL DATA FOR FIVE TIME HORIZONS.

was a more pernicious problem outside the US than within (Azam and Khan, 2020). For more information on the changes in structural variables when moving from US data to pooled world data, see Appendix H.6.

Despite the increased uncertainty of the model as illustrated by the increase in the standard deviation of the shocks, the parameters were more reliable when estimated with world data. The improvement in RMSE averages over 25% over all horizons, more than double the percentage improvement for reduced-form models. Part of the outperformance was due to the weaker performance of the models estimated on US data. This suggests that the Smets-Wouters model is no better at generalizing across policy regimes or countries than reduced-form models and benefits more, by generalizing better, because of its higher parameter count.

In Figure VI, we also provide a structural chart that parallels the out-of-sample chart in Figure IV. As mentioned in the methods, this removes US data from the world cross-sectional data set but still attempts to forecast US GDP. This doubly out-of-sample data led to 9% better performance on average than the performance estimated on the entire world data. However, the Diebold-Mariano tests are less significant with only the first two horizons having p-values with less than 1% and no significance in horizons four and five. This is interesting as although the world ex-US data set has fewer data points, it seems to lead to better predictions of US GDP. As the US is the only

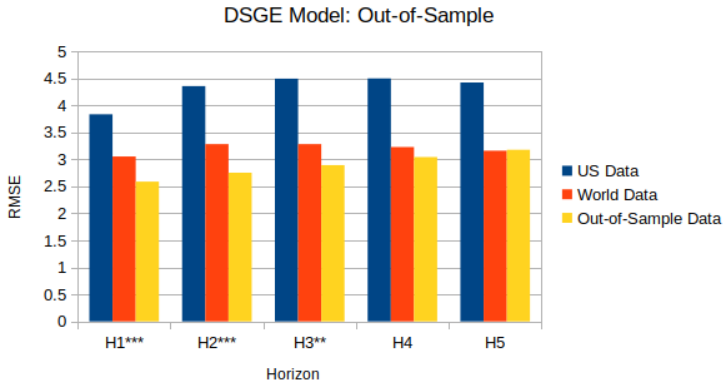

FIGURE VI: FORECAST US GDP RMSE FOR DSGE MODELS USING US ONLY DATA, POOLED WORLD PANEL DATA, AND US FORECAST WITH WORLD DATA EXCLUDING US DATA

country whose data extends to the early 1950s, dropping data further back in time may improve performance. While more data is generally better, it seems like more cross-sectional data of more recent vintage improves forecast more, but more data further in time, even for the forecasting country of interest, could hurt forecasting performance. The US in 2020 may be more similar to France in 2020, than the US in 1970. This could be due to the rise of the internet among other changes in economies. Thus, in contrast to Pesaran and Smith (1995), DSGE parameters could be more stable when using data across space than time. We examine this further in Appendix H.2.

VI.C. Nonparametric Machine Learning Models

Given the forecasting improvement for both the reduced-form and structural models and the improving relative performance of complex models, we decided to test the performance of nonparametric models that are even more flexible than the DSGEs and some of the larger VARs, using both the US-only data and the cross-sectional world pooled data. We tested an RNN, as well as the AutoML algorithm. While the performance improvement was less than the DSGE improvement from pooled data, it still seems impressive given that the flexible models had much better baseline performance even on US-only data. This again illustrates the trend that increased parameter count leads to synergistic performance gains from pooling.

The two charts below illustrate the performance of the RNN and AutoML models on both US and pooled world data.

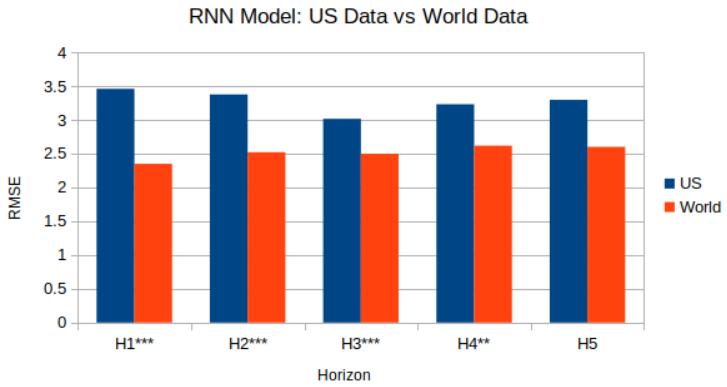

FIGURE VII: FORECAST US GDP RMSE FOR RNN MODELS USING BOTH US ONLY DATA AND POOLED WORLD PANEL DATA FOR FIVE TIME HORIZONS.

The RNNs GDP forecasting improvement from US-only to World data in Figure VII was statistically significant for all horizons except five quarters ahead. The average improvement was around 23% over all horizons, which was similar to the improvement for the Smets-Wouters model and almost double the improvement of linear models. This is a reassuring confirmation as the RNN is a data-hungry model that benefits more from data-rich regimes. We also attempted to add a country identifier term to our model. For example, we used GDP per capita at the time of prediction as an input to localize the pooled data to some degree. While this might be expected to reduce bias, it didn't improve out-of-sample performance to any degree. This suggests that countries are more similar than different and the bias of pooling different countries has a limited negative effect, while adding in such a covariate leads to a greater risk of overfitting.

Figure VIII shows the same performance graph for AutoML. The performance gain is not as easily interpreted as AutoML benefits from the pooled data but can also pick different models that gain relatively in both data-poor and data-rich regimes. Because of that, the large gains of the RNN are more representative of performance gains from moving to pooled data on fixed Machine Learning models. It has the least improvement in performance when using the panel of countries

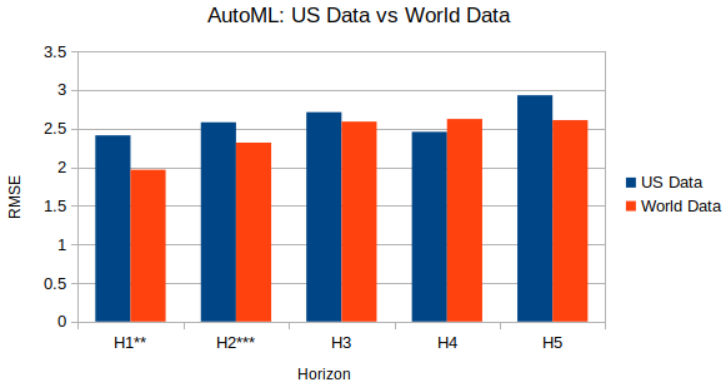

FIGURE VIII: FORECAST US GDP RMSE FOR AUTOML MODELS USING BOTH US ONLY DATA AND POOLED WORLD PANEL DATA FOR FIVE TIME HORIZONS.

as training data, with average improvements in RMSE of about 7.5%. When estimated on world data, AutoML outperforms all Economic models on all horizons except three quarters ahead.

VI.D. Summary

The previous sections outlined the performance of all reduced-form, structural, and machine learning models. This section takes all the data and provides results from a holistic perspective. We first compare forecasting models using all approaches, estimated on both pooled and US data. Table **??** demonstrates the effectiveness of the machine learning forecasting methods in data-rich regimes. We do not report the maximum likelihood of the Smets-Wouters model, as the original Bayesian parameterization has better performance than either of our maximum likelihood Smets-Wouters models estimated on world or US data. Introducing our other DSGE variations would be difficult to justify and would also have no effect on the results of the horse race. Regardless, all of our models that outperform the baseline models on a horizon are bolded. The best-performing model along all horizons was either an AutoML model or an RNN model, likely because the additional pooled data allowed a more powerful model to be used without overfitting.

For an additional discussion on how cross-sectional world pooled data and US data differentially affect the model based on size, see Appendix H.3.

TABLE I: US GDP RMSE Horserace of the RNN, AutoML, and Baseline Economic Models on Both US Data and World Data

Time (Q's Ahead)	1Q	2Q	3Q	4Q	5Q
VAR(4)					
US Data	2.99	3.03	3.10	3.08	3.08
World Data	2.37	2.52	2.56	2.63	2.63
AR(2)					
US Data	2.53	2.88	3.03	3.14	3.13
World Data	2.57	2.62	2.67	2.72	2.72
Smets-Wouters DSGE Bayesian					
US Data	2.79	2.95	2.89	2.80	2.71
Factor					
US Data	2.24	2.48	2.50	2.67	2.86
RNN (Ours)					
US Data	3.46	3.37	3.01	3.23	3.30
World Data	2.35	2.52	**2.50**	**2.62**	**2.60**
AutoML (Ours)					
US Data	2.41	2.58	2.71	**2.45**	2.92
World Data	**1.97**	**2.32**	2.59	**2.62**	**2.61**
SPF Median	1.86	2.11	2.36	2.46	2.65

VII. Conclusion

In this paper, we show how estimating Macroeconomic models on a panel of countries, as opposed to a single country, can significantly improve forecasting. Using a panel of countries as a training set, we statistically improved the RMSE performance of reduced-form models – AR(2), VAR(1), and VAR(4) – by 12% on average. We further show that we can make these reduced-form models more policy/country invariant, suggesting that these models can learn to generalize GDP forecasting even to countries the model has never been trained on.

We also showed that a similar training set of a panel of countries can improve the external validity of structural models which again are typically estimated only on a single country of interest. We focus on the Smets-Wouters model (Smets and Wouters, 2007). Using a panel of countries improves the forecasting performance of the Smets-Wouters model estimated with maximum likelihood by roughly 24% averaged across horizons. These results are again statistically significant for most time horizons. We then demonstrated that we can again improve policy invariance and generalization to out-of-sample countries by using a panel of countries in our training set. Additionally, we addressed

one potential roadblock to the adoption of pooling country data, which is the fact that the structural parameters may not be stable across countries and hence the pooled parameter value can only be interpreted as a mean value. While our results are less conclusive on this front, we argue that based on forecasting exercises, parameter generalization and stability are likely as good across space as across time. Finally, concluding our section on structural models, we capitalize on the consistency of improvements and discuss the likelihood that our results will extend to other estimation techniques like the generalized method of moments, calibration, and Bayesian approaches.

Our last set of results recognizes that our dataset has increased from 300 timesteps to around 3000 timestep countries, showing that nonparametric Machine Learning models are able to outperform all the Economic baseline models even after being estimated in this more data-rich regime. Our RNN outperforms all Economic baselines for horizons longer than two periods ahead. Likewise, our AutoML model outperforms all baselines for all horizons except for the three quarters ahead. Combined, the best-performing model over all horizons is either an AutoML or an RNN model, which suggests there is likely much more room to test other nonparametric models in the more data-rich Macroeconomic regime.

References

Adjemian, Stéphane, Houtan Bastani, Michel Juillard, Fréderic Karamé, Junior Maih, Ferhat Mihoubi, Willi Mutschler, George Perendia, Johannes Pfeifer, Marco Ratto, and Sébastien Villemot. 2011. "Dynare: Reference Manual Version 4." CEPREMAP Dynare Working Papers 1.

Agarap, Abien Fred. 2018. "Deep Learning using Rectified Linear Units (ReLU)." *CoRR*, abs/1803.08375.

Azam, Muhammad, and Saleem Khan. 2020. "Threshold effects in the relationship between inflation and economic growth: Further empirical evidence from the developed and developing world." *International Journal of Finance & Economics*.

Bai, Jushan, and Serena Ng. 2010. "Instrumental variable estimation in a data rich environment." *Econometric Theory*, 1577–1606.

Bergmeir, Christoph, Rob J Hyndman, and Bonsoo Koo. 2018. "A note on the validity of cross-validation for evaluating autoregressive time series prediction." *Computational Statistics & Data Analysis*, 120: 70–83.

Breitung, Joerg. 2015. "The analysis of macroeconomic panel data." In *The Oxford Handbook of Panel Data*.

Cho, KyungHyun, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. "On the Properties of Neural Machine Translation: Encoder-Decoder Approaches." *CoRR*, abs/1409.1259.

Chung, Junyoung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. "Empirical evaluation of gated recurrent neural networks on sequence modeling." *arXiv preprint arXiv:1412.3555*.

Diebold, Francis X. 1998. *Elements of forecasting.* South-Western College Pub.

Diebold, Francis X, and Robert S Mariano. 2002. "Comparing predictive accuracy." *Journal of Business & economic statistics*, 20(1): 134–144.

Diebold, Francis X, Glenn D Rudebusch, and S Boragan Aruoba. 2006. "The macroeconomy and the yield curve: a dynamic latent factor approach." *Journal of econometrics*, 131(1-2): 309–338.

Doran, Howard E, and Peter Schmidt. 2006. "GMM estimators with improved finite sample properties using principal components of the weighting matrix, with an application to the dynamic panel data model." *Journal of econometrics*, 133(1): 387–409.

Edge, Rochelle M, Michael T Kiley, and Jean-Philippe Laforte. 2010. "A comparison of forecast performance between federal reserve staff forecasts, simple reduced-form models, and a DSGE model." *Journal of Applied Econometrics*, 25(4): 720–754.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. "Deep Residual Learning for Image Recognition."

Hochreiter, Sepp, and Jürgen Schmidhuber. 1997*a*. "Long short-term memory." *Neural computation*, 9(8): 1735–1780.

Hochreiter, Sepp, and Jürgen Schmidhuber. 1997*b*. "Long Short-Term Memory." *Neural Comput.*, 9(8): 1735–1780.

Hornik, Kurt, Maxwell Stinchcombe, and Halbert White. 1989. "Multilayer feedforward networks are universal approximators." *Neural Networks*, 2(5): 359 – 366.

Hutter, F, R Caruana, R Bardenet, M Bilenko, Guyon, B B Kegl, and H.Larochelle. 2014. "AutoML 2014 @ ICML."

Ioffe, Sergey, and Christian Szegedy. 2015. "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift."

Kingma, Diederik P., and Jimmy Ba. 2014. "Adam: A Method for Stochastic Optimization."

Koop, Gary, Roberto Leon-Gonzalez, and Rodney Strachan. 2012. "Bayesian model averaging in the instrumental variable regression model." *Journal of Econometrics*, 171(2): 237–250.

Litterman, Robert B. 1981. "Techniques of forecasting using vector autoregressions."

Lyu, Yifei, Jun Nie, and Shu-Kuei X Yang. 2021. "Forecasting US economic growth in downturns using cross-country data." *Economics letters*, 198: 109668.

McCracken, Michael W, and Serena Ng. 2016. "FRED-MD: A monthly database for macroeconomic research." *Journal of Business & Economic Statistics*, 34(4): 574–589.

Pesaran, M Hashem, and Allan Timmermann. 1992. "A simple nonparametric test of predictive performance." *Journal of Business & Economic Statistics*, 10(4): 461–465.

Pesaran, M Hashem, and Ron Smith. 1995. "Estimating long-run relationships from dynamic heterogeneous panels." *Journal of econometrics*, 68(1): 79–113.

Philadelphia Federal Reserve. 1968. "Survey of Professional Forecasters." *https://www.philadelphiafed.org/research-and-data/real-time-center/survey-of-professional-forecasters*, Accessed: 2020-09-01.

Polyak, Boris. 1964. "Some methods of speeding up the convergence of iteration methods." *Ussr Computational Mathematics and Mathematical Physics*, 4: 1–17.

Sims, Christopher A. 1980. "Macroeconomics and Reality." *Econometrica*, 48(1): 1–48.

Smets, Frank, and Rafael Wouters. 2007. "Shocks and frictions in US business cycles: A Bayesian DSGE approach." *The American Economic Review*, 97(3): 586–606.

Stock, James H, and Mark W Watson. 2002*a*. "Forecasting using principal components from a large number of predictors." *Journal of the American statistical association*, 97(460): 1167–1179.

Stock, James H, and Mark W Watson. 2002*b*. "Macroeconomic forecasting using diffusion indexes." *Journal of Business & Economic Statistics*, 20(2): 147–162.

Tieleman, T., and G. Hinton. 2012. "Lecture 6.5—RmsProp: Divide the gradient by a running average of its recent magnitude." *COURSERA: Neural Networks for Machine Learning.*

Timmermann, Allan. 2006. "Forecast combinations." *Handbook of economic forecasting*, 1: 135–196.

West, Kenneth D. 1996. "Asymptotic inference about predictive ability." *Econometrica: Journal of the Econometric Society*, 1067–1084.

Wright, Jonathan H. 2008. "Bayesian model averaging and exchange rate forecasts." *Journal of Econometrics*, 146(2): 329–341.

Zerroug, A, L Terrissa, and A Faure. 2013. "Chaotic dynamical behavior of recurrent neural network." *Annu. Rev. Chaos Theory Bifurc. Dyn. Syst*, 4: 55–66.

A Appendix

A Selected Countries

Countries in reduced-form data set: Australia, Austria, Belgium, Brazil, Canada, Switzerland, Chile, Columbia, Cyprus, Czech Republic, Germany, Denmark, Spain, Estonia, European Union, Finland, France, Great Britain, Greece, Hong Kong, Croatia, Hungry, Ireland, Israel, Italy, Japan, Korea, Luxembourg, Latvia, Mexico, Mauritius, Malaysia, Netherlands, Norway, New Zealand, Peru, Philippines, Poland, Portugal, Romania, Russia, Singapore, Slovakia, Slovenia, Sweden, Thailand, Turkey, USA and South Africa.

Countries in structural data set: Australia, Austria, Belgium, Canada, Chile, Columbia, Germany, Denmark, Spain, Estonia, Finland, France, Iceland, Israel, Italy, Japan, Korea, Lithuania, Luxembourg, Mexico, Netherlands, New Zealand, Poland, Portugal, Slovakia, Slovenia, Sweden, USA

B Selected Performance: Graphs

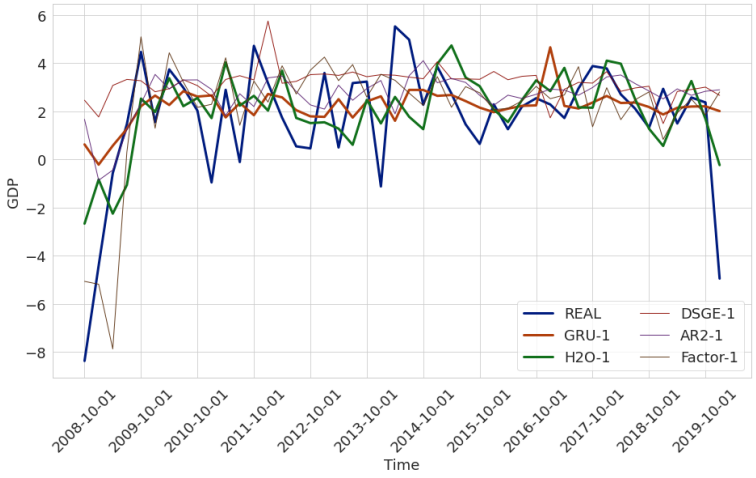

Figure IX: One Quarter Ahead - US GDP Forecasts

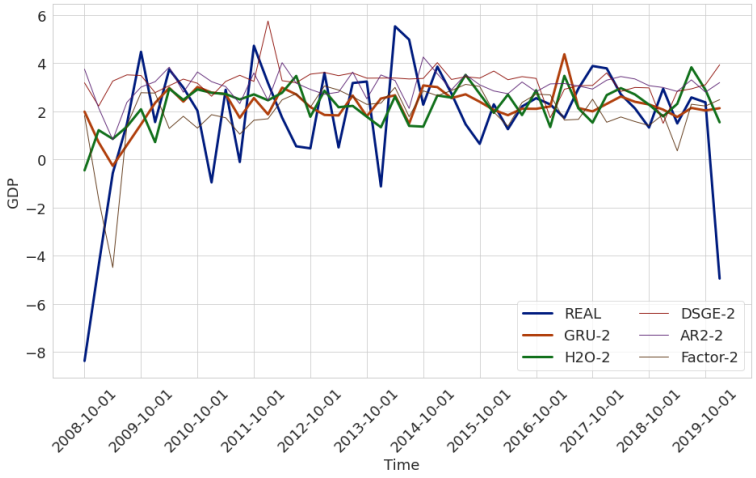

Figure X: Two Quarters Ahead - US GDP Forecasts

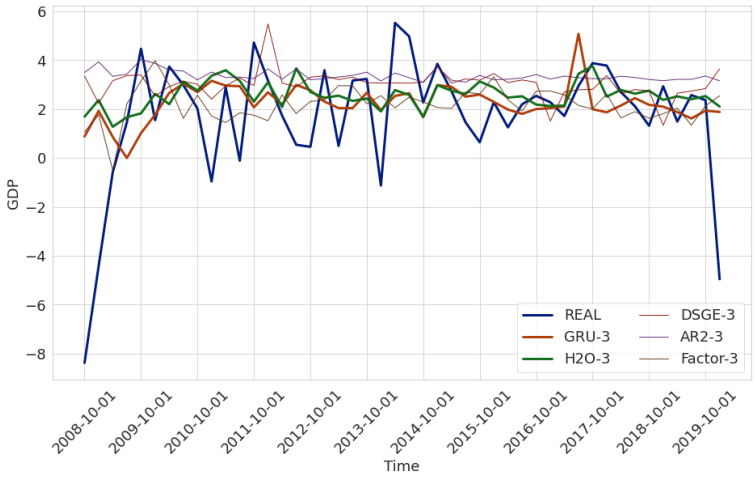

Figure XI: Three Quarters Ahead - US GDP Forecasts

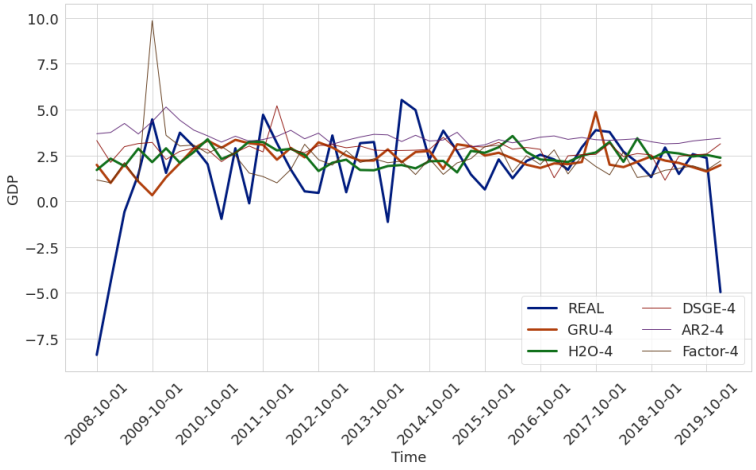

Figure XII: Four Quarters Ahead - US GDP Forecasts

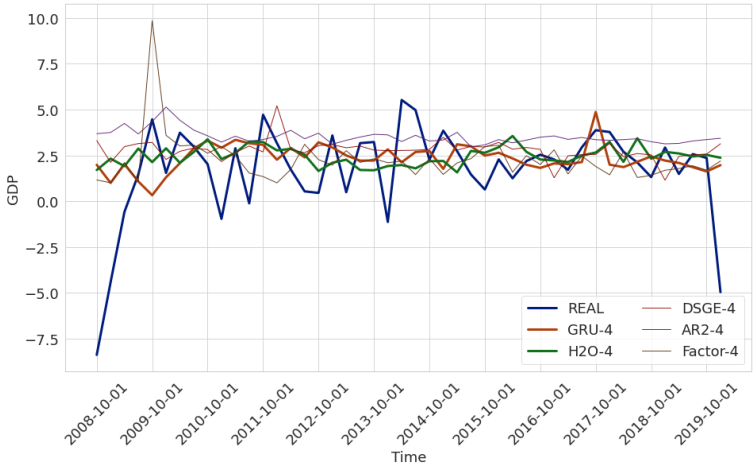

Figure XIII: Five Quarters Ahead - US GDP Forecasts

C Details on the Survey of Professional Forecasters (SPF)

While our model used the 2020 vintage data, in reality, the forecasters for the Survey of Professional Forecasters were working with pseudo-out-of-sample vintages when forecasting over the entire test. While reproducing this would be possible by using old vintages, it would require estimating the model at every time step of the test set as the data would change every period. We wanted to avoid this pseudo-out-of-sample forecasting as it would result in estimating 4600 models instead of 100 at each horizon. Beyond this, the benefit of this increased computation was not clear as we would still be using 2020 vintage data for countries outside the US, for which old vintages are difficult to find. So, it was easier to compare the SPF performance on the 2020 vintage. In addition, all our baseline models including world forecasts were estimated and evaluated using the 2020 vintage as well. This choice allowed us to compare the SPF performance with the performance of all the baseline models.

D Detailed Description of Economic Baseline Models

The first model we use is the autoregressive model, AR(n). An oft-used benchmark model, it estimates a linear relationship using the independent variable lagged N times. In terms of forecasting ability, this model is competitive with or outperforms the other Economic models in our tests which is consistent with Diebold (1998). We used an autoregressive model with two lags and a constant term.

Additionally, we compared the Smets-Wouters 2007 model (Smets and Wouters, 2007), as DSGE models share many similarities with Recurrent Neural Networks and Smets-Wouters (2007) suggests that this particular model can outperform VARs and BVARs in forecasting. When running this, we used the standard Smets-Wouters Dynare code contained in the published paper's data appendix. We take the point forecasts from the Smets and Wouters (2007) and use that to forecast. Like Smets and Wouters (2007), we use Dynare (Adjemian et al., 2011) to solve and estimate the model.

A final model we included in our baseline Economic models was the Factor Model (see Stock and Watson (2002a) and Stock and Watson (2002b)). In short, the Factor Model approach takes

a large cross-section of data and uses a technique like principal components analysis to reduce the dimensionality of the problem. In our case, we concatenate five to eight principal components based on the information criteria of the high dimensional data with a lagged value of GDP and regress future GDP. We modified and used the code from FRED-QD as our baseline Factor Model (McCracken and Ng, 2016). While these models were extremely effective at shorter horizons, they were also dependent on a large cross-section of economic data with a long history in a country. In reality, only a few other developed countries have a cross-section of data that would be large enough to permit using these models as effectively as can be used in the United States. That being said, factor models do outperform our neural networks at shorter time intervals, and we imagine there is promise in combining the factor approach with a RNN or AutoML approach.

We also tested the the forecasting performance of vector autoregressions (Sims, 1980). In addition to displaying performance in our main table, we compared this model and the AR(2) in our 50 countries cross-section test as well. Since we were only forecasting GDP, the vector autoregressive models used lagged GDP, consumption, and unemployment to forecast the single GDP variable.

All the economic models were estimated on US GDP as is standard. While we ran preliminary tests on estimating these models on our cross-section of 50 countries, we ran into issues with estimating both Factor Models and DSGE models this way. However, preliminary results on the AR(2) model suggest there could be some improvement to using a cross-section even on a three-parameter AR(2) model. The improvement is not as large as the RNN, which is not surprising as the RNN has more parameters to take advantage of a larger data set.

E The Rectified Linear Unit

A nonlinearity used in our architecture, but not in the GRU layers, is the rectified linear unit (ReLU) (Agarap, 2018). The rectified linear unit is defined as:

$$ReLU(x) = max(0, x) \tag{20}$$

The ReLU is the identity operation with a floor of zero much like the payoff of a call option. Despite being almost the identity map, this nonlinearity applied in a wide enough neural network can approximate any function (Hornik, Stinchcombe and White, 1989).

F Skip Connections and Batch Norm

Skip connections (He et al., 2015) allow the input to skip the operation in a given layer. The input is then just added onto the output of the skipped layer, forming the final output of the layer. This allows the layer being skipped to learn a difference between the "correct" output and input instead of transforming the input to output. Additionally, if the model is overfitting, the neural network can learn the identity map easily. Skip connections are used when the input and the output are the same dimension which allows each input to correspond to one output. Because our network does not have this property, we learn a linear matrix that converts the input to the output dimension. All the skip connections are linear operations and have no activation or batch norm, which differs from the pair of dense layers at the beginning of the network, which have both batch norm and rectified linear unit activations.

Batch normalizing (Ioffe and Szegedy, 2015) is used to prevent the drift of output through a deep neural network. Changes to parameters in the early layers will cause an outsized effect on the output values for the later layers. Batch norm fixes this problem by normalizing the output to look like a standard normal distribution after the output of each layer. Thus, the effect of changes in parameters will not greatly affect the magnitude of the output vector because, between each batch, the data is re-normalized to have a mean of 0 and a standard deviation on 1.

G Adam Optimizer

Adam (Kingma and Ba, 2014) combines momentum (Polyak, 1964), a technique that uses recent history to smooth out swings orthonormal to the objective direction, with RMSprop (Tieleman and Hinton, 2012), a technique used to adjust step size based on gradient volatility.

Traditional gradient descent hill climbing updates the parameters with a single equation:

$$\theta_t = \theta_{t-1} - \lambda * \nabla_\theta L_\theta(x, y) \tag{21}$$

Here $\nabla_\theta L_\theta(x, y)$ denotes taking the gradient of the loss with respect to θ, the parameters of the model. For convenience, I will denote this term g_t. By subtracting the gradient multiplied by a small step size, λ, one moves the parameters, θ, in the direction that reduces the loss the most.

If we wanted to use information from the second derivative to inform optimization, we can use Newton-Raphson instead:

$$\theta_t = \theta_{t-1} - H_t^{-1} * g_t \tag{22}$$

This uses the Hessian to determine an optimal step size based on steepness in the loss function. Typically, this approach is not used in deep learning as deep learning models typically have a large number of parameters, and calculating the Hessian has a quadratic cost in the number of parameters and inverting also has a super-linear cost. However, there are quasi-Newton methods that attempt to approximate the Hessian to determine the step size without the high computational cost. Adam is similar to these methods. The equations that define Adam are as follows:

$$\nu_t = \beta_1 * \nu_{t-1} - (1 - \beta_1) g_t \tag{23}$$

$$s_t = \beta_2 * s_{t-1} - (1 - \beta_2) * g_t^2 \tag{24}$$

$$\delta\theta_t = -\eta \frac{\nu_t}{\sqrt{s_t + \epsilon}} * g_t \tag{25}$$

$$\theta_{t+1} = \theta_t + \delta\theta_t \tag{26}$$

The first equation is a moving average of the gradient. This "momentum" term is used because often in training the direction of the gradient would move nearly perpendicular to the direction of the optimum. Gradient descent would spend a lot of time zig-zagging while only making slow

progress towards an optimum (see Figure XIV). Taking a moving average of previous gradients preserves the principal direction while the orthogonal directions cancel each other out.

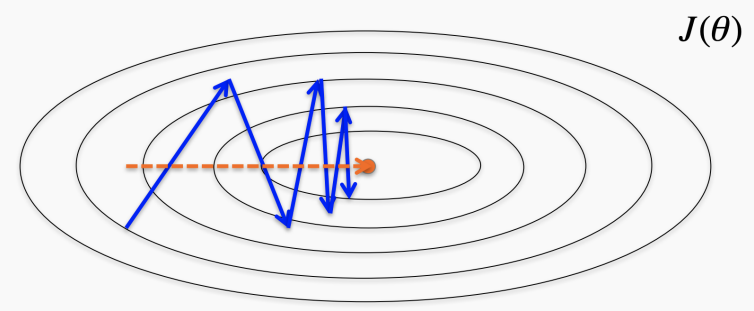

FIGURE XIV: MOMENTUM

Likewise, the s_t equation is a moving average approximation for the Hessian (the second derivative). The approximate Hessian is used for adjusting the step size of the algorithm based on the curvature of the loss function at a given point. β_1 and β_2 are hyperparameters – parameters that have to be adjusted in a validation set rather than by derivative-based optimization – that determine the smoothness of the moving average. Again, the resulting update term is applied to the previous values of the parameters. This approach is empirically shown to lead to more stable optimization and even better optima than simpler gradient descent approaches for large networks.

H Additional Forecasting Information

H.1 Linear Models GDP Forecast Performance Cross-Sectionally Tested Across Countries Over the Entire World

We performed the same tests over our entire cross-section of countries, shown in Table H.1 For example, for each local forecast, we used only French data to forecast French GDP. With world data, we estimate a single model with all the data and use it to forecast every country and average the RMSE. For the out-of-sample data, we estimate a new model for each country that takes every country's data into account except for the country whose GDP data we are forecasting.

This data is analogous to the world data ex-US in the other exercises and is both time-step out-

TABLE II: Average Linear Models GDP Forecasting Performance (RMSE) Evaluated over US, the World, and Out-of-Sample Data With a GDP Test Set Extending Over The Entire World

Time (Q's Ahead)	1Q	2Q	3Q	4Q	5Q
AR(2)					
Local Data	5.22	5.33	5.51	5.56	5.60
World Data	4.88	4.98	5.10	5.19	5.19
Out-of-Sample Data	5.07	5.13	5.36	5.36	5.38
VAR(1)					
Local Data	5.10	5.17	5.19	5.29	5.26
World Data	4.80	4.94	5.04	5.11	5.12
Out-of-Sample Data	4.92	5.00	5.07	5.20	5.25
VAR(4)					
Local Data	7.90	7.05	7.74	7.87	9.27
World Data	4.72	4.90	5.03	5.10	5.11
Out-of-Sample Data	4.70	4.87	5.02	5.17	5.26

of-sample and country out-of-sample, so we call this data Out-of-Sample Data. This table provides a strong robustness check for the idea that pooling data leads to better forecasts across the board with more flexible models having much worse single-country results, but much better-pooled results. However, large model pooled results outperform all models here.

Additionally, flexible models like the VAR(4) can predict a country's GDP with indistinguishable levels of accuracy whether that country's data is included or excluded from the pooled world data set. This plays into the idea countries are more similar across space, than going into a country's deep past time-wise. The correlation is somewhat nuanced, as only more flexible models seem to be able to effectively adjust using world data as a proxy for the individual country data that is being forecasted. This is evidence, that different countries are more similar than different when it comes to the relationship between past and feature GDP forecasts and the same or very similar parameters uniformly govern most countries' data-generating process.

H.2 Removing Time-steps from Pooled DSGE Data

Probing the hypothesis that countries are more similar across time than than across space (same country but long time lag between two pieces of time-series data) led to somewhat mixed results.

We estimated a model trained on the entire panel of countries with data post 1995-Q1 onward. This affected three countries – the US, Japan, and New Zealand. This procedure isolates more sharply the effect of similarity across space versus across time on model generalization, rather than just removing all US data. The US lost about 140-190 data points (as the test set requires rolling forecasts), while New Zealand and Japan both lost about 15-60 timesteps. Figure XV, illustrates this experiment to compare the performance of models estimated on data since 1995 to models estimated with full country and timesteps.

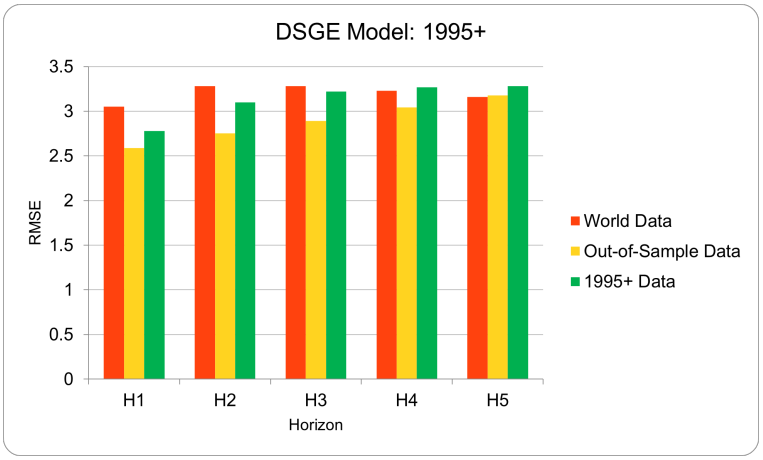

Figure XV: Compares Parameter Stability via US GDP Forecasts Going Back in Time (Out of Sample Datatset) Versus Space (1995+ Dataset)

The 1995 onwards data performed worse than the out-of-sample test which could suggest some of the outperformance of the out-of-sample DSGE model was due to chance. However, it seems that the 1995 data performed at least as well as a model estimated on world data both pre and post-1995, despite our robust results suggesting that more data is generally better. This makes sense when considering the advent of software, for example. The results seem inconclusive but certainly don't suggest any more parameter stability across time than space, in contrast to Pesaran and Smith (1995) and generally accepted in the literature.

H.3 Pooled Data and Model Size

To illustrate the effect that pooling data has on forecasting, we show a graph that orders RMSE performance based on increasing model complexity with RMSE performance, comparing the trend when estimated on US data versus pooled data. Even though the RNN nests the VAR(1), it underperforms with US data, because it overfits the training data and performs worse on the test set.

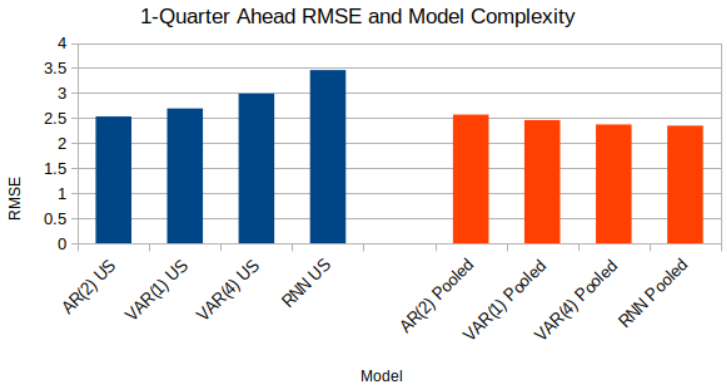

FIGURE XVI: AS MODEL COMPLEXITY INCREASES, US GDP FORECAST RMSE PERFORMANCE WORSENS USING ONLY US DATA. HOWEVER, THIS TREND REVERSES USING WORLD DATA

Even if the RMSE decline in Figure XVI is less striking in pooled data, this decline is nevertheless compelling. In fact, despite the appearance of only a small improvement due to model complexity, the performance of the RNN on pooled data is state-of-the-art, while the performance of the AR(2) on pooled data is pedestrian. The smaller improvements on pooled data are because even the AR(2) on pooled data is already an accurate model. A similar story holds across other horizons with less striking consistency compared to the one period ahead story.

H.4 Information Content Regressions

We regress true GDP on a varying collection of forecasts to test for statistically significant contribution of a given forecast like our gated recurrent unit model. For example, Table III, attempts to predict GDP from both the SPF forecasts and AutoML-H2O forecasts. The coefficients don't

mean much, but the statistical significance of each coefficient indicates that the given forecast adds information on GDP forecasts above and beyond the other predictions. So SPF at one quarter ahead adds information significant at the 1% level. Likewise, AutoML-H20 is also statistically significant at that horizon, but no others.

Here is the AutoML-H2O forecast compared to the SPF on the baseline test set ranging from one-quarter ahead forecasts to five-quarters ahead:

TABLE III: AutoML-H20 Compared to SPF For Forecast from 1 to 5 quarters

	Real GDP Growth				
	(1-Qtr)	(2-Qtrs)	(3-Qtrs)	(4-Qtrs)	(5-Qtrs)
SPF	0.796***	1.554***	3.042***	2.888***	1.218
	(0.238)	(0.346)	(0.607)	(0.742)	(1.025)
H2O	0.505**	0.564	0.511	0.213	0.337
	(0.236)	(0.399)	(0.571)	(0.669)	(0.666)
N	46	46	46	46	46
R^2	0.532	0.474	0.409	0.271	0.033
Adjusted R^2	0.510	0.449	0.382	0.237	−0.012
Residual Std. Error (df = 43)	1.790	1.898	2.012	2.235	2.573
F Statistic (df = 2; 43)	24.464***	19.366***	14.879***	7.997***	0.743

Notes:

***Significant at the 1 percent level.
**Significant at the 5 percent level.
*Significant at the 10 percent level.

The following table contains regressions comparing the information content of the H2O and baselines, excluding the SPF:

TABLE IV: AutoML-H20, DSGE, AR(2), and Factor Models For Forecast from 1 to 5 quarters

	Real GDP Growth				
	(1-Qtr)	(2-Qtrs)	(3-Qtrs)	(4-Qtrs)	(5-Qtrs)
H2O	0.778***	1.385***	1.198*	0.665	0.151
	(0.259)	(0.427)	(0.691)	(0.779)	(0.684)
DSGE	0.724	−0.041	0.138	0.047	0.081
	(0.478)	(0.581)	(0.647)	(0.679)	(0.712)
AR2	−0.554	−1.102	−0.981	−0.797	−1.576
	(0.438)	(0.781)	(1.759)	(1.088)	(1.634)
Factor	0.358*	0.657*	0.898	0.501	0.506
	(0.188)	(0.354)	(0.539)	(0.323)	(0.463)
N	46	46	46	46	46
R^2	0.501	0.291	0.127	0.070	0.031
Adjusted R^2	0.453	0.222	0.041	−0.021	−0.063
Residual Std. Error (df = 41)	1.893	2.257	2.505	2.586	2.638
F Statistic (df = 4; 41)	10.307***	4.202***	1.485	0.767	0.329

Notes:

***Significant at the 1 percent level.
**Significant at the 5 percent level.
*Significant at the 10 percent level.

This final table performs the same regression but has the SPF, H2O, and all baseline models:

TABLE V: AutoML-H20, DSGE, AR(2), Factor Models, and SPF For Forecast from 1 to 5 quarters

	REAL				
	(1-Qtr)	(2-Qtrs)	(3-Qtrs)	(4-Qtrs)	(5-Qtrs)
H2O	0.614**	0.629	0.495	0.233	0.254
	(0.239)	(0.403)	(0.594)	(0.693)	(0.689)
SPF	1.071***	1.648***	3.085***	2.835***	1.256
	(0.326)	(0.394)	(0.693)	(0.776)	(1.145)
DSGE	0.544	−0.141	−0.220	−0.332	−0.123
	(0.433)	(0.492)	(0.542)	(0.605)	(0.734)
AR(2)	−1.029**	−0.981	0.776	−0.749	−1.672
	(0.420)	(0.660)	(1.509)	(0.954)	(1.632)
Factor	0.003	0.115	0.101	0.321	0.418
	(0.201)	(0.326)	(0.481)	(0.287)	(0.469)
N	46	46	46	46	46
R^2	0.607	0.506	0.416	0.302	0.059
Adjusted R^2	0.558	0.445	0.343	0.215	−0.058
Residual Std. Error (df = 40)	1.701	1.907	2.074	2.267	2.632
F Statistic (df = 5; 40)	12.363***	8.208***	5.697***	3.467**	0.505

Notes:
***Significant at the 1 percent level.
**Significant at the 5 percent level.
*Significant at the 10 percent level.

None of the models except the SPF have consistent statistically significant information above and beyond the other models.

H.5 Bias and Variance in the Forecasting Models

The following table contains the mean bias as well as the variance of the models. For the gated recurrent unit, we use the median forecast:

TABLE VI

	Forecast Bias				
	(1-Qtr)	(2-Qtrs)	(3-Qtrs)	(4-Qtrs)	(5-Qtrs)
GRU Bias	0.459	0.480	0.506	0.620	0.644
Variance	5.51	6.34	6.23	6.85	6.75
AutoML-H2O Bias	0.293	0.511	0.833	0.723	0.422
Variance	3.86	5.36	6.70	6.88	6.80
SPF Bias	0.331	0.600	0.723	0.804	0.901
Variance	3.48	4.46	5.57	6.07	7.04
DSGE Bias	1.75	1.93	1.88	1.78	1.65
Variance	9.32	10.77	10.76	10.42	9.99
AR2 Bias	0.404	0.389	0.431	0.472	0.481
Variance	6.61	6.88	7.12	7.40	7.41
VAR4 Bias	0.233	0.214	0.201	0.200	0.195
Variance	5.63	6.36	6.56	6.89	6.91
Factor Bias	0.432	0.163	0.459	0.533	0.699
Variance	5.03	6.17	6.26	7.12	8.19

H.6 Parameter Comparison across US DSGE Model and World DSGE Model

The table shows the parameters of the DSGE model estimated on world data versus US data as well as the standard deviations of the parameters over time. This illustrates the important parameters that change when adding in global data. In addition to the model growing less confident and more accurate when world data is added, the parameters that are most modified are parameters governing wage stickiness and inflation.

Dynare Variable	Variable Description	World Param Values	US Param Values	World Standard Deviation	US Standard Deviatio
'ea'	Factor Productivity Shock Error	0.976842	0.474861	0.034192	0.014886
'eb'	Risk Permium Shock Error	0.522046	0.260422	0.022821	0.115139
'eg'	Government Spending Shock Error	1.002773	0.659571	0.028303	0.017909
'eqs'	Technology Shock Error	1.377699	0.426465	0.114479	0.033718
'em'	Monetary Policy Shock Error	0.202341	0.209711	0.018569	0.005405
'epinf'	Inflation Shock Error	0.785736	0.184302	0.034779	0.019632
'ew'	Wage Shock Error	0.595469	0.316343	0.107472	0.023861
'crhoa'	AR Parameter on productivity Shock	0.997478	0.984393	0.002449	0.003019
'crhob'	AR Parameter on Risk Premium Shock	0.132395	0.383984	0.039344	0.332214
'crhog'	AR Parameter on Government Shock	0.984414	0.980403	0.009976	0.009258
'crhoqs'	AR Parameter on Technology Shock	0.688845	0.776188	0.109162	0.027836
'crhoms'	AR Parameter on Monetary Shock	0.30509	0.148434	0.02064	0.041207
crhopinf'	AR Parameter on Inflation Shock	0.611081	0.977832	0.032108	0.024071
'crhow'	AR Parameter on Wage Shock	0.982497	0.899716	0.010334	0.045943
'cmap'	AR Moving Average Error Term on Inflation	0.539008	0.875734	0.038097	0.058007
'cmaw'	AR Moving Average Error Term on Wages	0.970411	0.882725	0.011541	0.046395
'csadjcost'	Elasticity of the Capital Adjustment Cost	9.80101	8.15361	0.8643	1.03545
'csigma'	Elasticity of Subsitution	1.978862	1.895245	0.166713	0.253334
'chabb'	Habbit Formation	0.868449	0.663259	0.016705	0.173953
'cprobw'	Wage Flexibility Probability	0.930401	0.924096	0.028501	0.031956
'csigl'	Wage Elasticity of Labor Supply	1.749702	3.763315	0.471358	1.127552
'cprobp'	Price Flexibility Probability	0.945883	0.663557	0.005239	0.034912
'cindw'	Wage Indexation	0.01	0.635949	5.26E-18	0.14763
'cindp'	Indexation to Past Inflation	0.01	0.13723	5.26E-18	0.123999
'czcap'	Elasticity of Capital Utilization	0.364196	0.812247	0.194422	0.080674
'cfc'	Fixed Costs in Production	2.040895	1.93295	0.0107	0.141187
'crpi'	Taylor Rule Inflation	1	2.114895	0	0.321337
'crr'	Taylor Rule Interest Rate Smoothing	0.966891	0.912953	0.011906	0.016476
'cry'	Taylor Rule Output Gap	0.315889	0.118081	0.110856	0.063766
'crdy'	Taylor Rule Output Gap Change	0.024124	0.15335	0.006121	0.030106
constepinf'	Gap between Model and Observed Inflation	0.187268	1.240916	0.023902	0.2128
'constelab'	Gap between Model and Observed Labor	2.993805	0.640214	0.998992	1.963757
'ctrend'	Gap between Model and Observed Output	0.34744	0.426019	0.038315	0.026315
'cgy'	Productivity Shocks on Government Spending	0.480685	0.550493	0.058911	0.01411
'calfa'	Elasticity of Capital in Production Function	0.091021	0.212888	0.015633	0.02167

I Selected Additional Information

I.1 Smets-Wouters Model: US Data vs. World Data

TABLE VII: Smets-Wouters Forecast US GDP RMSE: US, World, Out-Of-Sampe, 1995+, and Bayesian Model Estimates

The performance forecasts for the Smets-Wouters model on the test set 2009-Q1 to 2020-Q1 (Lower is better)

Time (Q's Ahead)	1Q	2Q	3Q	4Q	5Q
Smets-Wouters DSGE Max Like					
US Data	3.83	4.36	4.49	4.50	4.2
World Data	3.05***	3.28***	3.28***	3.22**	3.16*
Out-Of-Sample Data	2.59***	2.75***	2.89**	3.04	3.18
1995+	2.77***	3.09**	3.22*	3.26	3.28
Smets-Wouters DSGE Bayesian					
US Data	2.79	2.95	2.89	2.80	2.71

* Significance indicates outperformance of world data models over US data models

I.2 Recurrent Neural Network Robustness Checks

For our RNN model, we found we could improve forecasting performance by taking the mean prediction of many models estimated by stochastic gradient descent. The ensembling improves performance slightly, but later graphs will show it also improves model stability and variance. Bolded entries indicate outperformance over all Economic models.

TABLE VIII: GRU Mean, Median and Best GDP Forecasts, RMSE

Time (Q's Ahead)	1Q	2Q	3Q	4Q	5Q
GRU with World Data					
Best Model	2.4	2.5	**2.5**	**2.6**	**2.6**
Mean Model	2.3	2.5	**2.5**	**2.6**	**2.6**
Median Model	2.3	2.5	**2.5**	**2.6**	**2.6**

We provide a Monte Carlo simulation (Table IX), estimating our RNN model at each time horizon 100 times. At every horizon, the average root mean squared error of our simulated models indicates competitive, if not outperformance, against baseline models. Interestingly, it seems like

the best-performing model on validation data, when tested on the test data, often performs worse than the average performance over all the models. This is something that should be investigated further, but based on this phenomenon, we recommend that practitioners take a simple mean or median forecast across many different models.

TABLE IX: Baseline GRU Monte Carlo Simulation over Initializations

The mean and standard deviation of the performance of GRUs on the test set 2009-Q1 to 2020-Q1

Time (Q's Ahead)	1Q	2Q	3Q	4Q	5Q
Mean RMSE	2.4	2.6	2.5	**2.6**	**2.6***
Std Dev RMSE	0.06	0.06	0.05	0.06	0.06

A common criticism of deep learning attempts at forecasting is that the models are unreliable, but due to high variance, one can p-hack a model that performs well. The Monte Carlo simulation in Table IX, anticipates this critique. The standard deviation of our RMSE is 0.06 which suggests that all our models have a similar performance on the test set when optimization is complete. This cannot resolve all issues, as the Monte Carlo result only deals with numerical instability. The model could still fit this particular data window or architecture choice, due to chance. In order to respond to those critiques, we also provide robustness checks across different architectures and data periods.

One test we performed was to replace the GRU with a long short-term memory (LSTM) layer (Hochreiter and Schmidhuber, 1997*b*) – another type of RNN. We use the same test data as the main result (USA 2009-Q1 to 2020-Q1) as well as the same data as inputs. The LSTM in Table X are analogous to the gated recurrent unit neural networks models in the table in Section VI.C. in the main text. Mean RMSE and standard deviation RMSE correspond to the entries in the table below. The baseline performances are still the same as the test set has not changed.

The LSTM networks outperform the baseline models along essentially the same time horizons. Performance is also competitive, but consistently a little worse than the gated recurrent unit over all time frames. The LSTM has a similar standard deviation of root mean squared error, suggesting that the two models consistently find a similar optimum when it comes to forecasting. Again, taking a model average through the mean or median forecast results in small but consistent root mean

TABLE X: Baseline LSTM Monte Carlo Simulations

The performance of the best, mean, and median forecasts as well as the mean and standard deviation of the long short-term memory networks on the test set 2009-Q1 to 2020-Q1 (lower is better)

Time (Q's Ahead)	1Q	2Q	3Q	4Q	5Q
Best RMSE	2.4	2.6	2.6	**2.6**	**2.6***
RMSE of Mean	2.4	2.6	2.5	**2.6**	**2.6***
RMSE of Median	2.4	2.5	2.5	**2.6**	**2.6***
Mean RMSE	2.4	2.6	2.5	**2.6**	**2.6***
Std Dev RMSE	0.05	0.07	0.05	0.06	0.06

squared error performance improvements.

Additionally, we re-estimate the model with the slightly different test set from 2009-Q4 to 2019-Q4 as opposed to 2008-Q4 to 2020-Q1, comparing the benchmark economic models to our original GRU (Table XI). The reason we use this alternative training set is that it contains no recessions. Since the highly flexible neural network will have an advantage in forecasting periods with a significant departure from a more linear-friendly period of expansion. Removing the recessions would hamstring our model compared to the more linear model baselines.

Our gated recurrent units were completely re-estimated as we additionally included 2009-Q1 to 2009-Q3 in the validation set. Performance would improve if we left those (recession) timesteps out of the validation set as the test set contains no recessions. However, this decision cannot be rationalized from the point-of-view of an out-of-sample forecaster. Although this version of our model did not outperform the best baseline models along any horizon, considering performance over all horizons, we think our median and mean models are better than the US AR(2), VAR(1), and the Factor Model on this test set, while performing slightly worse than the DSGE model and the world AR(2). This supports our hypothesis that the main outperformance of our model was in highly nonlinear domains like recessions and other regime changes although using the cross-sectional data reduced the tendency for the models to be biased upwards and was a contributor to the RNN's outperformance over models trained only on US data.

This provides supplementary evidence that the outperformance of our neural network is not due to either over-fitting the test set or over-fitting the architecture choice. Additionally, we ran

Monte Carlo simulations (Table XII) which show that given one hundred random initialization and optimization routines over all five horizons, the model still consistently achieves low root mean squared error and has a low standard deviation – demonstrating stability and reproducibility.

TABLE XI: Expansion Root Mean Squared Error

The RMSE Performance of Each Model Only on US Expansion Data

Time (Q's Ahead)	1Q	2Q	3Q	4Q	5Q
VAR(1)					
US data	2.3	2.6	2.9	3.0	3.0
World data	2.1	2.2	2.2	2.2	2.2
AR(2)					
US data	1.7	1.7	1.8	1.9	1.9
World data	**1.6**	1.6	1.6	**1.5**	**1.5***
Smets Wouters DSGE					
US data	1.8	1.8	1.7	1.6	1.5*
Factor					
US data	1.6	**1.6**	**1.6**	1.9	2.1
GRU*					
Best	1.8	2.3	2.0	2.0	1.9
Mean Forecast	1.7	1.7	1.7	1.7	1.7
Median Forecast	1.7	1.7	1.7	1.7	1.7
SPF Median	1.4	1.5	1.5	1.5	1.5

*All RNN models use entire world data cross-section

Table XI, discusses forecast prediction only in expansions. (Lyu, Nie and Yang, 2021) suggest pooling improves forecasting mainly in downturns. However, this chart shows other models, especially the machine learning ones, can also improve in expansions.

TABLE XII: Expansion RNN Monte Carlo Simulations

The mean and standard deviation of the performance of gated recurrent units on the test set 2009-Q4 to 2019-Q4

Time (Q's Ahead)	1Q	2Q	3Q	4Q	5Q
Mean RMSE	1.8	1.8	1.8	1.8	1.7
Std Dev RMSE	0.18	0.18	0.21	0.11	0.08

Department of Economics, University of Michigan, Ann Arbor