Estimating Nonlinear Heterogeneous Agent Models with Neural Networks *

Hanno Kase
European Central Bank

Leonardo Melosi
University of Warwick,
FRB Chicago, DNB, & CEPR

Matthias Rottner
Deutsche Bundesbank

June 27, 2024

Abstract

We leverage recent advancements in machine learning to develop an integrated method to solve globally and estimate models featuring agent heterogeneity, nonlinear constraints, and aggregate uncertainty. Using simulated data, we show that the proposed method accurately estimates the parameters of a nonlinear Heterogeneous Agent New Keynesian (HANK) model with a zero lower bound (ZLB) constraint. We further apply our method to estimate this HANK model using U.S. data. In the estimated model, the interaction between the ZLB constraint and idiosyncratic income risks emerges as a key source of aggregate output volatility.

Keywords: Neural networks, likelihood, global solution, heterogeneous agents, nonlinearity, aggregate uncertainty, HANK, zero lower bound.

JEL classification: C11, C45, D31, E32, E52.

*We thank Borağan Aruoba, Jesús Fernández-Villaverde, Ed Herbst, David Levine, Christian Matthes, Galo Nuño, Giorgio Primiceri, and Frank Schorfheide for their very helpful suggestions. We also thank seminar participants at the NBER Summer Institute 2023, Banque de France, Deutsche Bundesbank, Bank of Finland, University of Copenhagen, Schumpeter-BSE-Seminar in Berlin, University of Liverpool, CEBRA 2022 Annual Meeting, Conference on advanced analytics by the ECB, BoE and KCL, EUI, Conference on Non-traditional Data, Machine Learning and NLP in Macroeconomics at Sveriges Riksbank, Ghent University Workshop on Empirical Macroeconomics, 29th Symposium of the SNDE, SEM Conference 2022, EEA Conference 2022, Midwest Macro 2022, T2M 2023, CEF 2023, and Lancaster-ETM 2024 Workshop on Empirical and Theoretical Macroeconomics. We thank Jesús Laso Pazos for his excellent work as research assistant. Hanno Kase gratefully acknowledges the financial support of the US Army under award number W911NF2010242. The views in this paper are those of the authors and should not be interpreted as reflecting the views of the De Nederlandsche Bank, the Deutsche Bundesbank, the European Central Bank, the Federal Reserve Bank of Chicago, or any other person associated with the Eurosystem and the Federal Reserve System.

1 Introduction

Over the past three decades, a significant advancement in macroeconomic research has been the integration of heterogeneity into dynamic general equilibrium models. These models enable the study of distributional issues, economic fluctuations, and stabilization policies within a single micro-founded framework. The complexity of these models often necessitates introducing tractability assumptions that largely preclude scholars from considering nonlinear aggregate dynamics and their interactions with heterogeneity. Yet, accounting for these nonlinear dynamics is essential to understand recent macroeconomic events, including recurring and prolonged periods at the zero lower bound (ZLB), deep recessions, and the recent rise in inflation. In this paper, we leverage recent advances in machine learning to develop a neural network (NN)-based method for solving and estimating models with heterogeneous agents, fully incorporating nonlinear aggregate dynamics. We apply our method to perform likelihood estimation of a nonlinear Heterogeneous Agent New Keynesian (HANK) model using US data.

Estimating structural models entails searching for parameter values that globally maximize an objective function, which could be a likelihood function, a posterior distribution, or the distance between a set of empirical moments or impulse response functions and their model counterparts. This search involves evaluating the objective function at potentially numerous parameter values. For each parameter value, two steps are typically taken. First, the model's policy functions are characterized (*solution step*). Second, once the policy functions are properly approximated for that parameter value, the objective function is evaluated (*evaluation step*).

When estimating the parameters of an economic model, these two steps are repeated multiple times to find the global maximum of the objective function. This repetition is not problematic if the model can be solved within a fraction of a second, using a routine that can be easily automated. For instance, this is the case of linearized representative-agent models. However, repeatedly solving a highly complex

nonlinear model, such as nonlinear HANK models with aggregate uncertainty, is unmanageable. Additionally, in the context of HANK models, the solution step may involve computing the steady-state equilibrium afresh, which is a quite computationally intense task to perform.

To overcome these computational hurdles, we approximate the model's policy functions without conditioning on specific parameter values. By treating the parameters as inputs, or pseudo-state variables, of the policy functions, we only need to solve the model once, and only before launching the estimation. While this approach expands the dimensionality of the model's policy functions to approximate, NNs are highly effective in managing high-dimensional problems (*scalability*), such that the increase in computational burden remains manageable.[1] Once the model's policy functions are approximated over the parameter space, the model's solution can be retrieved in a fraction of a second for any combination of parameters.

An additional computational challenge arises when performing likelihood estimation of nonlinear models. Evaluating the likelihood of these models requires the use of filters based on Monte Carlo (MC) methods, such as the particle filter. These filters are computationally more expensive and less accurate compared to those used for evaluating the likelihood of linearized Dynamic Stochastic General Equilibrium (DSGE) models. Therefore, minimizing the frequency of likelihood evaluations and improving the accuracy of filter-based assessments are critical objectives.

To achieve these objectives, we train another NN –*the neural network particle filter* – to obtain an approximation of the likelihood prior to estimation. This additional NN is trained through three steps. First, we obtain several thousand quasi-random parameter draws (training sample). Second, we evaluate the likelihood at each of these draws using a standard particle filter. To perform this step we do not need to solve the model multiple times as we can utilize the NNs trained to provide a fast

[1]NNs are the fundamental building blocks of deep learning, which is a subset of machine learning methods. They provide a flexible functional form that can efficiently approximate arbitrarily complex, nonlinear functions, potentially with a large set of inputs.

and reliable approximation of the model solution at each draw of the training sample. Third, the NN is trained to predict the likelihood evaluated at these draws.[2] The NN particle filter delivers an accurate and smooth approximation of the likelihood function by effectively ironing out the inaccuracies of the particle filter. With the approximated likelihood function at hand, there is no need to repeatedly run a time-consuming MC filter to maximize this function. Instead, the function can be evaluated almost instantaneously at any parameter value using the NN approximation.

Importantly, our methodology facilitates the estimation of parameters influencing the model's steady-state equilibrium. By treating model parameters as pseudo-state variables, the steady-state equilibrium needs to be solved only once with a specifically trained NN. This feature alleviates a significant bottleneck, as solving the steady state of these models at each solution step is computationally intensive. Furthermore, this property of our method presents a crucial advantage, as it enables the estimation of a greater number of parameters, expanding the scope of the estimation analysis.

We provide three proofs of concept to validate our methodology. First, we show that our approach replicates the policy functions of a small-scale, linearized Representative Agent New Keynesian (RANK) model, which can be solved analytically. Second, we compare our method for likelihood estimation to an alternative state-of-the-art method for nonlinear models. Since this alternative method cannot treat parameters as pseudo state variables, its scope of application is considerably limited. In light of these limitations, we are compelled to consider a stylized representative-agent model featuring only one source of aggregate nonlinearity – the ZLB constraint. We show that the two methods deliver remarkably similar estimation outputs, suggesting that our method can perform equally well as a state-of-the-art global method, but with the advantage of being applicable to a much broader set of models. Third, we simulate a nonlinear HANK model and use the simulated data to estimate its parameters, including those affecting the steady-state equilibrium outcomes. We show

[2]We minimize the mean squared errors between the likelihood vlaues approximated by the NN and the likelihood values computed via the standard particle filter in the second step.

that our method successfully recovers the true parameter values of the model.

We then estimate a HANK model using US data. This model integrates both idiosyncratic and aggregate shocks, along with nonlinearities such as individual borrowing limits and a ZLB constraint. Despite its stylized nature, the estimated model matches a set of key moments of the data fairly closely. Idiosyncratic income risk is found to explain a large share of volatility in the macro data. This finding is primarily due to the presence of the ZLB constraint. As idiosyncratic risk rises, agents save more and the interest rate falls, increasing the risk of encountering the ZLB. The heightened ZLB risk leads to a higher volatility in inflation and output, as the central bank's capacity to stabilize the economy is diminished by the binding constraint.

Literature. Since Krusell and Smith (1998) wrote their seminal paper, the literature has introduced a variety of methods to solve models with heterogeneous agents (e.g. Algan et al., 2008; Reiter, 2009; Den Haan et al., 2010 Ahn et al., 2018; Boppart et al., 2018; Bayer et al., 2019; Auclert et al., 2021; and Winberry, 2021). These methods rely on linear approximations of the aggregate dynamics of the model and are therefore not suitable for studying the interactions between nonlinearities and heterogeneity in general equilibrium. Recently, the literature has applied these linearized techniques to estimate HANK models using likelihood methods (e.g. Challe et al., 2017; Bayer et al., 2024a; Auclert et al., 2020; Lee, 2020; Auclert et al., 2021; Bilbiie et al., 2023; and Acharya et al., 2023). Unlike in our paper, estimation is very often confined to parameters that do not affect the model's steady state and aggregate nonlinearities (e.g. the ZLB constraint) are not taken into account. Bhandari et al. (2023) and Bayer et al. (2024b) pioneered methods to solve heterogeneous agent models by approximating their aggregate dynamics to higher orders. In this paper, we introduce NN-based methods to solve these models using global methods. Our method also allows structural estimation of models with heterogeneous agents.

Our paper is connected to the literature that develops NNs to solve complex

dynamic macroeconomic models globally such as Fernández-Villaverde et al. (2020), Ebrahimi Kahou et al. (2021), Maliar et al. (2021), Azinovic et al. (2022), Duarte et al. (2024), and Valaitis and Villa (2024). Our contribution to this literature is to show how to use NNs to not only solve but also to estimate nonlinear heterogeneous agent models. Estimation – including the simulated method of moments, impulse-response-functions matching, maximum likelihood, and Bayesian estimation – adds an additional important layer of complication in that it requires solving the model at hundreds of thousands of different points in the parameter space.

Fernández-Villaverde et al. (2023a) estimate the critical parameter of a model with a financial sector and heterogeneous households with NNs using the time-series of GDP growth. We can expand the scale of the estimation exercise and estimate several parameters using multiple observable variables because (i) we combine the scalability of NNs with the intuition of treating the model parameters as pseudo-state variables when solving the model and (ii) we use NNs to facilitate the evaluation of the likelihood via the filter - which we call *NN particle filter*. Furthermore, differently from Fernández-Villaverde et al. (2023a), we estimate a HANK model. To our knowledge, we are the first to estimate a HANK model in its nonlinear specification.

HANK models are attracting growing attention of scholars (e.g. Oh and Reis, 2012; McKay and Reis, 2016; McKay et al., 2016; Den Haan et al., 2017; Guerrieri and Lorenzoni, 2017; Ravn and Sterk, 2017; Kaplan and Violante, 2018; Kaplan et al., 2018; Auclert, 2019; Bilbiie, 2020 and Ravn and Sterk, 2020; Gornemann et al., 2021; Luetticke, 2021; and Auclert et al., 2023).[3] Methods to solve these models globally by relying on NNs have been recently developed (e.g. Maliar and Maliar, 2020; Gorodnichenko et al., 2021; Fernández-Villaverde et al., 2023b; and Han et al., 2021) or by introducing alternative approaches unrelated to machine learning (e.g. Schaab, 2020; and Lin and Peruffo, 2024). Compared to this strand of literature, we show how to leverage the scalability of NNs to enable estimation of nonlinear HANK models.

[3]Scholars have also explored general equilibrium models where heterogeneity characterizes the firm side – e.g. Ottonello and Winberry (2020).

The paper is organized as follows. In Section 2, we explain the architecture of our NN-based methodology to solve and estimate nonlinear heterogeneous agent models. In Section 3, we provide three proofs of concept of our method. We solve and estimate a nonlinear HANK model using US data in Section 4. In Section 5, we conclude.

2 A neural-network framework for estimation

In this section, we describe how to leverage the properties of NNs to enable the estimation of nonlinear models with heterogeneous agents. We first provide a primer on NNs. Second, we explain how NNs can be used to efficiently characterize the policy functions of a model treating the parameters as pseudo-state variables. Third, we show how another NN can be trained to characterize the deterministic steady-state equilibrium of the model, when this equilibrium influences its policy functions. Fourth, we explain how to train the NN particle filter.

2.1 Neural networks

NNs are function approximators well-suited for learning highly dimensional nonlinear functions.[4] Consider a function $Y = \psi(X)$, mapping a vector of inputs X into a vector of outputs Y. The function can be approximated by a NN denoted by:

$$Y = \psi_{NN}(X|W), \tag{1}$$

where W are the weights defining the NN. As illustrated in Figure 1, the NN consists of several layers, which can be divided into the input layer, hidden layer(s) in between, and the output layer. Each hidden layer consists of several neurons, which are represented as circles in the figure.

[4]NNs are the fundamental building blocks of deep learning – a class of machine learning techniques. The universal approximation theorem states that a feedforward network with at least one hidden layer can approximate any continuous function in a finite-dimensional space with any desired non-zero error given a sufficient width (Hornik et al., 1989.).

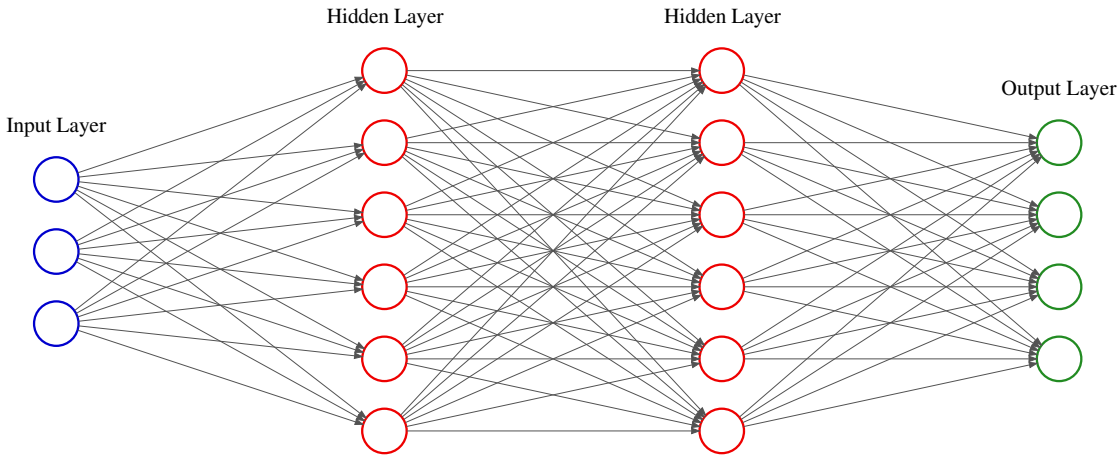

Figure 1: Illustration of a neural network. An illustrative example of a NN with three inputs (blue circles), two hidden layers (the two rows of red circles) with six neurons (the red circles) each, and four outputs (the green circles).

A single neuron is a composition of an affine function with inputs $x_1, x_2, \ldots x_M$, weights $w_1, w_2, \ldots, w_M$, and bias w_0 and a nonlinear activation function $g(\cdot)$ that returns a single output $\tilde{y}$:[5]

$$\tilde{y} = g(w_0 + \sum_{i=1}^{M} w_i x_i). \tag{2}$$

The structure and weights characterize a NN. For example, the NN illustrated in Figure 1 has four layers containing in total 94 weights (including biases).[6] In matrix notation, we can write a single layer h of the NN as:

$$y = g\left(\mathcal{A}^h(x)\right), \text{ where } \mathcal{A}^h(x) \equiv W^h x, \tag{3}$$

where the inputs, outputs, and weights are now written as vectors.[7] As implied by equation (3) and Figure 1, NNs are natural to vectorize, making them efficient

[5]The nonlinear activation function $g(\cdot)$ makes it possible for the NN to approximate nonlinear functions. Popular examples of activation functions are ReLU (rectified linear unit), CELU (continuously differentiable exponential linear units), or Tanh (hyperbolic tangent).

[6]The first hidden layer has $(3+1) \times 6 = 24$ weights (including biases); the second hidden layer has $(6+1) \times 6 = 42$; the output layer has $(6+1) \times 4 = 28$. The first term in the brackets is the weights plus the bias at each neuron, and the second term is the number of neurons in the layer.

[7]The entire NN with H hidden layers can be written as a composition of functions $\psi_{NN}(X|W) = \left(\mathcal{A}^{H+1} \circ g \circ \mathcal{A}^H \circ g \circ \mathcal{A}^{H-1} \circ \cdots \circ g \circ \mathcal{A}^1\right)(X)$ where $\circ$ denotes a function composition.

to compute, especially on graphics processing units (GPUs) that excel at matrix multiplications and element-wise operations.[8]

The weights W are optimized (trained) to minimize a loss function; e.g. the mean squared error between observed and predicted values of the function to approximate. The optimization of the weights of the NNs relies on the back-propagation algorithm, which allows computing all partial derivatives of the loss with respect to the weights of the NN in approximately the same number of operations, and similar computational cost, as computing the loss itself. The partial derivatives inform the direction in which an optimization algorithm should adjust the weights of the NN to minimize the loss. In particular, we use a stochastic gradient descent based optimization algorithm called AdamW, which is currently one of the most widely used optimizers for training NNs. The accuracy of the NN approximation and the speed of the training process can be improved by adjusting the learning rate during the training, which is done using a learning rate scheduler. A common choice is to gradually decrease the learning rate as the training progresses, following, for example, a linear or exponential schedule.

2.2 Neural-network solution method

We consider a large class of dynamic general equilibrium models, whose dynamics are governed by the following transition equations or laws of motion:

$$\mathbb{S}_t = f\left(\mathbb{S}_{t-1}, \nu_t | \, \Theta\right), \tag{4}$$

where $\mathbb{S}_t$ is a finite vector of S state variables of the model. The economy is subject to U exogenous shocks ν_t that affect the response of the state variables. The model's structural parameters are denoted by Θ. The function $f(\cdot)$ is possibly nonlinear and typically does not have a closed-form analytical characterization.

Solving the model amounts to finding a set of O policy functions that map the state variables, $\mathbb{S}_t$ to a set of control variables, ψ_t for given values of the model

[8] To further improve efficiency, the calculations are typically done for batches of inputs.

parameters, Θ: $\psi_t = \psi(\mathbb{S}_t|\Theta)$. These policy functions satisfy a set of K equilibrium conditions derived from the model equations:[9]

$$F(\psi(\mathbb{S}_t|\Theta)) = 0. \tag{5}$$

With the policy functions $\psi(\cdot)$ at hand, the transition equations (4) can be obtained.

Extended policy functions. A critical bottleneck for estimation is the need to solve policy functions for a potentially very large and a-priori unknown number of times at different parameter values. In the context of the models studied in this paper, this task is often unmanageable. To address this issue, we treat the parameters to estimate as pseudo-state variables when approximating the policy functions.[10]

We define the *extended policy functions* as

$$\psi_t = \psi\left(\mathbb{S}_t, \tilde{\Theta}|\bar{\Theta}\right), \tag{6}$$

where the vector of parameters Θ is split into $\tilde{\Theta}$ – the set of parameters to estimate – and $\bar{\Theta}$ – the set of parameters that are fixed in estimation. The characterization of the extended policy functions leads to the extended transition equations:

$$\mathbb{S}_t = f\left(\mathbb{S}_{t-1}, \nu_t, \tilde{\Theta}|\bar{\Theta}\right). \tag{7}$$

Agents' heterogeneity. In the literature studying heterogeneous agent models, it is commonly assumed that there exists a continuum of agents. However, this assumption makes the state vector infinite dimensional, which of course presents insurmountable computational challenges. Therefore, when approximating the policy functions of these models, we discretize the model population by keeping track of a finite number L of agents.[11]

[9] Examples of these conditions may be the Euler equation or the resource constraint.

[10] Examples of computational papers that use pseudo-state variables in combination with machine learning are Norets (2012), Scheidegger and Bilionis (2019), and Duarte and Fonseca (2024).

[11] Our methods can be applied to solve models featuring a finite number of industries or countries.

The transition equations of heterogeneous agent models can be recast in exactly the same form as in equation (4), where the state vector $\mathbb{S}_t$ comprises both the set of aggregate state variables, $\mathbb{S}_t^A$, and the set of individual state variables for each agent, $\{\mathbb{S}_t^i\}_{i=1}^L$, and the vector of shocks, ν_t, includes both aggregate shocks, ν_t^A, and idiosyncratic shock for each agent, $\{\nu_t^i\}_{i=1}^L$.

In contrast to the previous case with no heterogeneity, we now have two sets of policy functions to approximate: the aggregate policy functions, denoted as $\psi^A(\cdot)$, and the individual policy function, represented by $\psi^I(\cdot)$.[12] Following the approach introduced earlier, these functions are expressed in their extended form by treating the parameters to estimate, $\tilde{\Theta}$, as pseudo-state variables.

$$\psi_t^A = \psi^A(\mathbb{S}_t, \tilde{\Theta}|\bar{\Theta}) \quad \text{and} \quad \psi_t^i = \psi^I(\mathbb{S}_t^i, \mathbb{S}_t, \tilde{\Theta}, |\bar{\Theta}), \tag{8}$$

where $\mathbb{S}_t^i$ denotes the individual state variables. The number of individual and aggregate state variables are S^i and S_t^A, respectively, so the total number of state variables is $S = S^i \times L + S^A$. The number of exogenous shocks and policy functions is similarly defined as $U = U^i \times L + U^A$ and $O = O^i \times L + O^A$, respectively.[13]

Neural networks and training. We set up two NNs, $\psi_{NN}^I(\cdot)$ and $\psi_{NN}^A(\cdot)$, to approximate the individual and aggregate extended policy functions, $\psi^I(\cdot)$ and $\psi^A(\cdot)$, defined in equation (8). We train these two NNs by choosing their weights, W, so as to minimize the weighted squared residual error for the equilibrium conditions:

$$\Phi^L = \sum_{k=1}^{K} \alpha_k [F_k(\psi_{NN}(\mathbb{S}_t, \tilde{\Theta}|\bar{\Theta}))]^2, \tag{9}$$

[12]Note that the individual function is the same across agents, indicating that agents with the same individual state variables $\mathbb{S}_t^i$ make the exactly same decisions, ψ_t^i.

[13]In this paper, we track all the states of the agents as in Maliar et al. (2021) for instance. An alternative approach is to reduce the dimension of the individual states to track by exploiting the symmetry of the agents following Ebrahimi Kahou et al. (2021). Our method can be combined with either approach.

where $F_k(\cdot)$ denotes the residual error associated with the k-th equilibrium condition and α_k is the weight of this condition.[14] We include at least as many optimality conditions as the number of policy functions, so that $K \geq O^i \times L + O^A$.

To leverage the potential for parallelization provided by GPUs, the two NNs are trained on a batch with size B. This step amounts to having B pseudo-randomly drawn state vectors, $\mathbb{S}_{t,b}$ and parameters to estimate, $\tilde{\Theta}_b$. In each optimization step, we minimize the loss functions Φ^L averaged across batch size B:

$$\bar{\Phi}^L = \frac{1}{B} \sum_{b=1}^{B} \sum_{k=1}^{K} \alpha_k \left[F_k(\psi_{NN}(\mathbb{S}_{t,b}, \tilde{\Theta}_b | \bar{\Theta})) \right]^2. \tag{10}$$

This loss function is minimized by using a stochastic gradient descent method.

As standard, to evaluate the residual error for some of the equilibrium conditions, $F_k(\psi(\cdot))$, we are required to compute agents' expectations. We do so by using Monte Carlo (MC) integration. Specifically, we randomly draw next period's shocks to approximate expectations.[15] To increase precision, one can conduct several optimization rounds with new sets of drawn shocks to evaluate expectations.

The training of the NNs to approximate the aggregate and idiosyncratic policy functions concludes the solution of the model. Since the parameters to be estimated are treated as pseudo-state variables of the approximated policy functions, the model can be simulated and its transition equations obtained in just a fraction of a second, as these tasks only require evaluating the previously trained NNs. This approach eliminates the need to repeatedly solve the model, which is a critical bottleneck when estimating models with a large set of state variables.

Stochastic solution domain and truncated parameter space. The grid points for the state variables and parameter values at the beginning of each optimization step are drawn randomly. We aim to train the algorithm only on the relevant domain of the

[14] The weights can vary so as to take into account that losses across different equations possibly have different magnitudes.

[15] To reduce the variance of MC integration, we use the antithetic variate method in this paper.

state space. Following the approach of Maliar et al. (2021), we directly sample from the model to approximate the ergodic distribution of the state variables, also known as the stochastic solution domain. Specifically, we simulate each economy (batch) forward using the NNs trained at the current optimization step. The endpoint of the simulation provides an approximation of the ergodic distribution of the state variables, which we use for the next optimization step.

We truncate the space from which we draw the model parameters to estimate (the pseudo-state variables). By limiting the admissible set of parameter values, we achieve a more accurate evaluation of the ergodic distribution of the state at each optimization step. In the case of Bayesian estimation, truncating the parameter space when approximating the model's solution can be made consistent through the use of truncated prior distributions.

Parameters affecting the deterministic steady state (DSS). Often, solving a heterogeneous agent model requires knowing its DSS.[16] Finding the DSS necessitates solving a Bewley-Huggett-Aiyagari type economy over the parameter space, which is a computationally intensive step. To address this, we train an auxiliary neural network (NN) to approximate the DSS. This auxiliary NN is then used when training the other two NNs (ψ^I_{NN} and ψ^A_{NN}) that approximate the aggregate and idiosyncratic policy functions defined in equation (8).

2.3 The neural network particle filter

The likelihood function of DSGE models is typically evaluated in two steps. In the *solution step*, the model's policy functions must be characterized. The neural networks (NNs), trained as described in the previous section, can be utilized to perform this task quickly and at several points in the model's parameter space. With the policy

[16]The DSS is typically defined as the model economy in the absence of aggregate shocks (e.g., Fernández-Villaverde et al., 2023b). For example, in the context of New Keynesian models, the monetary policy rule is typically specified as a function of the model's DSS interest rate and output.

functions at hand, we can construct the transition equations (7) which, combined with the measurement equations that map the observable variables to the appropriate model concepts, form the state-space representation of the model. Since the likelihood cannot be analytically characterized, it is typically evaluated by applying a filter to the state-space representation in the *evaluation step*. Filters return the model's one-step-ahead predictive densities over the available sample period, $p_{\bar{\Theta}}\left(y_t|y_{t-1},...,y_1,\tilde{\Theta}\right)$, where y_t denotes the vector of observable variables in period t. The product of the model's predictive densities in every period $t \in 1,...,T$ is the likelihood function:

$$\mathcal{L}_{\bar{\Theta}}\left(\mathbb{Y}_{1:T}|\tilde{\Theta}\right) = \prod_{t=1}^{T} p_{\bar{\Theta}}\left(y_t|y_{t-1},...,y_1,\tilde{\Theta}\right). \tag{11}$$

Thus, for a given data set of T periods, $\mathbb{Y}_{1:T}$, a filter can be thought as a mapping from the set of model parameters, $\tilde{\Theta}$, to the value of the likelihood function.[17] We train a new NN – the *NN particle filter* – to accurately approximate this mapping.

The likelihood function of nonlinear models requires MC filters for evaluation.[18] Compared to filters used with linear models, MC filters are both more time-consuming and less accurate. These challenges are particularly pronounced when estimating models with a highly dimensional state space such as nonlinear heterogeneous agent models. As we will show, our NN particle filter significantly mitigates these shortcomings of MC filters.

The NN particle filter is trained through the following three steps. First, we obtain several thousands of quasi-random draws from the parameter space. Second, we execute the (standard) particle filter to derive the likelihood value at each of these draws. It is noteworthy that for this step, we employ the trained NNs to approximate

[17] The notation makes it clear that while the likelihood function depends on all the parameters $\Theta \in \left\{\tilde{\Theta},\bar{\Theta}\right\}$, we maximize this function only with respect to the subset $\tilde{\Theta}$.

[18] An example of MC filter is the particle filter (Fernández-Villaverde and Rubio-Ramírez, 2007). This filter is often used by the economic literature to estimate the distribution of the state variables or to evaluate the likelihood of nonlinear models; e.g. models with occasionally binding constraints (Gust et al., 2017; Atkinson et al., 2020; Aruoba et al., 2021) or with (endogenous) multiple equilibria (Aruoba et al., 2018; Rottner, 2023).

the policy functions and the DSS, which were described in Section 2.2. Consequently, this step is not particularly time-consuming. Third, the NN particle filter is trained to minimize the mean squared error between its predicted values and the likelihood values evaluated in the second step.[19]

The NN particle filter undergoes training over thousands or even millions of optimization steps. Specifically, the training sample is divided into batches of size B. Each batch is fed into the NN particle filter, which generates a vector of approximated likelihoods. The loss is then calculated as the mean squared deviation with respect to the likelihood values evaluated by the standard particle filter across batches. A gradient-based optimizer adjusts the weights of the NN particle filter to minimize this loss. This process continues until all points in the training sample have been used, completing the first epoch. This procedure repeats for thousands of epochs. Over epochs, the learning rate of the gradient descent method is progressively lowered, allowing for broader exploration in earlier epochs and finer search in later epochs.

Dealing with overfitting. It is critical to prevent the NN particle filter from overfitting the training sample of drawn parameters. Overfitting occurs when NNs learn too many details from the training sample, including noise generated by computational inaccuracies in the likelihood values approximated by an MC filter.

To mitigate the problem of overfitting, after training the NN particle filter, we use a validation sample of quasi-randomly drawn parameter values. Importantly, we do not optimize the weights of the NN when it is tested over the validation sample. Therefore, the loss in the validation sample provides a helpful metric to determine when to stop minimizing the loss criteria before the NN starts overfitting the training sample. In Section 3.1, we show how to assess potential issues with overfitting of the NN particle filter.

[19]It is often convenient to work directly with the log-likelihood. This approach entails summing the log of the one-step-ahead predictive densities across all the periods in the sample, T. It should be noted, however, that using the log-likelihood introduces a slight bias due to Jensen's inequality (specified for the logarithm as $E(\ln(X)) \leq \ln E(X)$) because our approach relies on averaging.

Bayesian estimation and forecasting. The NN approximation for the likelihood explained above can be used to approximate the moments of the posterior distribution for the parameters quite straightforwardly. Details on how to combine the likelihood approximated with NNs and standard simulators of posterior draws, such as the Random-Walk Metropolis-Hastings (RWMH) algorithm, are provided in Appendix A. Our accompanying code relies on *PyTorch* as the machine learning framework.

"Real-time estimation and forecasting often entail the iterative estimation of model parameters as new data becomes available. One way to do that in the context of our methodology is to store the filtered states in the last period of the sample. As the new data become available, we run the filter only over the new periods and recompute the values of the likelihood to retrain the NN particle filter.

Another approach involves calculating the likelihood values using future data as pseudo-state variables. We can train the NN that approximates the likelihood function by randomly drawing future data that have not yet been observed. As the data are released, we can obtain the value of the likelihood in a fraction of a second from the NN particle filter trained with simulated future data. The second approach could be adjusted to accommodate data revisions for a few periods back.

2.4 Further discussion

The accuracy of estimation can be enhanced by adopting a sequential approach to the proposed NN-based estimation method. First, parameter values can be drawn from an uninformative unit hypercube to train the relevant NNs, which then provide an approximation of the likelihood. Second, we use this approximated likelihood to refine the bounds and the proposal distribution of the parameters in order to retrain the NNs approximating the policy functions as well as the NN particle filter. This iterative cycle could be repeated several times to improve accuracy.

Non-likelihood methods, such as the method of moments or impulse-response matching, can be embedded in this framework. Instead of using the NN to approxi-

mate the likelihood, NNs are trained using a loss function defined with respect to the moments to match or impulse response functions.

3 Validation of the method

We provide three proofs of concept to establish the validity of our estimation method. Specifically, we assess our NN-based method i) by comparing the policy functions and the likelihood approximated with NNs to the true solution of a linearized small-scale New Keynesian DSGE model, which admits analytical solution; ii) by comparing the policy functions approximated with NNs to those obtained with a state-of-the-art (non-NN) global method when solving a RANK model with a recurrently binding ZLB constraint; iii) by showing the ability of our methodology to recover the true parameter values of a nonlinear HANK model with an occasionally binding ZLB constraint using simulated data.

3.1 Proof of concept 1: a linearized small-scale DSGE model

We consider an off-the-shelf three-equation New Keynesian model with one shock (TFP shock). The model is log-linearized around its unique steady-state equilibrium and the resulting equations are shown in Appendix B.

3.1.1 The extended policy functions

We use our NN approach to approximate the extended policy functions of the output gap and inflation. For reasons explained earlier, we truncate the parameter space, as shown in Table 1. We use 500,000 iterations to train the NN by minimizing the residual error for the Euler equation and the Phillips curve. These policy functions are conditioned on the state variable and the parameters, leaving us with nine input.[20]

[20]The NN contains five hidden layers with 256 neurons each. The activation function for the hidden layers is continuously differentiable exponential linear units (CELU). The optimizer is AdamW. The learning rate follows a cosine annealing scheduler starting from an initial rate of $1e-4$ and

Parameters		LB	UB	Parameters		LB	UB
β	Discount factor	0.95	0.99	θ_Π	Mon.pol. inflation response	1.25	2.5
σ	Relative risk aversion	1	3	θ_Y	Mon.pol. output response	0.0	0.5
η	Inverse Frisch elasticity	1	4	ρ_A	Persistence TFP shock	0.8	0.95
φ	Price duration	0.5	0.9	σ_A	Std. dev. TFP shock	0.02	0.1

Table 1: Model parameters values – first proof of concept. The table shows the parameters of the three-equation New Keynesian model. The lower bound (LB) and upper bound (UB) show how we truncate the parameter space when training the NN to approximate the model solution.

The batch size is set to $1,000$. We conduct five optimization steps to approximate the policy functions. The convergence of the NN is shown in Figure 2, which underscores a considerable degree of accuracy in approximating the two policy functions, with the mean squared residual error averaged across the two loss components reaching values around 10^{-10}.

In Figure 3, we show the extended policy function of the output gap as the structural parameters of the model vary. The extended policy function is evaluated at a negative one standard deviation shock, and the unvaried parameters are fixed at the middle value of their bounds, shown in Table 1. The comparison with the analytical solution demonstrates that our NN is able to replicate the true solution extremely well as the lines almost perfectly coincide. It is also noteworthy that NNs have the ability to approximate the nonlinearities in the policy function well.Identical conclusions on the accuracy of the method can be drawn from a similar graph showing the policy function of inflation (Appendix B).

3.1.2 Likelihood approximation and estimation

As a next step, we estimate the likelihood of this model with our NN approach. We simulate inflation and the output gap from the linearized RANK model for 100 periods and use these simulated series as data. In the simulation, the parameters are

terminating at a rate of $1e-6$.

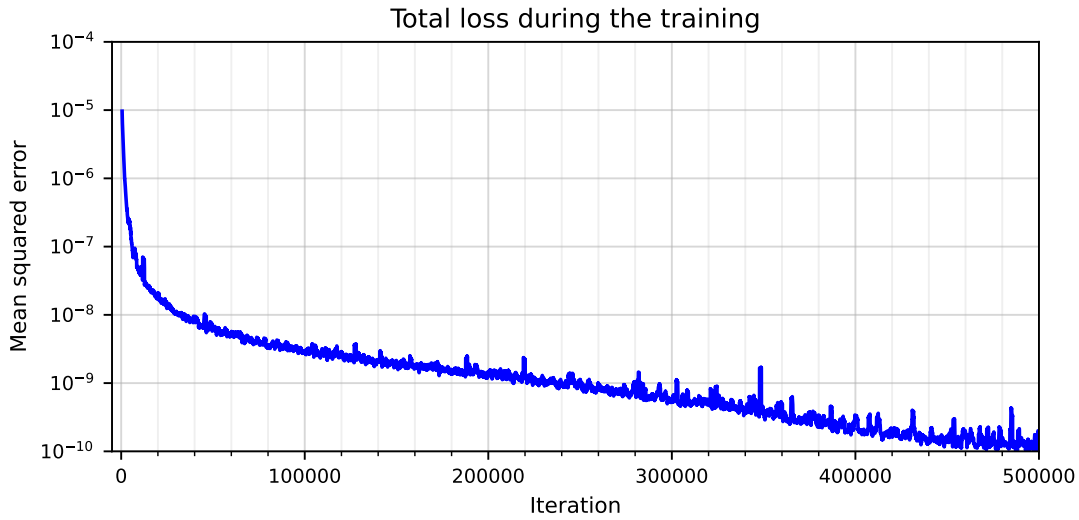

Figure 2: Convergence of the neural network solution. The figure shows the dynamics of the mean squared error. The loss is calculated as a moving average over 1000 iterations. The vertical axis has a logarithmic scale.

set to the middle of their bounds.[21]

We train the NN particle filter using a sample of likelihood values evaluated at 10,000 quasi-random parameter draws.[22] These likelihood approximations are obtained by running the standard particle filter for each draw. When running the particle filter, we choose a low number of particles (100 particles) to deliberately add noise to the likelihood approximations with the objective of later illustrating the problem of overfitting and how one can successfully deal with it. We will show that even though the filter delivers only very noisy estimates of the likelihood, the NN is able to correct this inaccuracy and interpolate the likelihood function by cutting through the noise generated by the filter.

We divide our sample of 10,000 likelihood approximations into a training sample (80%) and a validation sample (20%). The NN is trained using the training dataset across 20,000 epochs, wherein each epoch, the entire training dataset is processed. The training sample is divided into batches of 100. Each batch is passed

[21] We add a small measurement error (5% of the variance of each observable variable) to avoid the well-known problem of degeneracy of the likelihood function when using the particle filter.

[22] This NN contains 4 hidden layers with 128 neurons each. The activation function for the hidden layers are CELU. The optimizer is AdamW. The learning rate follows a cosine annealing scheduler from an initial level of $2e-4$ and terminating at $1e-7$.

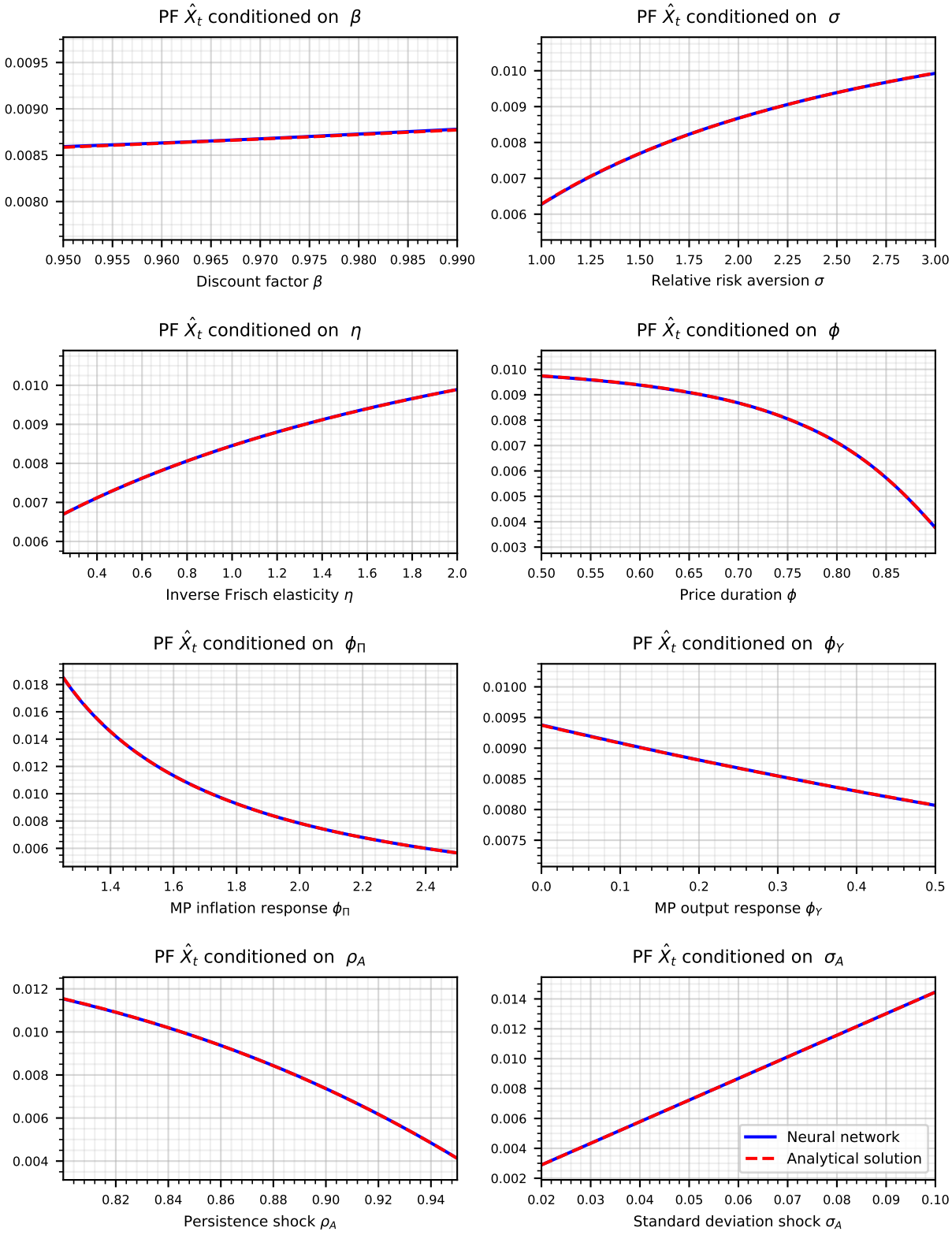

Figure 3: Accuracy of the NN solution method. Comparison between the analytical policy function for the output gap and the approximation based on NNs. The values of the fixed parameters in each graph are set to the middle of their bounds shown in Table 1.

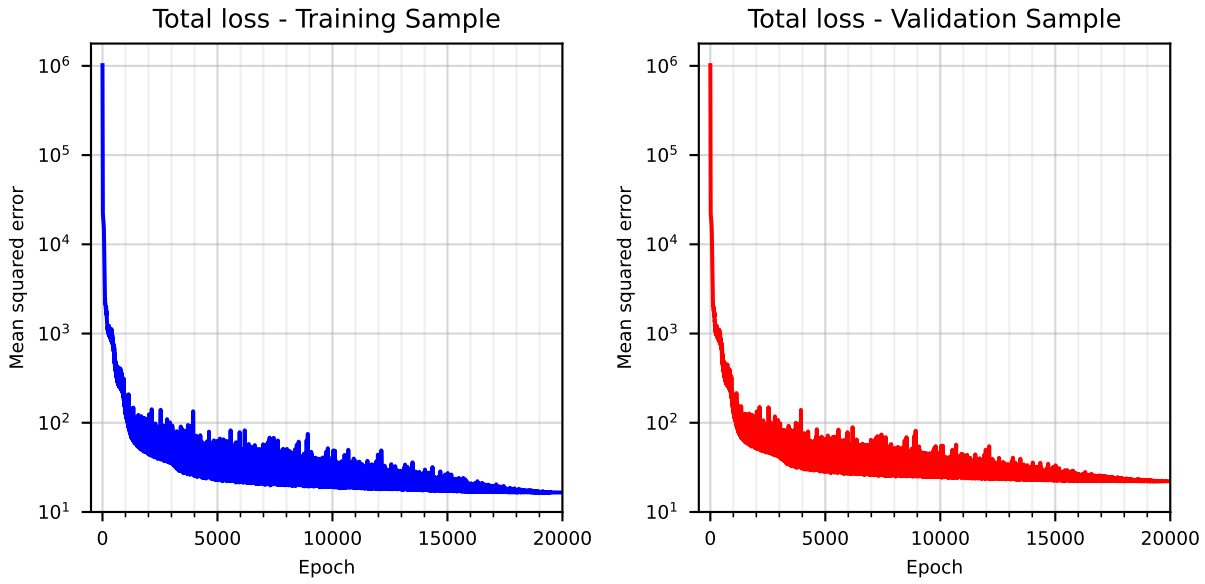

Figure 4: Training and validation convergence. The figure shows the total loss over the training sample (left) and over the validation sample (right). An epoch is completed when all the training or validation sample points are utilized. The vertical axis is expressed in a logarithmic scale.

simultaneously to the optimization routine. An optimization step is completed when the routine converges and returns the optimal NN weights. With 8,000 parameter draws in our training sample, this optimization step is performed 80 times within each epoch. In this application, we consider 20,000 epochs, resulting in a total of 1.6 million optimization steps across all epochs. The dynamics of total loss during the training of the NN are depicted in Figure 4. The total loss converges to a value whose magnitude reflects the accuracy of the particle filter, primarily dependent on the number of particles used (in this case, 100 particles).

To assess potential concerns about overfitting of the NN particle filter, in the right chart of Figure 4, we report the total loss when the likelihood is evaluated at the parameters of the validation sample using the NN weights trained in every epoch (x-axis). The proximity of errors across epochs in both samples, depicted on the left and right charts, along with the consistent decrease in error over the validation sample, suggests that the NN is not overfitting.

To more directly assess potential issues of overfitting, we compare likelihood approximations using different methods. In Figure 5, we plot how the likelihood –

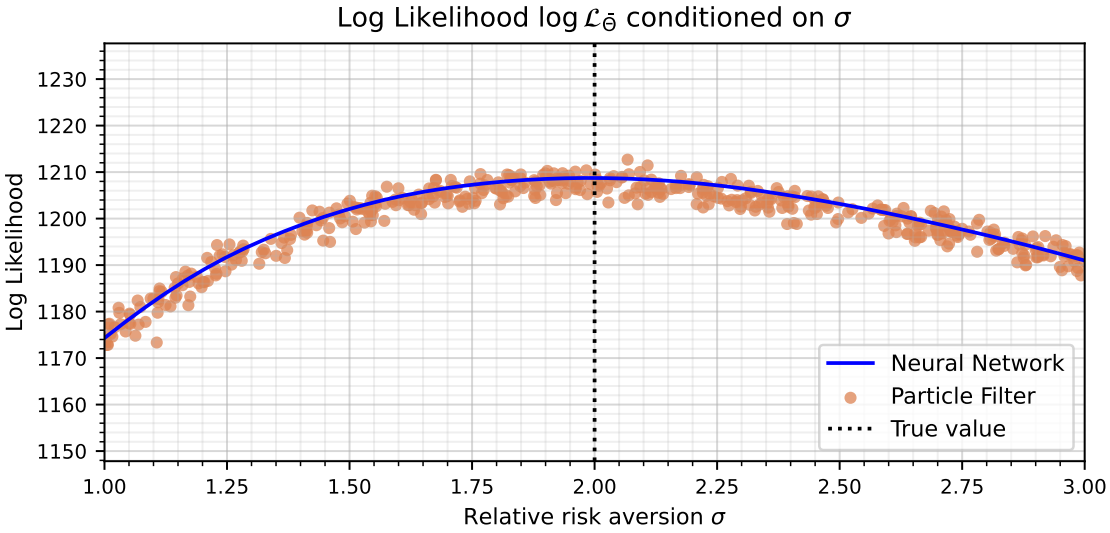

Figure 5: Accuracy in likelihood evaluations: NN particle filter vs. standard particle filter. The logarithm of the likelihood of the model as a function of the risk aversion parameter σ. The value of the fixed parameters is set to the middle of their bounds.

approximated with two alternative methods – varies with respect to the relative risk aversion, σ. The orange dots denote likelihood values evaluated with a standard particle filter using only 100 particles. As expected, this approximation is poor due to the limited number of particles used to deliberately exaggerate potential overfitting issues for instructional purposes. The solid blue line depicts the likelihood approximation using the NN particle filter, also trained with 100 particles for consistency.

Interestingly, the NN approximation cuts through the cloud of likelihood points generated by the particle filter, effectively filtering out the noise the filter introduces. Moreover, the parameter value at which the NN-approximated likelihood peaks closely aligns with the true parameter value, as indicated by the vertical dotted line. Appendix B shows how the approximated likelihood varies with respect to other parameters.

3.2 Proof of concept 2: A RANK model with the ZLB

We evaluate the effectiveness of our NN estimation strategy in recovering the true data-generating process of a stylized nonlinear RANK model augmented with a ZLB constraint, which is detailed in Appendix C.

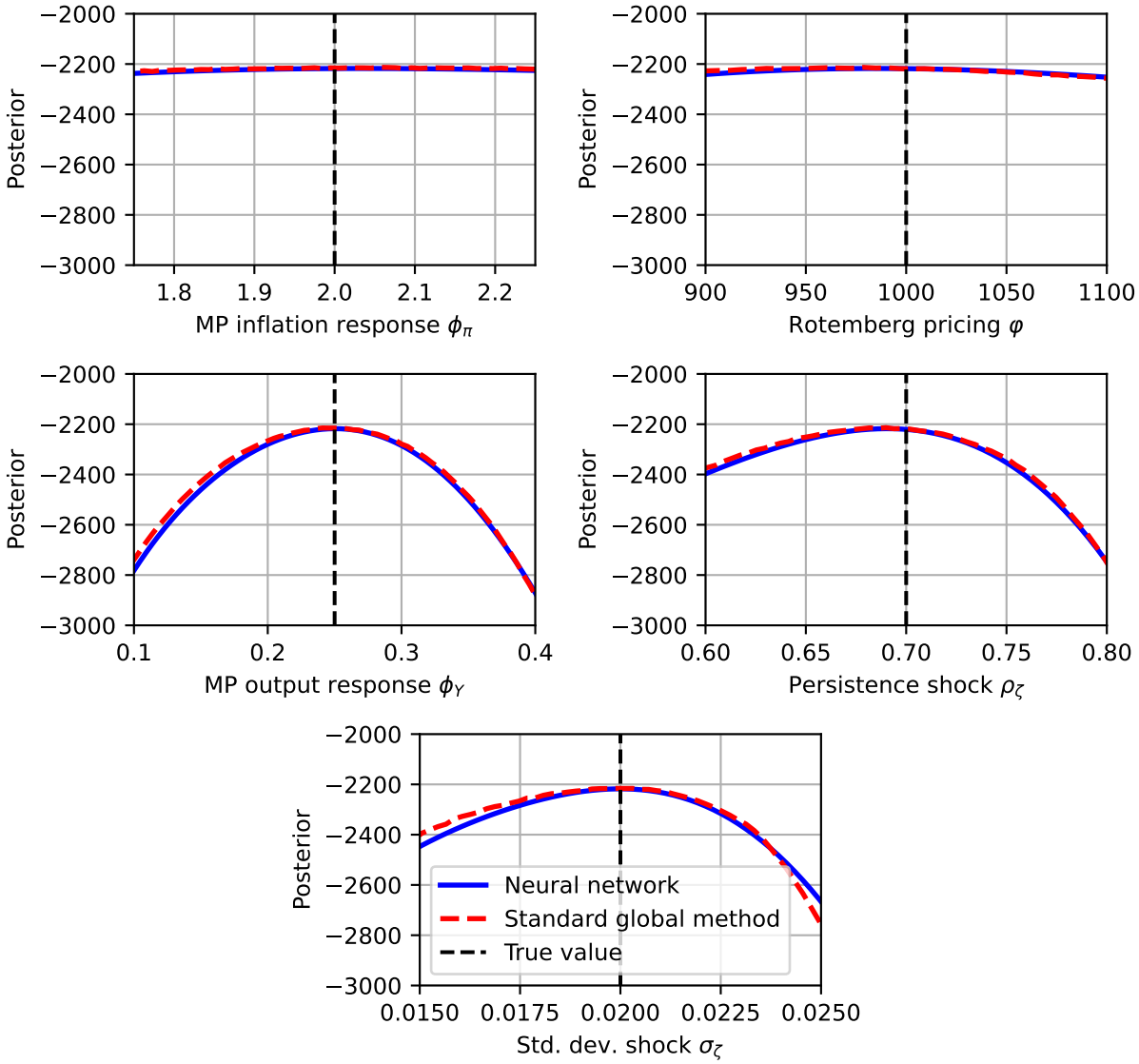

Figure 6: Posterior distribution. Comparison of the posterior distribution approximated with the NN method (solid blue), and a state-of-the-art global method (dashed red). The vertical black dash line denotes the true value of each parameter. The posterior distribution is evaluated at its median by varying one parameter at a time.

First, we simulate output growth, inflation, and the nominal interest rate for 500 periods from the model. Second, we estimate the model with our NN-based estimation method and with an alternative method combining a state-of-the-art global solution method (Richter et al., 2014) and the particle filter (Herbst and Schorfheide, 2015). While this alternative method is too slow for estimating more complex nonlinear models, it serves as a useful benchmark for evaluating the effectiveness of our method.

Our method demonstrates strong performance in recovering the parameters of the true data-generating process. The posterior median closely aligns with the true value and consistently falls within the 90% credible interval (shown in Appendix C). Moreover, the shape of the posterior closely resembles those obtained using the alternative method, as shown in Figure 6. However, the alternative method is significantly slower than our proposed NN method.

3.3 Proof of concept 3: A nonlinear HANK model

We now consider an off-the-shelf, stylized HANK model that features jointly idiosyncratic and aggregate risk. The economy consists of a continuum of households. The households choose consumption C_t^i, labor H_t^i, and assets B_t^i to maximize their utility:

$$E_0 \sum_{t=0}^{\infty} \beta^t \exp(\zeta_t) \left[\left(\frac{1}{1-\sigma} \right) \left(\frac{C_t^i}{A_t} \right)^{1-\sigma} - \chi \left(\frac{1}{1+\eta} \right) (H_t^i)^{1+\eta} \right], \tag{12}$$

where ζ_t is an aggregate preference shock, which follows an AR(1) process $\zeta_t = \rho_\zeta \zeta_{t-1} + \epsilon_t^\zeta$. A_t is the total factor productivity, which is nonstationary and introduces a trend in consumption. The budget constraint in real terms can be written as:

$$C_t^i + B_t^i = \tau_t \left(\frac{W_t}{A_t} \exp(s_t^i) H_t^i \right)^{1-\gamma_\tau} + \frac{R_{t-1}}{\Pi_t} B_{t-1}^i + Div_t^i, \tag{13}$$

where $Div_t^i = Div_t \exp(s_t^i)$ is the dividend in real terms paid out to household i and scaled by the level of its labor productivity, W_t is real wage, R_t is the gross nominal interest rate, and Π_t is the gross inflation rate. The taxation is progressive and governed by γ_τ and τ_t. The wage is divided by the productivity level to avoid a fiscal drag in taxation. The agents individual labor productivity s_t^i follows an AR(1) process in logs: $s_t^i = \rho_s s_{t-1}^i + \epsilon_t^{s,i}$. The agents face a borrowing limit $\underline{B}$ such that $B_t \geq \underline{B}$, in every period t.

Firms are monopolistically competitive and face nominal rigidities à la Rotemberg

with and φ denotes the price adjustment cost parameter. Total factor productivity, A_t, follows a nonstationary process with trend growth: $z_t = \rho_z z_{t-1} + \epsilon_t^z$. We denote the average growth rate of TFP as $\bar{a}$.

Monetary policy sets the interest rate according to a standard rule that is subject to a ZLB constraint. In symbols, $R_t = \max[1, R_t^n]$, where R_t^n is the notional or shadow interest rate that behaves as follows:

$$R_t^n = R \left(\frac{\Pi_t}{\Pi} \right)^{\theta_\Pi} \left(\frac{Y_t}{A_t Y} \right)^{\theta_Y} \exp(mp_t), \tag{14}$$

where Y_t stands for output in period t and R and Y denote the DSS equilibrium values of the nominal interest rate and output, respectively. Π denotes the inflation target, which is a parameter. The monetary shock follows an AR(1) process: $mp_t = \rho_m mp_{t-1} + \epsilon_t^m$. All shocks are Gaussian, with standard deviations denoted by σ_l, $l \in \{s, \zeta, z, m\}$. The full description of the model is provided in Appendix D.

We estimate this model in its nonlinear specification using simulated data generated from the same nonlinear model.[23] The objective is to assess whether our solution and estimation methods based on NNs allow us to retrieve the "true" parameter values used to generate the simulated data. The model is simulated for 144 periods by fixing the values of the parameters that will not be estimated at the values shown in the upper panel of Table 2 and the values of the parameters that will be estimated at the values shown in the column labelled "True" in the lower panel of the table. The observable variables are output growth, inflation, and nominal interest rate. In the lower panel of Table 3, we show the prior distributions for the parameters to estimate. Prior elicitation is presented in the next section when the model is estimated with actual data. The prior mean is set to the true value of the parameters.

[23]Little is known about the existence and uniqueness of rational expectations equilibria for nonlinear heterogeneous agent models. Acharya and Dogra (2020) derive the conditions for determinacy for HANK models with linearly approximated aggregate dynamics. We always check that these conditions are satisfied in our analysis. Additionally, highly persistent or highly volatile preference shocks may lead to nonexistence of a stable equilibrium in DSGE models with a ZLB constraint (Bianchi et al., 2021). We rule out these non-stationary equilibrium dynamics by appropriately truncating the prior for these two parameters in all the applications where this can be an issue.

Calibrated Parameter Values			
Parameters		Value	Target/Source
β	Discount factor	0.9975	4% nominal interest rate
η	Inverse Frisch elasticity	0.72	Chetty et al. (2011)
σ	Relative risk aversion	1	Log utility
$\bar{a}$	Average growth rate	1.0033	Real GDP growth = 0.33% (quarterly)
χ	Disutility labor	0.74	Labor supply is approximately 1
γ^{τ}	Tax progressivity	0.18	Heathcote et al. (2017)
D	DSS government debt	1.0	Wealth share=25% GDP (Kaplan et al., 2018)
Π	Inflation target	1.00625	Inflation target = 2.5% (annualized)
ρ_s	Persistence labor prod.	0.9	Share of borrowers = 34%
ρ_{ζ}	Persistence pref. shock	0.7	Frequency of ZLB = 15%

Estimation									
Par.	True	Prior					NN		
	Value	Type	Mean	Std	Lower Bound	Upper Bound	Posterior Median	5%	95%
Parameters affecting the DSS									
$100\sigma_s$	5.00	Trc.N	5.00	1.000	2.50	10.0	4.28	3.17	5.31
$\underline{B}$	-0.50	Trc.N	-0.50	0.010	-0.65	-0.35	-0.50	-0.54	-0.46
Other parameters									
φ	100	Trc.N	100	5.000	70	120	100	92	108
θ_{Π}	2.25	Trc.N	2.25	0.125	1.75	2.75	2.40	2.25	2.55
θ_Y	1.00	Trc.N	1.00	0.025	0.75	1.25	1.01	0.97	1.05
ρ_z	0.40	Trc.N	0.40	0.025	0.20	0.60	0.40	0.37	0.45
ρ_m	0.90	Trc.N	0.90	0.005	0.85	0.95	0.90	0.89	0.91
$100\sigma_{\zeta}$	1.50	Trc.N	1.50	0.100	1.00	2.00	1.45	1.34	1.57
$100\sigma_z$	0.40	Trc.N	0.40	0.100	0.30	0.60	0.36	0.32	0.40
$100\sigma_m$	0.06	Trc.N	0.06	0.010	0.05	0.20	0.06	0.05	0.07

Table 2: Parameters of the nonlinear HANK model. The upper panel shows the calibrated parameters for the nonlinear HANK model. The lower panel shows the true values used for simulating the model in Section 3.3 as well as the prior and the posterior distributions. Trc.N stands for truncated normal distribution.

The posterior moments of the estimated parameters are shown in the lower panel of Table 2. The estimation includes 10 structural parameters. The posterior median

is very close to the true value, underscoring the ability of the method to retrieve the true values of the parameters. In particular, the true value is mostly contained in the 90% credible interval for all parameters.

Note that the interest rate and the output targets (R and Y) in the monetary policy rule (14) are defined using their values at the DSS, which are affected by parameters we want to estimate (σ_s and $\underline{B}$). Therefore, as explained in Section 2, we need to train a separate NN for the task. Overall, we set up NNs to perform the following four tasks: (*i*) characterization of the deterministic steady state (DSS) equilibrium; (*ii*) approximation of the aggregate policy functions; (*iii*) approximation of the individual policy functions; (*iv*) approximation of the particle filter needed to evaluate the likelihood.

To approximate the mapping from the parameters to the DSS equilibrium values, we set the number of agents L to 100, implying a problem with 200 state variables – the four aggregate shocks are shut down in the DSS – and 2 pseudo-state variables (the parameters affecting the steady state – i.e., σ_s and $\underline{B}$). The batch size is set to 100. The NN has four layers. We use 128 neurons in each layer and rely on CELU and PReLU as activation functions.[24]

The next step is to solve the HANK model in its nonlinear specification with idiosyncratic and aggregate risk. We train two NNs to approximate the aggregate and idiosyncratic extended policy functions on the 204 states and 10 parameters.[25] In Appendix E.2, we provide more information about how these two NNs are trained. The NN trained to approximate the policy functions has five hidden layers. We use 128 neurons in each layer and rely on CELU and PReLU as activation functions.

[24]To solve the system of equations in the DSS, it is necessary to also solve for the individual agents' policy functions. Therefore, we employ a temporary NN that maps agents' individual states and parameters to their labor decisions. The loss function is conditioned on the Euler equation, adjusted for the borrowing limit and market clearing. Details are in Appendix E.1.

[25]The NN for the individual policy functions is also conditional on the individual agent's state variables. We capture the Karush-Kuhn-Tucker conditions due to the borrowing limit using the Fischer-Burmeister equation, as discussed in Appendix E. Alternatively, the constraints could be directly embedded in the NN architecture, e.g., in Azinovic and Žemlička (2023).

Finally, we train the NN particle filter. We draw from the parameter space and generate 15,000 likelihood values with a standard version of the particle filter. We use 75% of these values to train the NN (training sample) and the remaining values to validate the NN (validation sample). The NN features 64 nodes and four hidden layers. We use SiLU (Sigmoid Linear Unit) as an activation function. The NN is trained over 2,500 epochs with a batch size of 100.

After completing the training and validation of the NN, we can rapidly evaluate the likelihood for a large number of parameter values. Consequently, we are now able to execute the RWMH algorithm to approximate the moments of the posterior distribution for the parameters. We generate 1 million parameter draws (after a burn-in) to approximate the posterior distribution. We target an acceptance rate of $20\% - 40\%$. Details on how we conduct the posterior simulation step via RWMH are provided in Appendix A. It takes approximately three days with a GPU (Nvidia Tesla V100 32GB) to estimate the HANK model.

4 Estimation with actual data

In this section, we show how to apply our methodology to estimate the nonlinear HANK model introduced in Section 3.3 using actual data.

4.1 Data, calibrated parameters, and priors

To estimate the nonlinear HANK model, we use data on U.S. quarterly real per-capita GDP growth, U.S. quarterly GDP deflator inflation rate, and the shadow interest rate estimated by Wu and Xia (2016). The model's counterpart of the shadow rate is the notional rate, R_t^n – representing the interest rate that would have been set by the central bank in the absence of the ZLB constraint. All observables are in percent and annualized. Our analysis covers a sample period starting from the first quarter of 1990 and extending through the fourth quarter of 2019. To avoid degeneracy

problem in evaluating the likelihood, we introduce an i.i.d. measurement error for each observable.

While it is feasible to include cross-sectional variables directly in our estimation exercise, we prefer to introduce cross-section information via the prior distribution for the volatility of the idiosyncratic income risk. Research on likelihood estimation combining aggregate and cross-sectional data is still at an early stage, and we believe it is not desirable to have too many new features in this application of our method (e.g. Liu and Plagborg-Møller, 2023).

However, one might question if we can identify the idiosyncratic income risk, σ_s, from aggregate data without incorporating any information on the wealth distribution in the estimation process. It is indeed possible because due to the occasionally binding ZLB constraint, the level of idiosyncratic income risk can significantly influence macroeconomic volatility in the model. We illustrate these effects in Section 4.4.

We calibrate some parameters of the model, and their values are presented in the upper panel of Table 2. We set the discount factor $\beta = 0.9975$ to be consistent with a low interest rate environment characterizing our sample period. The inverse Frisch elasticity η is based on Chetty et al. (2011). The labor disutility χ is set such that labor supply is normalized to 1 in the steady state of a RANK model. The average growth rate of the economy, $\bar{a}$, is set to match the average real GDP growth rate over our sample period. Tax progressivity, γ^{τ}, follows the estimated value of Heathcote et al. (2017). We set the level of government debt equal to 25% of annual GDP in line with the estimate of liquid wealth in Kaplan et al. (2018). The persistence of the labor productivity ρ_s is set to 0.9 to target a share of borrowers around 1/3. The persistence of the preference shock ρ_ζ is set to 0.7; a value implying a ZLB frequency broadly in line with the data. The variance of the measurement error is set to equal 5% of the in-sample variance of the observable variables as in Gust et al. (2017).

The priors distribution is shown in the lower panel of Table 2. The prior distributions are centered at values that are broadly consistent with the parameter values

of linearized RANK models estimated in the literature. Regarding the parameters affecting the DSS, the prior mean of the borrowing limit is set to approximately 2 month labor income as in Fernández-Villaverde et al. (2023b), while the prior of idiosyncratic income volatility is set in line with the Gini coefficient on the US wealth inequality measured by the World Inequality Database over a period ranging from 1990 to 2019.

4.2 Neural networks based estimation

Given that the model is identical to the one of Section 3.3, we can use the same NNs trained to approximate the DSS and the extended policy functions. However, we must train the NN particle filter again due to the different dataset. Details regarding the specifications of the NNs and convergence statistics are provided in Appendix E.2.

After completing the training of these NNs, evaluating the likelihood of the nonlinear HANK model takes only seconds. Hence, we can conduct the RWMH exactly as discussed in Section 3.3, aiming for an acceptance rate of $20\% - 40\%$. Details on how to perform Bayesian estimation of this model are provided in Appendix A.

The posterior moments are shown in Table 3. Bayesian updating is modest for most parameters despite our prior standard deviation being fairly large and the average acceptance rate of the RWMH being around 27%. Since the prior mean for these parameters was set to reflect typical estimates from linearized RANK models, we interpret this finding as suggesting that heterogeneity and nonlinearities do not lead to substantial revisions to the estimated values of those parameters.[26]

The Bayesian updating is somewhat stronger for the standard deviation of the idiosyncratic income shock, σ_s. The posterior median is 2 percentage points above the prior median. The interactions between heterogeneity and nonlinearities, particularly the nonlinear ZLB constraint, play a critical role in identifying this parameter. As

[26]Bayesian updating might be stronger if more observable variables (e.g., consumption or wages) are used in estimation but this requires extending the HANK model, which is feasible but beyond the scope of this methodological paper.

Estimation								
Par.	Prior					NN		
	Type	Mean	Std	Lower Bound	Upper Bound	Posterior Median	5%	95%
Parameters affecting the DSS								
$100\sigma_s$	Trc.N	5.00	1.000	2.50	10.0	7.04	5.67	8.10
$\underline{B}$	Trc.N	-0.50	0.010	-0.65	-0.35	-0.50	-0.54	-0.46
Other parameters								
φ	Trc.N	100	5.000	70	120	101	94	107
θ_Π	Trc.N	2.25	0.125	1.75	2.75	2.43	2.20	2.67
θ_Y	Trc.N	1.00	0.025	0.75	1.25	0.96	0.92	1.00
ρ_z	Trc.N	0.40	0.025	0.2	0.6	0.43	0.39	0.47
ρ_m	Trc.N	0.90	0.005	0.85	0.95	0.91	0.90	0.91
$100\sigma_\zeta$	Trc.N	1.50	0.100	1.00	2.00	1.22	1.10	1.33
$100\sigma_z$	Trc.N	0.40	0.100	0.30	0.60	0.47	0.43	0.53
$100\sigma_m$	Trc.N	0.06	0.010	0.05	0.20	0.15	0.14	0.16

Table 3: Prior and Posterior distributions. Prior and posterior moments for the estimation with real data. The prior type indicates the prior density function. Trc.N stands for truncated normal distribution.

we will show, in our model, an increase in the level of idiosyncratic income risk (σ_s) heightens the ZLB risk and frequency, leading to greater volatility in output and inflation, which are observed in estimation.

4.3 How good is what we got?

The empirical fit of the model is quite promising given the stylized nature of the HANK model we have estimated. In Table 4, we compare the second moment of the observables in the estimated HANK model and in the data. The standard deviation of output growth, inflation, and the nominal interest rate in the model are quite close to the ones measured in the data.

Standard deviations			Autocorrelations			Avg. Gini coef.		
	Model	Data		Model	Data		Model	Data
GDP	0.6947	0.5831	GDP	0.1355	0.4050			
Inflation	1.1511	0.9045	Inflation	0.8146	0.5456	Wealth	0.8793	0.8410
FFR	2.561	2.7537	FFR	0.7219	0.9707			

Table 4: Moments comparison: model vs. data. The table reports the standard deviations (left panel) and the autocorrelation coefficients (middle panel) in the model and in the U.S. data for the three observable variables: real GDP growth (GDP), GDP deflator growth (Inflation), and the federal funds rate (FFR). The right panel of the table shows the average Gini coefficient in the model and in the U.S. data (World Inequality Database). The model-implied moments are obtained by simulating the estimated model for 1,000,000 periods and with parameters set at the posterior median. The empirical moments are computed over the sample period used in estimation (1990:Q1-2019:Q4).

The match is somewhat unsatisfactory, when it comes to the serial correlation of the three observables. This finding underscores the need for extensions of the HANK model aimed at enhancing its endogenous persistence mechanism. Introducing additional nominal and real frictions could be an effective strategy for improving the empirical fit of the stylized nonlinear HANK model we have estimated.

Acharya et al. (2023) estimate a HANK model using likelihood methods and find that its empirical performance is inferior to that of a state-of-the-art RANK model. One potential reason for this subpar performance is the challenge of accurately estimating certain parameters of the HANK model, particularly those affecting its DSS, due to the high computational cost of recomputing it. Our NN estimation method addresses this challenge by efficiently solving the steady state equilibrium of HANK models, allowing the estimation of these parameters. In contrast to Acharya et al. (2023), we estimate the fully nonlinear version of a HANK model, including the ZLB constraint.

In Table 4, we also present the average Gini index of the wealth distribution in our model, which we estimated without observing any cross-sectional data, alongside the corresponding index measured by the World Inequality Database for the US over the same sample period of our estimation. The two coefficients are quite close.

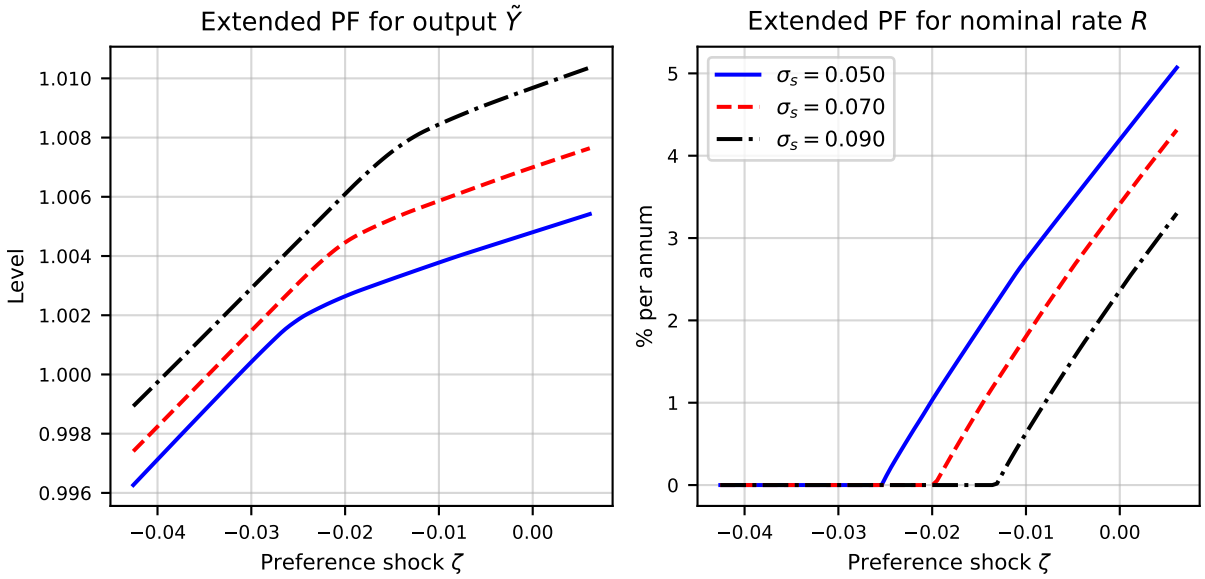

Figure 7: Aggregate policy functions of the nonlinear HANK model. The figure illustrates how equilibrium output (left) and nominal interest rate (right) vary as a function of some realizations of the preference shock, ζ_t. The three colored lines denote the two policy functions when the standard deviation of the idiosyncratic shock to households' income, σ_s, is set to 0.050 (prior mean), 0.070 (posterior median), and 0.090. The value of the other parameters is set to their posterior median.

4.4 Interactions between heterogeneity and nonlinearities

In this section, we emphasize the importance of preserving aggregate nonlinearity when estimating a HANK model. Figure 7 illustrates how output (left chart) and nominal interest rate (right chart) vary with preference shocks (x-axis) in the estimated model. The three lines represent different levels of volatility of idiosyncratic shocks to income, with the red dashed line indicating the value consistent with the posterior median of this parameter (our estimate). The blue solid line represents their value at the prior mean for the volatility of idiosyncratic shocks. When the preference shock is negative, households save more, leading to a contraction in aggregate demand. Consequently, output declines, explaining the upward slope of the aggregate policy function on the left side of the plot.

For sufficiently negative realizations of preference shocks, the slope of the aggre-

gate policy functions for output becomes steeper. The kink in this function occurs at the point where the negative preference shock is large enough to cause a contraction in aggregate demand leading to the ZLB constraint becoming binding. The binding ZLB constraint exacerbates the fall in aggregate demand by constraining monetary policy, resulting in a more pronounced decline in output.

When the idiosyncratic risk is higher (moving from the blue solid line to the black dashed-dotted line), output increases as households want to work more in response to heightened income risk. The effects of a larger idiosyncratic risk on output are less pronounced when the interest rate is constrained by the ZLB, indicated to the left of the kink points in the left chart. In this scenario, an increase in idiosyncratic risk prompts households to save more, thereby reducing the real interest rate and prolonging the expected duration of the ZLB. This, in turn, has contractionary effects on real activity. Consequently, the positive effects of a larger idiosyncratic risk on the level of output are weaker when the economy is at the ZLB.

A higher idiosyncratic income risk increases the frequency of the ZLB constraint. This effect is evident by observing the shift in the kink point of the policy function for the interest rate to the right as idiosyncratic income risk rises (i.e., moving from the solid blue line to the black dashed-dotted line on the right chart of Figure 7). This shift in the kink indicates that it would require less severe (negative) realizations of the preference shock to drive the nominal interest rate to the ZLB. Since smaller realizations of the preference shocks are more likely to occur, the risk of encountering the ZLB increases when idiosyncratic risk becomes larger.

Idiosyncratic risk and aggregate output volatility. The ZLB constraint interacts significantly with idiosyncratic income risk, amplifying the volatility of macroeconomic variables. In Figure 8, we illustrate that a heightened risk of encountering the ZLB amplifies output volatility in our nonlinear HANK model. This amplification is represented by the vertical distance between the blue solid line and the red dashed-

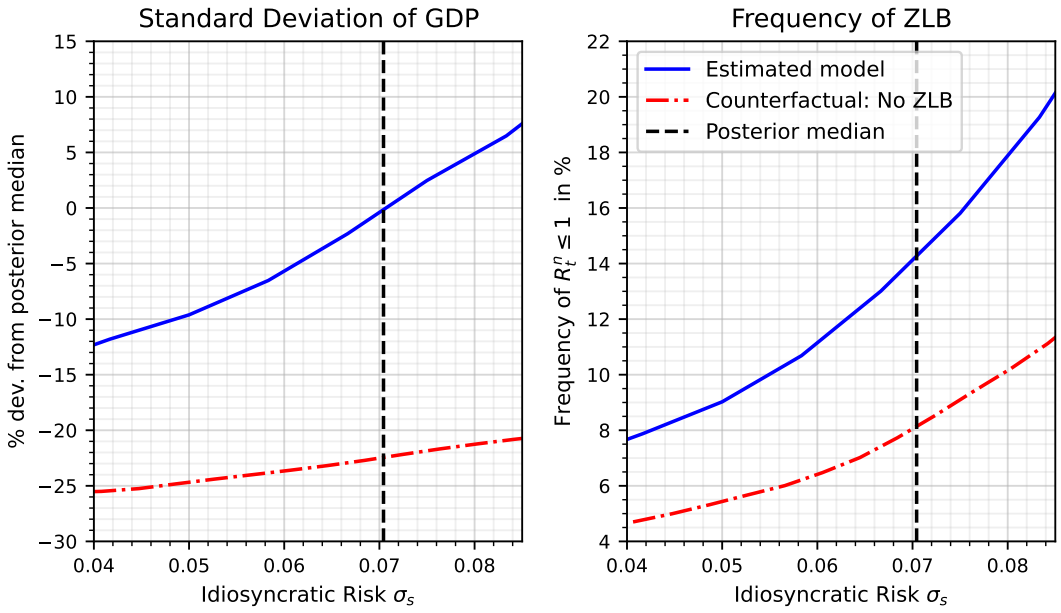

Figure 8: Volatility of output and ZLB frequency. *Left chart*: Volatility of output as a function of the idiosyncratic risk in the estimated model (blue solid line) and in the counterfactual case in which the ZLB is not enforced (red dashed-dotted line). The reported volatility is expressed in percentage deviations from its posterior median. *Right chart*: Frequency of ZLB episodes as a function of the idiosyncratic risk in the estimated model (blue solid line). Frequency of episodes when the interest rate turns negative in the counterfactual case in which the ZLB is not enforced (red dashed-dotted line). In both charts, the vertical black dashed line marks the value of the posterior median of the standard deviation of idiosyncratic shocks. The value of the other parameters is set to their posterior median. The standard deviation of output (in deviations from balanced growth) and the frequency of the ZLB and the frequency of negative interest rate are computed by simulation (100,000 periods).

dotted line. The ZLB risk exacerbates output volatility because the effectiveness of monetary policy in stabilizing the economy at the ZLB is diminished. The vertical black line, representing the estimated volatility of idiosyncratic risk, shows that approximately 22 percentage points of GDP volatility are explained by the occasionally binding ZLB constraint.

Importantly, the amplification effect of the ZLB on output volatility intensifies with higher levels of idiosyncratic income risk. In Figure 8, we observe that the vertical distance between the solid blue line and the red dashed-dotted line expands as the level of idiosyncratic risk increases along the x-axis.

To delve into the rationale behind this finding, we examine how the frequency of

encountering the ZLB changes with the level of idiosyncratic risk. As shown by the blue solid line on the right chart, as idiosyncratic risk increases, the ZLB becomes binding more often. This occurs because higher idiosyncratic risk reduces the nominal interest rate, leading to a heightened frequency of encountering the ZLB. With the ZLB binding more frequently, recessions and deflation episodes become more severe, ultimately amplifying macroeconomic volatility.

If we were to relax the ZLB constraint, the frequency of negative interest rates would decrease by roughly 6 percentage points in the estimated model. This difference is illustrated by the vertical distance between the solid blue line and the red dashed-dotted line at the posterior median on the right chart.

The relationship between the frequency of negative interest rates and the level of idiosyncratic risk remains positive – albeit somewhat less strong – even when the ZLB constraint is not enforced, as indicated by the red dashed-dotted line on the right chart. This is because higher idiosyncratic income risk prompts households to increase precautionary savings, thereby depressing the nominal interest rate and making it more likely for nominal interest rates to fall into the negative territory.

An important takeaway is that the parameter σ_s, which governs the level of idiosyncratic risk, can be identified using the volatility of output and inflation in nonlinear HANK models featuring the ZLB constraint. Therefore, observing any cross-sectional variables in estimation is not strictly necessary to identify the volatility of idiosyncratic income risk in our application.

The deflationary bias. Another advantage of preserving the nonlinear features of a HANK model is the ability to assess the deflationary bias resulting from the occasionally binding ZLB constraint. As shown by Fernández-Villaverde et al. (2023b), in a nonlinear HANK model, the deflationary bias is influenced by the volatility of idiosyncratic income shocks. Our estimated nonlinear HANK model predicts a deflationary bias of around 30 basis points, measured as the difference between the DSS

inflation rate, Π, and the average inflation rate in the model – often termed stochastic or risky steady-state inflation rate.

Our results should be interpreted as a conservative estimate of the deflationary bias, given the limited degree of heterogeneity in our model – for example, agents only trade one asset. HANK models with more sophisticated heterogeneity could exhibit much more significant interactions, potentially introducing dynamics beyond the scope of our simple HANK model. Exploring this avenue further could be an interesting direction for future research.

5 Concluding remarks

We have introduced a novel method based on machine learning techniques to estimate nonlinear dynamic general equilibrium models with heterogeneous agents. Our approach offers three significant advantages: i) it enables the estimation of models featuring a large set of state variables, ii) it can capture nonlinear dynamics, such as the ZLB constraint or borrowing limits, and their interactions with the decisions of heterogeneous agents, iii) it allows the estimation of parameters affecting the model's steady-state equilibrium. Future applications of the proposed methodology include nonlinear models with a very large set of state variables, such as models with multiple sectors or multiple countries.

We applied our techniques to estimate the likelihood of a nonlinear HANK model using US data. Standard macroeconomic data can identify the degree of idiosyncratic risk, as this risk influences the frequency of ZLB episodes and, consequently, aggregate volatility. Despite its relatively stylized nature, the estimated nonlinear HANK model exhibits a very promising ability to match key moments in the data as well as the average level of wealth inequality. Our exercise also highlights how the model can be improved to achieve an even better empirical performance.

References

Acharya, S., Chen, W., Del Negro, M., Dogra, K., Gleich, A., Goyal, S., Matlin, E., Lee, D., Sarfati, R., Sengupta, S., 2023. Estimating HANK for central banks. FRB of New York Staff Report 1071.

Acharya, S., Dogra, K., 2020. Understanding hank: Insights from a prank. Econometrica 88, 1113–1158.

Ahn, S., Kaplan, G., Moll, B., Winberry, T., Wolf, C., 2018. When inequality matters for macro and macro matters for inequality. NBER Macroeconomics Annual 32, 1–75.

Algan, Y., Allais, O., Den Haan, W.J., 2008. Solving heterogeneous-agent models with parameterized cross-sectional distributions. Journal of Economic Dynamics and Control 32, 875–908.

Aruoba, B., Cuba-Borda, P., Higa-Flores, K., Schorfheide, F., Villalvazo, S., 2021. Piecewise-linear approximations and filtering for dsge models with occasionally-binding constraints. Review of Economic Dynamics 41, 96–120.

Aruoba, B., Cuba-Borda, P., Schorfheide, F., 2018. Macroeconomic dynamics near the zlb: A tale of two countries. The Review of Economic Studies 85, 87–118.

Atkinson, T., Richter, A.W., Throckmorton, N.A., 2020. The zero lower bound and estimation accuracy. Journal of Monetary Economics 115, 249–264.

Auclert, A., 2019. Monetary Policy and the Redistribution Channel. American Economic Review 109, 2333–2367.

Auclert, A., Bardóczy, B., Rognlie, M., Straub, L., 2021. Using the sequence-space jacobian to solve and estimate heterogeneous-agent models. Econometrica 89, 2375–2408.

Auclert, A., Rognlie, M., Straub, L., 2020. Micro Jumps, Macro Humps: Monetary Policy and Business Cycles in an Estimated HANK Model. NBER Working Papers 26647. National Bureau of Economic Research, Inc.

Auclert, A., Rognlie, M., Straub, L., 2023. The Intertemporal Keynesian Cross. Journal of Political Economy, forthcoming.

Azinovic, M., Gaegauf, L., Scheidegger, S., 2022. Deep equilibrium nets. International Economic Review 63, 1471–1525.

Azinovic, M., Žemlička, J., 2023. Intergenerational Consequences of Rare Disasters.

Bayer, C., Born, B., Luetticke, R., 2024a. Shocks, frictions, and inequality in us business cycles. American Economic Review 114, 1211–47.

Bayer, C., Luetticke, R., Pham-Dao, L., Tjaden, V., 2019. Precautionary savings, illiquid assets, and the aggregate consequences of shocks to household income risk. Econometrica 87, 255–290.

Bayer, C., Luetticke, R., Weiss, M., Winkelmann, Y., 2024b. An Endogenous Grid-point Method for Distributional Dynamics. mimeo.

Bhandari, A., Bourany, T., Evans, D., Golosov, M., 2023. A Perturbational Approach for Approximating Heterogeneous Agent Models. Working Paper 31744. National Bureau of Economic Research.

Bianchi, F., Melosi, L., Rottner, M., 2021. Hitting the elusive inflation target. Journal of Monetary Economics 124, 107–122.

Bilbiie, F.O., 2020. The new keynesian cross. Journal of Monetary Economics 114, 90–108.

Bilbiie, F.O., Primiceri, G., Tambalotti, A., 2023. Inequality and Business Cycles. Working Paper 31729. National Bureau of Economic Research.

Boppart, T., Krusell, P., Mitman, K., 2018. Exploiting mit shocks in heterogeneous-agent economies: the impulse response as a numerical derivative. Journal of Economic Dynamics and Control 89, 68–92.

Challe, E., Matheron, J., Ragot, X., Rubio-Ramirez, J.F., 2017. Precautionary saving and aggregate demand. Quantitative Economics 8, 435–478.

Chetty, R., Guren, A., Manoli, D., Weber, A., 2011. Are micro and macro labor supply elasticities consistent? a review of evidence on the intensive and extensive margins. American Economic Review 101, 471–475.

Den Haan, W.J., Judd, K.L., Juillard, M., 2010. Computational suite of models with heterogeneous agents: Incomplete markets and aggregate uncertainty. Journal of Economic Dynamics and Control 34, 1–3.

Den Haan, W.J., Rendahl, P., Riegler, M., 2017. Unemployment (Fears) and Deflationary Spirals. Journal of the European Economic Association 16, 1281–1349.

Duarte, V., Duarte, D., Silva, D., 2024. Machine Learning for Continuous-Time Finance. Working Paper 10909. CESifo.

Duarte, V., Fonseca, J., 2024. Global identification with gradient-based structural estimation.

Ebrahimi Kahou, M., Fernández-Villaverde, J., Perla, J., Sood, A., 2021. Exploiting symmetry in high-dimensional dynamic programming. Working Paper 28981. National Bureau of Economic Research.

Fernández-Villaverde, J., Hurtado, S., Nuño, G., 2023a. Financial frictions and the wealth distribution. Econometrica 91, 869–901.

Fernández-Villaverde, J., Marbet, J., Nuño, G., Rachedi, O., 2023b. Inequality and the Zero Lower Bound. Working Paper 31282. National Bureau of Economic Research.

Fernández-Villaverde, J., Nuño, G., Sorg-Langhans, G., Vogler, M., 2020. Solving high-dimensional dynamic programming problems using deep learning.

Fernández-Villaverde, J., Rubio-Ramírez, J.F., 2007. Estimating Macroeconomic Models: A Likelihood Approach. The Review of Economic Studies 74, 1059–1087.

Gornemann, N.M., Kuester, K., Nakajima, M., 2021. Doves for the Rich, Hawks for the Poor? Distributional Consequences of Systematic Monetary Policy. Opportunity and Inclusive Growth Institute Working Papers 50. Federal Reserve Bank of Minneapolis.

Gorodnichenko, Y., Maliar, L., Maliar, S., Naubert, C., 2021. Household Savings and Monetary Policy under Individual and Aggregate Stochastic Volatility. CEPR Discussion Papers 15614.

Guerrieri, V., Lorenzoni, G., 2017. Credit Crises, Precautionary Savings, and the Liquidity Trap*. The Quarterly Journal of Economics 132, 1427–1467.

Gust, C., Herbst, E., López-Salido, D., Smith, M.E., 2017. The Empirical Implications of the Interest-Rate Lower Bound. American Economic Review 107, 1971–2006.

Han, J., Yang, Y., E, W., 2021. DeepHAM: A Global Solution Method for Heterogeneous Agent Models with Aggregate Shocks.

Heathcote, J., Storesletten, K., Violante, G.L., 2017. Optimal tax progressivity: An analytical framework. The Quarterly Journal of Economics 132, 1693–1754.

Herbst, E.P., Schorfheide, F., 2015. Bayesian estimation of DSGE models. Princeton University Press.

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

Kaplan, G., Moll, B., Violante, G.L., 2018. Monetary policy according to hank. American Economic Review 108, 697–743.

Kaplan, G., Violante, G.L., 2018. Microeconomic heterogeneity and macroeconomic shocks. Journal of Economic Perspectives 32, 167–94.

Krusell, P., Smith, A., 1998. Income and wealth heterogeneity in the macroeconomy. Journal of Political Economy 106, 867–896.

Lee, D., 2020. Quantitative Easing and Inequality. mimeo.

Lin, A., Peruffo, M., 2024. Aggregate Uncertainty, HANK, and the ZLB. Working Paper Series 2911. European Central Bank.

Liu, L., Plagborg-Møller, M., 2023. Full-information estimation of heterogeneous agent models using macro and micro data. Quantitative Economics 14, 1–35.

Luetticke, R., 2021. Transmission of monetary policy with heterogeneity in household portfolios. American Economic Journal: Macroeconomics 13, 1–25.

Maliar, L., Maliar, S., 2020. Deep learning: Solving hanc and hank models in the absence of krusell-smith aggregation.

Maliar, L., Maliar, S., Winant, P., 2021. Deep learning for solving dynamic economic models. Journal of Monetary Economics 122, 76–101.

McKay, A., Nakamura, E., Steinsson, J., 2016. The power of forward guidance revisited. American Economic Review 106, 3133–58.

McKay, A., Reis, R., 2016. The role of automatic stabilizers in the u.s. business cycle. Econometrica 84, 141–194.

Norets, A., 2012. Estimation of dynamic discrete choice models using artificial neural network approximations. Econometric Reviews 31, 84–106.

Oh, H., Reis, R., 2012. Targeted transfers and the fiscal response to the great recession. Journal of Monetary Economics 59, S50–S64.

Ottonello, P., Winberry, T., 2020. Financial heterogeneity and the investment channel of monetary policy. Econometrica 88, 2473–2502.

Ravn, M.O., Sterk, V., 2017. Job uncertainty and deep recessions. Journal of Monetary Economics 90, 125–141.

Ravn, M.O., Sterk, V., 2020. Macroeconomic Fluctuations with HANK & SAM: an Analytical Approach. Journal of the European Economic Association 19, 1162–1202.

Reiter, M., 2009. Solving heterogeneous-agent models by projection and perturbation. Journal of Economic Dynamics and Control 33, 649–665.

Richter, A.W., Throckmorton, N.A., Walker, T.B., 2014. Accuracy, speed and robustness of policy function iteration. Computational Economics 44, 445–476.

Rottner, M., 2023. Financial crises and shadow banks: A quantitative analysis. Journal of Monetary Economics 139, 74–92.

Schaab, A., 2020. Micro and Macro Uncertainty. mimeo.

Scheidegger, S., Bilionis, I., 2019. Machine learning for high-dimensional dynamic stochastic economies. Journal of Computational Science 33, 68–82.

Valaitis, V., Villa, A.T., 2024. A machine learning projection method for macro-finance models. Quantitative Economics 15, 145–173.

Winberry, T., 2021. Lumpy investment, business cycles, and stimulus policy. American Economic Review 111, 364–96.

Wu, J.C., Xia, F.D., 2016. Measuring the Macroeconomic Impact of Monetary Policy at the Zero Lower Bound. Journal of Money, Credit and Banking 48, 253–291.

Online Appendix

A Neural network based Bayesian estimation

The following Random Walk Metropolis-Hastings (RWMH) algorithm can be used to estimate the parameters $\tilde{\Theta}$, where we use the NN approximation of the likelihood function. To integrate our NN approach for Bayesian estimation, we adapt the RWMH algorithm of Atkinson et al. (2020).[27] The approach is as follows:

1. Initialize the algorithm: Obtain a first proposal density from which the parameters: $\tilde{\Theta}$ are drawn, as follows:

   Draw from the prior distribution a candidate vector $\tilde{\Theta}^{New}$. Evaluate the log-likelihood of the candidate vector using the trained NN for the particle filter $\log \mathcal{L}_{\bar{\Theta}}\left(\mathbb{Y}_{1:T}|\tilde{\Theta}^{New}\right)$ and combine it with the prior to obtain the log-posterior $\log g(\mathbb{Y}_{1:T}|\tilde{\Theta}^{New})$.

   Repeat these steps N^{init} times and collect all draws as $\tilde{\Theta}^{init}$. Construct an approximation for the covariance matrix by selecting draws where the likelihood is above the 90% quantile, denoted by $\hat{\Theta}$, and subtract the mean $\ddot{\Theta} = \hat{\Theta} - \bar{\hat{\Theta}}$. Calculate the covariance matrix $\Sigma = (\ddot{\Theta}\ddot{\Theta}')/(0.1N^{init})$. Set $\breve{\Theta}$ to equal the mode, the draw associated with the highest likelihood.

2. Run the RWMH algorithm as burn-in. Repeat the steps below N^{Burn} times:

   Draw a new parameter vector $\tilde{\Theta}^{New}$ from the proposal density to evaluate the log-posterior. The proposal density is a multivariate normal distribution with mean vector $\breve{\Theta}$ and covariance matrix $c\Sigma$, where c is set to achieve an acceptance rate between 20 and 40%.

[27] See also Herbst and Schorfheide (2015) for more information on RWMH.

Evaluate the log-likelihood of the parameter vector using the trained NN for the particle filter $\log \mathcal{L}_{\bar{\Theta}}\left(\mathbb{Y}_{1:T}|\tilde{\Theta}^{New}\right)$ and combine it with the prior to obtain the log-posterior $\log g(\mathbb{Y}_{1:T}|\tilde{\Theta}^{New})$. Accept the draw if $\exp(\log g(\mathbb{Y}_{1:T}|\tilde{\Theta}^{New}) - \log g(\mathbb{Y}_{1:T}|\tilde{\Theta}))$ is larger than the draw from a standard uniform distribution. If the draw is accepted, the mean of the proposal density is updated to $\breve{\Theta} = \tilde{\Theta}^{New}$.

3. Use these N^{Burn} draws to get an updated proposal density:

   Keep only the last 75% of draws, denoted as $\hat{\Theta}^b$. Calculate the deviations of each draw from the mean $\ddot{\Theta}^b = \hat{\Theta}^b - \bar{\hat{\Theta}}^b$. Calculate the covariance matrix $\Sigma^b = (\ddot{\Theta}^b \ddot{\Theta}^{b\prime})/(0.75N^{Burn})$. Calculate the mode $\breve{\Theta}^b$ and update the mean of the proposal density $\breve{\Theta} = \breve{\Theta}^b$.

4. Run the RWMH algorithm to obtain the posterior:

   Use the proposal density from above and repeat the steps below N^{final} times: Draw a new parameter vector $\tilde{\Theta}^{New}$ from the proposal density to evaluate the log-posterior. The proposal density is a multivariate normal distribution with mean vector $\breve{\Theta}$ and covariance matrix $c\Sigma$, where c is again tuned to have an acceptance rate that is between 20-40%. Evaluate the log-likelihood of the parameter vector using the trained NN for the particle filter $\log \mathcal{L}^{NN}_{\bar{\Theta}}\left(\mathbb{Y}_{1:T}|\tilde{\Theta}^{New}\right)$ and combine it with the prior to obtain the log posterior $\log g(\mathbb{Y}_{1:T}|\tilde{\Theta}^{New})$. Accept the draw if $\exp(\log g(\mathbb{Y}_{1:T}|\tilde{\Theta}^{New}) - \log g(\mathbb{Y}_{1:T}|\tilde{\Theta}))$ is larger than the draw from a standard uniform distribution. If the draw is accepted, the mean of the proposal density is updated to $\breve{\Theta} = \tilde{\Theta}^{New}$.

5. N^{final} draws characterize the posterior distribution.

B Linearized DSGE model

The model is a prototypical three-equation New Keynesian DSGE model with a TFP shock. The model is log-linearized around its unique steady-state equilibrium, which results in the following equations.

$$\hat{Y}_t = E_t\hat{Y}_{t+1} - \sigma^{-1}\left(\hat{R}_t - E_t\hat{\Pi}_{t+1}\right), \tag{15}$$

$$\hat{\Pi}_t = \kappa(\hat{Y}_t - \hat{Y}_t^*) + \beta E_t\hat{\Pi}_{t+1}, \tag{16}$$

$$\hat{R}_t = \phi_\Pi\hat{\Pi}_t + \phi_Y\hat{X}_t, \tag{17}$$

$$\hat{Y}_t^* = \omega\hat{A}_t, \tag{18}$$

$$\hat{A}_t = \rho_A\hat{A}_{t-1} + \sigma_A\epsilon_t^A, \tag{19}$$

where we define $\omega = (1+\eta)/(\eta+\sigma)$ as well as $\kappa = (1-\phi)(1-\phi\beta)(\sigma+\eta)/\phi$. Output is defined as $\hat{Y}_t = (Y_t - Y)/Y$, inflation as $\hat{\Pi}_t = \Pi_t - \Pi$, nominal rate as $\hat{R}_t = R_t - R$, output in the flex price economy as $\hat{Y}_t^* = (Y_t^* - Y^*)/Y^*$, TFP as $\hat{A}_t = (A_t - Y)/A$, and the output gap as $\hat{X}_t = \hat{Y}_t - \hat{Y}_t^*$. The shock ϵ_t^A follows a standard normal distribution.

After some manipulations, the model equations can be written as:

$$\hat{X}_t = E_t\hat{X}_{t+1} - \sigma^{-1}\left(\phi_\Pi\hat{\Pi}_t + \phi_Y\hat{X}_t - E_t\hat{\Pi}_{t+1} - \hat{R}_t^*\right), \tag{20}$$

$$\hat{\Pi}_t = \kappa\hat{X}_t + \beta E_t\hat{\Pi}_{t+1}, \tag{21}$$

$$\hat{R}_t^* = \rho_A\hat{R}_{t-1}^* + \sigma(\rho_A - 1)\omega\sigma_A\epsilon_t^A, \tag{22}$$

where $\hat{R}_t^*$ is the natural rate of interest, which follows an exogenous process derived from the TFP process.

We set up the NN as in section B.1 and train it by minimizing the residual error constructed from equations (20) and (21). The law of motion of the exogenous state variable, described in equation (22), is used for sampling.

The analytical solution for the output gap $\hat{X}_t$ and inflation $\hat{\Pi}_t$, which can be

derived with the method of undetermined coefficients, is given as:

$$\hat{X}_t = \frac{1-\beta\rho_A}{(\sigma(1-\rho_A)+\theta_Y)(1-\beta\rho_A)+\kappa(\theta_\Pi-\rho_A)}\hat{R}_t^* \tag{23}$$

$$\hat{\Pi}_t = \frac{\kappa}{(\sigma(1-\rho_A)+\theta_Y)(1-\beta\rho_A)+\kappa(\theta_\Pi-\rho_A)}\hat{R}_t^* \tag{24}$$

The analytical solution underscores the idea of extended policy functions, which are simultaneously conditioned on the parameters and state variables. The analytical solution for the output gap and inflation depends on both the state variable $\hat{R}_t^*$ and the parameters.

B.1 Mapping of the model to the general framework

We can map the linearized NK model in the general form of the solution method outlined in section 2.2. The state variable, the structural shock, and the control variables of the model are:

$$\mathbb{S}_t = \left\{\hat{R}_t^*\right\}, \qquad \text{and} \qquad \nu_t = \left\{\epsilon_t^A\right\}, \qquad \text{and} \qquad \psi_t = \left\{\hat{X}_t, \hat{\Pi}\right\}. \tag{25}$$

The parameters of the model are divided into calibrated ($\bar{\theta}$), which is an empty set in this application, and estimated ones ($\tilde{\theta}$):

$$\tilde{\Theta} = \{\beta, \sigma, \eta, \phi, \theta_\Pi, \theta_Y, \rho_A, \sigma_A\}. \tag{26}$$

The NN ψ_{NN} is trained to determine the output gap and inflation:

$$\begin{pmatrix} \hat{X}_t \\ \hat{\Pi}_t \end{pmatrix} = \psi_{NN}\left(\mathbb{S}_t, \tilde{\Theta}|\bar{\Theta}\right). \tag{27}$$

B.2 Neural network for inflation

Figure 9 shows the extended policy function for inflation that is conditioned on the parameters.

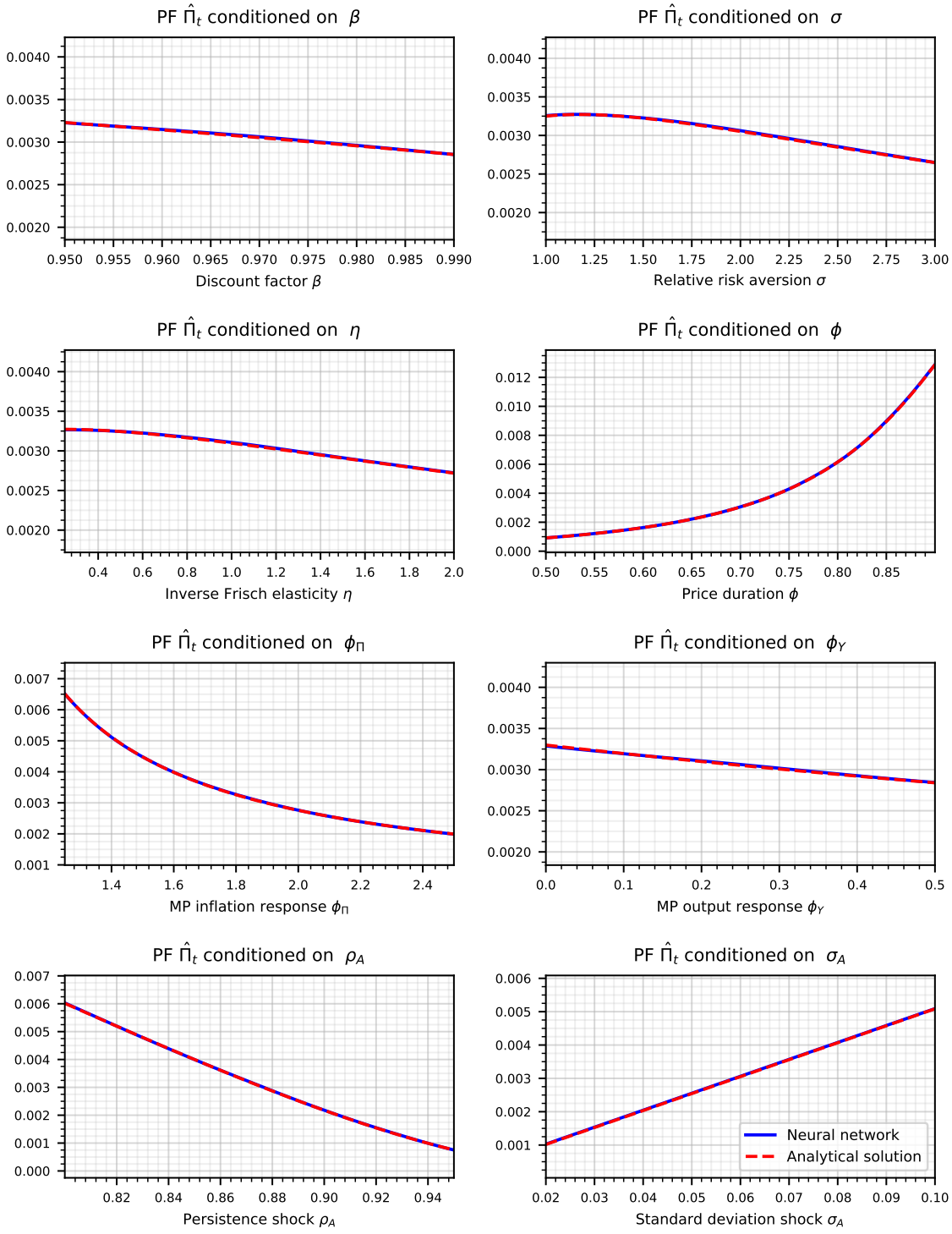

Figure 9: Accuracy of the NN solution method. Comparison between the analytical policy function for inflation and the approximation based on NNs. The values of the fixed parameters in each graph are set to the middle of their bounds.

B.3 Neural network for likelihood approximation

Figure 10 shows the likelihood produced by the NN for varying different parameters (blue line). The orange dots show the mapping between the parameter and the likelihood directly coming from the particle filter.

C Proof of concept 2: Comparison to a state-of-the-art estimation method for nonlinear models

We now assess whether our NN estimation method can recover the true data-generating process of a nonlinear model. Specifically, a nonlinear RANK model with a ZLB. Furthermore, we compare the results with a state-of-the-art estimation method for nonlinear macroeconomic models. This method involves solving the model using global methods and then evaluating the likelihood using a particle filter for every draw of the Metropolis-Hastings algorithm.

C.1 A nonlinear RANK model with the ZLB constraint

We use a small-scale RANK model with the ZLB as a nonlinear element.

Households. The economy consists of a representative household. The household chooses consumption C_t, labor H_t and assets B_t to maximize their utility:

$$E_0 \sum_{t=0}^{\infty} \beta^t \exp(\zeta_t) \left[\left(\frac{C_t}{1-\sigma} \right)^{1-\sigma} - \chi \left(\frac{1}{1+\eta} \right) (H_t)^{1+\eta} \right],$$

where ζ_t is an aggregate preference shock, which follows an AR(1) process $\zeta_t = \rho_\zeta \zeta_{t-1} + \epsilon_t^\zeta$. C_t is aggregate consumption. The budget constraint in real terms is:

$$C_t + B_t = W_t H_t + \frac{R_{t-1}}{\Pi_t} B_{t-1} - T_t + Div_t, \tag{28}$$

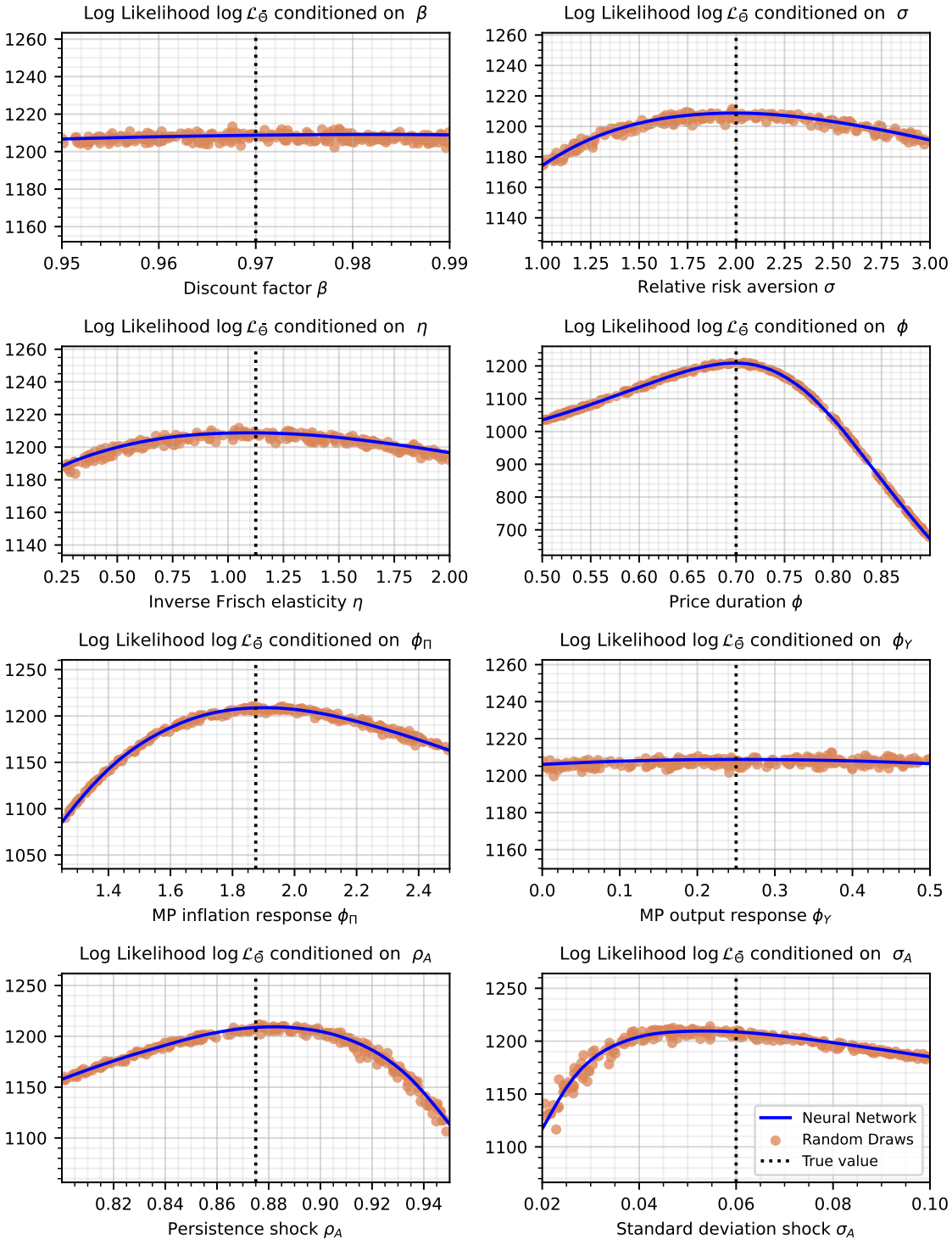

Figure 10: Accuracy in likelihood evaluations: NN particle filter vs. standard particle filter. The log-likelihood of the model as a function of different parameters. The value of the fixed parameters is set to the middle of their bounds.

where Div_t is the real dividend, W_t is real wage, R_t is the nominal interest rate, Π_t is the inflation rate and T_t is real lump sum tax. The first-order conditions are:

$$1 = \beta R_t \mathbb{E}_t \left[\left(\frac{\exp(\zeta_{t+1})}{\exp(\zeta_t)} \right) \left(\frac{C_t}{C_{t+1}} \right)^{\sigma} \frac{1}{\Pi_{t+1}} \right], \tag{29}$$

$$\chi(H_t)^{\eta} = (C_t)^{-\sigma} W_t. \tag{30}$$

Firms. The firm sector consists of a continuum of final goods producers and intermediate goods firms. The final goods retailers buy the intermediate goods and transform them into a homogeneous final good using a CES production technology:

$$Y_t = \left(\int_0^1 (Y_t^j)^{\frac{\epsilon-1}{\epsilon}} dj \right)^{\frac{\epsilon}{\epsilon-1}}, \tag{31}$$

where Y_t^j is the output of intermediate goods firm j. The equilibrium price of the final good and the demand for the intermediate goods of firm j can be expressed as:

$$P_t = \left(\int_0^1 (P_t^j)^{1-\epsilon} dj) \right)^{\frac{1}{1-\epsilon}}, \quad \text{and} \quad Y_t^j = \left(\frac{P_t^j}{P_t} \right)^{-\epsilon} Y_t. \tag{32}$$

Intermediate goods producers are monopolistically competitive. The firm j uses labor N_t^j as input to produce output Y_t^j with the following production technology:

$$Y_t^j = A N_t^j, \tag{33}$$

where A is the total factor productivity. Labor is hired in competitive markets, where the wage is determined as follows:

$$W_t = AMC_t. \tag{34}$$

The firm j sets the price of its goods to maximize its profit subject to the demand curve for intermediate goods and Rotemberg adjustment costs for changing prices:

$$\max_{P_t^j} P_t^j \left(\frac{P_t^j}{P_t} \right)^{-\epsilon} \frac{Y_t}{P_t} - MC_t \left(\frac{P_t^j}{P_t} \right)^{-\epsilon} Y_t - \frac{\varphi}{2} \left(\frac{P_t^j}{\Pi P_{t-1}^j} - 1 \right)^2 Y_t, \tag{35}$$

where Π is the inflation target of the central bank. Imposing a symmetric equilibrium and discounting future profits with the real interest rate, the New Keynesian Phillips curve can be written as:

$$\left[\varphi\left(\frac{\Pi_t}{\Pi}-1\right)\frac{\Pi_t}{\Pi}\right]=(1-\epsilon)+\epsilon MC_t+\varphi\mathbb{E}_t\frac{\Pi_{t+1}}{R_t}\left[\left(\frac{\Pi_{t+1}}{\Pi}-1\right)\frac{\Pi_{t+1}}{\Pi}\frac{Y_{t+1}}{Y_t}\right], \quad (36)$$

where $\Pi_t = P_t/P_{t-1}$. The Rotemberg adjustment costs are given back as a lump sum. The real dividends paid out by the firm sector are $Div_t = Y_t - W_t N_t$.

Policy makers. The central bank sets the nominal interest R_t using a Taylor rule that responds to inflation and output deviations from their targets Π and Y. The ZLB restricts the nominal interest rate. The rule is given as follows:

$$R_t = \max\left[1, R\left(\frac{\Pi_t}{\Pi}\right)^{\theta_\Pi}\left(\frac{Y_t}{Y}\right)^{\theta_Y}\right]. \quad (37)$$

The fiscal authority follows a passive policy rule, where it uses lump-sum taxes T_t to keep their debts D on a constant path:

$$D = \frac{R_{t-1}}{\Pi_t}D - T_t. \quad (38)$$

C.2 Mapping of the model to the general framework

We can map the RANK model in the general form of the outlined estimation procedure. The state variable, the structural shock, and the control variables are:

$$\mathbb{S}_t = \{\zeta_t\}, \quad \text{and} \quad \nu_t = \left\{\epsilon_t^\zeta\right\}, \quad \text{and} \quad \psi_t = \{N_t, \Pi_t\}. \quad (39)$$

The parameters of the model are divided in calibrated ($\bar{\theta}$) and estimated ones ($\tilde{\theta}$):

$$\bar{\Theta} = \{\beta, \sigma, \eta, \epsilon, \chi\}, \quad (40)$$

$$\tilde{\Theta} = \{\theta_\Pi, \theta_Y, \varphi, \rho_\zeta, \sigma_\zeta\}. \quad (41)$$

The NN is trained to determine labor supply and inflation, which is sufficient to back out other variables:

$$\begin{pmatrix} N_t \\ \Pi_t \end{pmatrix} = \psi_{NN}\left(\mathbb{S}_t, \tilde{\Theta}|\bar{\Theta}\right). \tag{42}$$

C.3 Data-generating process

The calibrated model is used as a data-generating process, providing a controlled environment for the estimation. The upper panel of Table 5 summarizes the calibration of the model, with parameters chosen such that the model occasionally encounters the ZLB. We use our developed approach to estimate the nonlinear model in a Bayesian setup. We then compare the results to an estimation with a state-of-the-art (non-NN) approach. The observables are the annualized quarterly output growth rate, annualized quarterly inflation rate, and the nominal interest rate, which are generated with the calibrated model and cover a span of 500 periods.

C.4 Estimation

Priors. The prior distributions are truncated normal densities.[28] The prior mean corresponds to the true value, while the standard deviation σ is loose to avoid the results being driven by the prior. The truncation ensures that the drawn parameters lie inside the bounds that have been imposed while solving the NN. The lower panel of Table 5 summarizes the priors.

Measurement equation. We base the analysis on the annualized quarterly output growth rate, annualized quarterly inflation rate, and nominal interest rate. All data

[28]The probability density function of the truncated normal is $f(x;\mu,\sigma,a,b,) = \frac{1}{\sigma}\frac{\phi((x-\mu)/\sigma)}{\Phi((b-\mu)/\sigma)-\Phi((a-\mu)/\sigma)}$. If the bounds are not symmetric, the parameter μ does not correspond to the mean of the truncated normal. For simplicity, we refer to μ as the mean independent of the bounds.

True parameters for the data-generating process					
Parameters		Value	Parameters		Value
β	Discount factor	0.9975	θ_Π	Mon. pol. inflation response	2
σ	Relative risk aversion	1	θ_Y	Mon. pol. output response	0.25
η	Inverse Frisch elasticity	1	$4\log(\Pi)$	Inflation target (annualized)	2
ϵ	Price elasticity demand	11	Y	Output target	1
χ	Disutility labor	0.91	ρ_ζ	Persistence preference shock	0.7
φ	Rotemberg pricing	1000	$100\sigma^\zeta$	Std. dev. preference shock	2

Estimation											
Par.	Prior					NN			Alternative Approach		
	Type	Mean	Std	Lower Bound	Upper Bound	Posterior Median	Posterior 5%	Posterior 95%	Posterior Median	Posterior 5%	Posterior 95%
θ_Π	Trc.N	2.0	0.1	1.5	2.5	2.04	1.92	2.15	2.06	1.93	2.20
θ_Y	Trc.N	0.25	0.05	0.05	0.5	0.250	0.240	0.260	0.248	0.237	0.260
φ	Trc.N	1000	50	700	1300	985	921	1047	970	909	1033
ρ_ζ	Trc.N	0.7	0.05	0.5	0.9	0.69	0.671	0.707	0.688	0.670	0.707
σ^ζ	Trc.N	0.02	0.0025	0.01	0.025	0.020	0.019	0.021	0.020	0.019	0.021

Table 5: Parameters of the nonlinear RANK model. The upper panel shows the calibrated parameters for the nonlinear RANK model. The lower panel shows the true values used for simulating the model, the prior distributions as well as the posterior distributions. The posterior is displayed for the NN-based estimation and an alternative (non-NN) state-of-the-art approach. Trc.N stands for a truncated normal distribution.

are in percent. The measurement equation is given as:

$$\begin{bmatrix} \text{Output Growth} \\ \text{Inflation} \\ \text{Interest Rate} \end{bmatrix} = \begin{bmatrix} 400\left(\frac{Y_t}{Y_{t-1}} - 1\right) \\ 400\left(\Pi_t - 1\right) \\ 400\left(R_t - 1\right) \end{bmatrix} + u_t, \tag{43}$$

where the measurement error follows a Gaussian distribution $u_t \sim \mathcal{N}(0, \Sigma_u)$. The variance of the measurement error for each time series is a fraction m_E of its own variance. We set $m_E = 0.1$.

Training of the extended policy functions. The computationally most challenging step in the estimation is the first one, where we solve for the policy functions. The NN approximates the labor and consumption policy and is conditioned on the

state variables and parameters to estimate. The network is trained by minimizing the Euler equation and New Keynesian Phillips curve errors.

We use 100,000 iterations to train the NN.[29] The batch size is set to 100, which can be thought of as having 100 different parallel economies in each iteration. We draw new parameters after every 100 iterations for the first 40,000 iterations and after 10 iterations during the rest of the training. After drawing new parameters, we simulate the economy for 100 periods to get a new draw for the state variable for each batch. For iterations where we don't draw new parameters, we simulate for 20 periods. The expectations are evaluated using 100 draws.[30]

The NN convergence during the training is shown in Figure 11, where the mean squared error averaged across the functions we minimize is shown. The shaded area indicates the periods 10,000 to 15,000 in which we introduce the ZLB. We slowly scale the impact of the ZLB using the following functional form:

$$R_t = \max[R_t^N, 1] + a^{ZLB} \min[R_t^N - 1, 0] \tag{44}$$

The parameter a^{ZLB} is gradually lowered from 1 to 0 to move from the economy without ZLB to the ZLB economy. The introduction of the ZLB complicates the problem, as can be seen by the increase in the loss. At the end of the training, the average mean squared error reaches a level of about 10^{-6}.

NN particle filter. We generate 15,000 likelihoods with the particle filter to train the NN-based particle filter. We use the RWMH algorithm to easily obtain 50,000 draws from the posterior distribution after a burn-in.

[29] The NN contains five hidden layers with 128 neurons each. The activation function for the four initial hidden layers is the Sigmoid Linear Unit (SiLU), while the fifth layer is Leaky Rectified Linear Unit (Leaky ReLU). The optimizer is AdamW. The learning rate follows a cosine annealing scheduler starting from an initial rate of $1e-3$ and terminating at a rate of $1e-7$.

[30] Our algorithm follows the general setup that we have described in Section 2.

Figure 11: Convergence of the NN solution. The figure shows the mean squared error averaged across the two functions we minimize. The loss is calculated as a moving average over 1000 iterations. The vertical axis has a logarithmic scale. The shaded area indicates the period in which we introduce the ZLB.

Comparison. We compare the results to a state-of-the-art nonlinear estimation approach that does not use machine learning techniques. The global solution method is based on time iteration with piecewise linear policy functions as in Richter et al. (2014), and the particle filter implementation follows Herbst and Schorfheide (2015). When applying the alternative approach, we similarly use the nonlinear model as the data-generating-process.[31] However, this approach is much slower because we need to resolve the model and run the particle filter for each draw. These limitations justify our decision to use only 50,000 draws in the Metropolis-Hastings algorithm.

Results. The estimation results are summarized in Table 5. First, the NN approach performs very well in recovering the true data-generating process. The posterior median is close to the true value and always contained inside the 90% credible interval. Furthermore, the posterior median and credible interval are very similar to those obtained with the alternative method.

[31] We use the same sequence of structural shocks, measurement error shocks, and the true parameter values to generate the data. However, the solution of the model comes either from our neural network or from the standard global method.

D HANK model

The model is a nonlinear HANK model that includes both idiosyncratic and aggregate risk. It incorporates two key features: households facing idiosyncratic income risk and a borrowing limit; and the ZLB on the nominal interest rate, which constrains monetary policy. Additionally, the model includes demand, supply, and monetary policy shocks.

Households. The economy consists of a continuum of households. The households choose consumption C_t^i, labor H_t^i, and assets B_t^i to maximize their utility:

$$E_0 \sum_{t=0}^{\infty} \beta^t \exp(\zeta_t) \left[\left(\frac{1}{1-\sigma} \right) \left(\frac{C_t^i}{A_t} \right)^{1-\sigma} - \chi \left(\frac{1}{1+\eta} \right) (H_t^i)^{1+\eta} \right],$$

where ζ_t is an aggregate preference shock, which follows an AR(1) process $\zeta_t = \rho_\zeta \zeta_{t-1} + \epsilon_t^\zeta$. C_t is aggregate consumption and A_t is the total factor productivity. The budget constraint in real terms can be written as:

$$C_t^i + B_t^i = \tau_t \left(\frac{W_t}{A_t} \exp(s_t^i) H_t^i \right)^{1-\gamma_\tau} + \frac{R_{t-1}}{\Pi_t} B_{t-1}^i + Div_t^i, \tag{45}$$

where $Div_t^i = Div_t \exp(s_t^i)$ is the household specific real dividends, W_t is real wage, R_t is the gross nominal interest rate, and Π_t is the gross inflation rate. The taxation is progressive, which is governed by γ_τ and τ_t determines the level of the post-tax income. The wage is divided by the productivity level to avoid a fiscal drag in taxation. The agents individual labor productivity s_t^i is stochastic and follows an AR(1) process: $s_t^i = \rho_s s_{t-1}^i + \epsilon_t^{s,i}$. The dividend payout is scaled with the level of the idiosyncratic shock. The agents face a borrowing limit $\underline{B}$, which implies:

$$B_t \geq \underline{B}. \tag{46}$$

The first-order conditions can be written as

$$1 = \beta R_t \mathbb{E}_t \left[\left(\frac{exp(\zeta_{t+1})}{exp(\zeta_t)} \right) \left(\frac{\lambda^i_{t+1}}{\lambda^i_t} \right) \frac{1}{\Pi_{t+1}} \right] + \mu^i_t, \tag{47}$$

$$\lambda^i_t = \left(\frac{C^i_t}{A_t} \right)^{-\sigma} \frac{1}{A_t}, \tag{48}$$

$$\chi (H^i_t)^\eta = (\lambda^i_t) \left[\tau_t (1 - \gamma_\tau) \exp(s^i_t) \frac{W_t}{A_t} \left(\exp(s^i_t) \frac{W_t}{A_t} H^i_t \right)^{-\gamma_\tau} \right], \tag{49}$$

where $\mu^i_t \geq 0$ is the normalized Lagrange multiplier on the individual borrowing limit in equation (46).

Firms. The firm sector consists of a continuum of final goods producers and intermediate goods firms. The final goods retailers buy the intermediate goods and transform them into the homogeneous final good using a CES production technology:

$$Y_t = \left(\int_0^1 (Y^j_t)^{\frac{\epsilon-1}{\epsilon}} dj \right)^{\frac{\epsilon}{\epsilon-1}}, \tag{50}$$

where Y^j_t is the output of intermediate goods firm j. The equilibrium price of the final good and the demand for the intermediate goods of firm j can be expressed as:

$$P_t = \left(\int_0^1 (P^j_t)^{1-\epsilon} dj \right)^{\frac{1}{1-\epsilon}}, Y^j_t = \left(\frac{P^j_t}{P_t} \right)^{-\epsilon} Y_t. \tag{51}$$

Intermediate goods producers are monopolistically competitive. The firm j uses labor N^j_t as input to produce output Y^j_t with the following production technology:

$$Y^j_t = A_t N^j_t. \tag{52}$$

Total factor productivity A_t follows a nonstationary process:

$$A_t = a_t A_{t-1}, \tag{53}$$

where the trend growth rate follows an AR(1) process:

$$a_t = \bar{a} exp(z_t), \tag{54}$$

$$z_t = \rho_z z_{t-1} + \epsilon_t^z. \tag{55}$$

Labor is hired in competitive markets so that the wage is given as follows:

$$W_t = A_t MC_t. \tag{56}$$

The firm j sets the price of its goods to maximize its profit subject to the demand curve for intermediate goods and Rotemberg adjustment costs for changing prices:

$$\max_{P_t^j} P_t^j \left(\frac{P_t^j}{P_t}\right)^{-\epsilon} \frac{Y_t}{P_t} - MC_t \left(\frac{P_t^j}{P_t}\right)^{-\epsilon} Y_t - \frac{\varphi}{2}\left(\frac{P_t^j}{\Pi P_{t-1}^j} - 1\right)^2 Y_t, \tag{57}$$

where Π is the inflation target of the central bank. Imposing a symmetric equilibrium and discounting future profits with the real interest rate, the New Keynesian Phillips curve can be written as:

$$\left[\varphi\left(\frac{\Pi_t}{\Pi} - 1\right)\frac{\Pi_t}{\Pi}\right] = (1-\epsilon) + \epsilon MC_t + \beta\varphi \mathbb{E}_t \left(\frac{\exp(\zeta_{t+1})}{\exp(\zeta_t)}\right)\left(\frac{\lambda_{t+1}}{\lambda_t}\right)\left[\left(\frac{\Pi_{t+1}}{\Pi} - 1\right)\frac{\Pi_{t+1}}{\Pi}\frac{Y_{t+1}}{Y_t}\right], \tag{58}$$

where $\Pi_t = P_t/P_{t-1}$ and $\lambda_t = (C_t/A_t)^{-\sigma} {A_t}^{-1}$. The Rotemberg adjustment costs are ex-post given back. The real dividends of the firm sector can then be written as

$$Div_t = Y_t - W_t N_t. \tag{59}$$

Policy makers. The central bank sets the nominal interest R_t using a Taylor rule that responds to inflation and output deviations from their targets Π and Y. The ZLB restricts the nominal interest rate. The rule is given as follows:

$$R_t^N = \left(R\left(\frac{\Pi_t}{\Pi}\right)^{\theta_\Pi}\left(\frac{Y_t}{A_t Y}\right)^{\theta_Y}\right)\exp(mp_t), \tag{60}$$

$$R_t = \max\left[1, R_t^N\right], \tag{61}$$

where R and Y denote the deterministic steady state (DSS) values of the nominal interest rate and output (in the detrended economy). While households face idiosyncratic shocks in the deterministic steady state, the economy does not face and the agents do not expect aggregate shocks. Additionally, there is also a persistent monetary policy shock mp_t to capture prolonged deviations from the rule:

$$mp_t = \rho_m mp_{t-1} + \epsilon_t^m. \tag{62}$$

The fiscal authority sets τ_t to keep their debts D_t on a constant path adjusted for productivity gains:

$$D_t = \frac{R_{t-1}}{\Pi_t} D_{t-1} - T_t, \tag{63}$$

$$T_t = \int \left(W_t \exp(s_t^i) H_t^i\right) di - \int \tau_t \left(\frac{W_t}{A_t} \exp(s_t^i) H_t^i\right)^{1-\gamma_\tau} di. \tag{64}$$

Market clearing. Market clearing for the labor market, bond market, and goods market requires

$$N_t = \int N_t^j dj = \int \exp(s_t^i) H_t^i di, \tag{65}$$

$$D_t = \int B_t^i di, \tag{66}$$

$$Y_t = \int C_t^i di. \tag{67}$$

D.1 Equilibrium conditions

To have stationarity, we need to define the variables as follows $\tilde{X}_t = \frac{X_t}{A_t}$. The relevant conditions can then be written as:

$$\tilde{C}_t^i + \tilde{B}_t^i = \tilde{\tau}_t \left(\tilde{W}_t \exp(s_t^i) H_t^i\right)^{1-\gamma_\tau} + \frac{R_{t-1}}{\Pi_t a_t} \tilde{B}_{t-1}^i + \tilde{Div}_t^i, \tag{68}$$

$$1 = \beta R_t \mathbb{E}_t \left[\left(\frac{exp(\zeta_{t+1})}{exp(\zeta_t)} \right) \left(\frac{\tilde{C}_t^i}{\tilde{C}_{t+1}^i} \right)^{\sigma} \frac{1}{\Pi_{t+1} a_{t+1}} \right] + \mu_t^i, \tag{69}$$

$$\chi (H_t^i)^{\eta} = \left(\tilde{C}_t^i \right)^{-\sigma} \left[\tilde{\tau}_t (1 - \gamma_\tau) \exp(s_t^i) \tilde{W}_t \left(\exp(s_t^i) \tilde{W}_t H_t^i \right)^{-\gamma_\tau} \right], \tag{70}$$

$$\tilde{Y}_t = N_t, \tag{71}$$

$$\tilde{W}_t = MC_t, \tag{72}$$

$$\tilde{Div}_t = \tilde{Y}_t - \tilde{W}_t \tilde{Y}_t, \tag{73}$$

$$\left[\varphi \left(\frac{\Pi_t}{\Pi} - 1 \right) \frac{\Pi_t}{\Pi} \right] = (1 - \epsilon) + \epsilon MC_t + \beta \varphi \mathbb{E}_t \left[\left(\frac{\exp(\zeta_{t+1})}{\exp(\zeta_t)} \right) \left(\frac{\tilde{C}_{t+1}}{\tilde{C}_t} \right)^{-\sigma} \left(\frac{\Pi_{t+1}}{\Pi} - 1 \right) \frac{\Pi_{t+1}}{\Pi} \frac{\tilde{Y}_{t+1}}{\tilde{Y}_t} \right], \tag{74}$$

$$R_t^N = \left(R \left(\frac{\Pi_t}{\Pi} \right)^{\theta_\Pi} \left(\frac{\tilde{Y}_t}{Y} \right)^{\theta_Y} \right) \exp(\epsilon_t^m), \tag{75}$$

$$R_t = \max \left[1, R_t^N \right], \tag{76}$$

$$\tilde{D} = \frac{R_{t-1}}{\Pi_t a_t} \tilde{D} - \tilde{T}_t, \tag{77}$$

$$\tilde{T}_t = \int \left(\tilde{W}_t \exp(s_t^i) H_t^i \right) di - \int \tilde{\tau}_t \left(\tilde{W}_t \exp(s_t^i) H_t^i \right)^{1-\gamma_\tau} di \tag{78}$$

$$N_t = \int \exp(s_t^i) H_t^i di, \tag{79}$$

$$D = \int \tilde{B}_t^i di, \tag{80}$$

$$\tilde{Y}_t = \int \tilde{C}_t^i di. \tag{81}$$

E NN-based solution algorithm for HANK

The algorithm uses the outlined NN approach to solve the HANK model described in the previous section over the parameter space.

The state variables and shocks of the HANK model are given as:

$$\mathbb{S}_t = \left\{ \left\{ \tilde{B}_{t-1}^i \right\}_{i=1}^{L}, \left\{ s_t^i \right\}_{i=1}^{L}, R_{t-1}^N, \zeta_t, a_t, \epsilon_t^m \right\}, \tag{82}$$

$$\nu_t = \left\{ \left\{ \epsilon_t^{s,i} \right\}_{i=1}^{L}, \epsilon_t^{\zeta}, \epsilon_t^{z}, \epsilon_t^{m} \right\}. \tag{83}$$

The parameters of the model are divided into calibrated ($\bar{\theta}$) and estimated ones ($\tilde{\theta}$). For instance, in the case of our application with US data, this yields:

$$\bar{\Theta} = \{\beta, \eta, \sigma, \bar{a}, \chi, \gamma^{\tau}, \Pi, D, \rho_s, \rho_{\zeta}\}, \tag{84}$$

$$\tilde{\Theta} = \{\sigma_s, \underline{B}, \varphi, \theta_{\Pi}, \theta_Y, \rho_z, \rho_m, \sigma_{\zeta}, \sigma_z, \sigma_m\}. \tag{85}$$

We use two NNs to separate between individual and aggregate policy functions. The individual NN $\psi_{NN}^{I}(\cdot)$ solves for labor supply:

$$\left\{ \left(H_t^i \right) = \psi_{NN}^{I} \left(\mathbb{S}_t^i, \mathbb{S}_t, \tilde{\Theta} | \bar{\Theta} \right) \right\}_{i=1}^{L}, \tag{86}$$

where $\mathbb{S}_t^i = \left\{ \tilde{B}_{t-1}^i, s_t^i \right\}$. The NN for the aggregate policy functions $\psi_{NN}^{A}(\cdot)$ determines the wage and inflation:

$$\begin{pmatrix} \Pi_t \\ \tilde{W}_t \end{pmatrix} = \psi_{NN}^{A} \left(\mathbb{S}_t, \tilde{\Theta} | \bar{\Theta} \right). \tag{87}$$

One challenge is that we must solve for the deterministic steady state values of the nominal rate and output, as the Taylor rule depends on them. To focus on the algorithm for the moment, let's assume that we have already trained a NN $\psi_{NN}^{SS}(\cdot)$ that maps the parameter values to the deterministic steady state values:

$$\begin{pmatrix} R \\ \tilde{Y} \end{pmatrix} = \psi_{NN}^{SS} \left(\tilde{\Theta} | \bar{\Theta} \right). \tag{88}$$

The training of the above NN precedes the training of other NNs approximating the individual and aggregate policy functions. The procedure is described in Section E.1 and adapts the below algorithm to a version of the model without aggregate risk (a Bewley-Huggett-Aiyagari type model). The steps of the algorithm are as follows:

1. Set up the NNs to approximate the policy functions and initialize their weights:

The NN $\psi^I_{NN}\left(\mathbb{S}^i_t, \mathbb{S}_t, \tilde{\Theta}|\bar{\Theta}\right)$ for the individual policy functions

$$\left\{\left(H^i_t\right) = \psi^I_{NN}\left(\mathbb{S}^i_t, \mathbb{S}_t, \tilde{\Theta}|\bar{\Theta}\right)\right\}^L_{i=1}. \tag{89}$$

The NN $\psi^A_{NN}\left(\mathbb{S}_t, \tilde{\Theta}|\bar{\Theta}\right)$ for the aggregate policy functions:

$$\begin{pmatrix}\Pi_t \\ \tilde{W}_t\end{pmatrix} = \psi^A_{NN}\left(\mathbb{S}_t, \tilde{\Theta}|\bar{\Theta}\right). \tag{90}$$

2. Solve for all time t variables for a given state vector of batch b. From the NN, we have a current guess for the policy functions:

$$\{H^i_t\}^L_{i=1}, \Pi_t, \tilde{W}_t, \tag{91}$$

and the deterministic steady state values of R and $\tilde{Y}$ are given by equation (88). The next step is to calculate the following (aggregate) variables:

$$a_t = \bar{a}exp(z_t), \tag{92}$$

$$N_t = \left(\frac{1}{L}\sum_{i=1}^{L} H^i_t \exp(s^i_t)\right), \tag{93}$$

$$\tilde{Y}_t = N_t, \tag{94}$$

$$\tilde{T}_t = \left(\frac{R_{t-1}}{\Pi_t a_t} - 1\right) D, \tag{95}$$

$$\tilde{\tau}_t = \frac{\frac{1}{L}\sum_{i=1}^{L}\left(\tilde{W}_t \exp(s^i_t) H^i_t\right) - \tilde{T}_t}{\frac{1}{L}\sum_{i=1}^{L}\left(\tilde{W}_t \exp(s^i_t) H^i_t\right)^{1-\gamma_\tau}}, \tag{96}$$

$$MC_t = \tilde{W}_t. \tag{97}$$

Then, we can calculate the nominal interest rate, where we impose the ZLB:

$$R^N_t = R\left(\frac{\Pi_t}{\Pi}\right)^{\theta_\Pi}\left(\frac{\tilde{Y}_t}{\tilde{Y}}\right)^{\theta_Y}\exp(\epsilon^m_t), \tag{98}$$

$$R_t = \max\left[1, R_t^N\right]. \tag{99}$$

After that, we calculate each household's individual variables:

$$\left\{\tilde{C}_t^i = \left(\frac{\tilde{\tau}_t(1-\gamma_\tau)\exp(s_t^i)\tilde{W}_t\left(\exp(s_t^i)\tilde{W}_t H_t^i\right)^{-\gamma_\tau}}{\chi(H_t^i)^\eta}\right)^{1/\sigma}\right\}_{i=1}^{L}, \tag{100}$$

$$\left\{\tilde{Div}_t^i = \left(\tilde{Y}_t - \tilde{W}_t N_t\right)\exp(s_t^i)\right\}_{i=1}^{L}, \tag{101}$$

$$\left\{\tilde{\omega}_t^i = \tilde{\tau}_t\left(\tilde{W}_t\exp(s_t^i)H_t^i\right)^{1-\gamma_\tau} + \frac{R_{t-1}}{\Pi_t a_t}\tilde{B}_{t-1}^i + \tilde{Div}_t^i\right\}_{i=1}^{L}, \tag{102}$$

$$\left\{\tilde{B}_t^i = \tilde{\omega}_t^i - \tilde{C}_t^i\right\}_{i=1}^{L}. \tag{103}$$

Aggregate consumption is:

$$\tilde{C}_t = \frac{1}{L}\sum_{i=1}^{L}\tilde{C}_t^i. \tag{104}$$

We use a Monte Carlo approach to evaluate the expectations. We randomly draw M sets of next period shocks to evaluate the expectations. We use the antithetic variate method to increase the precision and reduce the number of necessary draws. The antithetic variates technique creates to a given path $\{\nu_1, \nu_2, \nu_3, \ldots, \nu_{M/2}\}$ also its antithetic path $\{-\nu_1, -\nu_2, -\nu_3, \ldots, -\nu_{M/2}\}$. The drawn shocks are given as:

$$\left\{\left\{\epsilon_{t+1}^{s,i,m}\right\}_{i=1}^{L}, \epsilon_{t+1}^{\zeta,m}, \epsilon_{t+1}^{z,m}, \epsilon_{t+1}^{m,m}\right\}_{m=1}^{M}, \tag{105}$$

where the last superscript m indicates which shock draw the next period value is associated with. Note that, with a slight abuse of notation, we also use the superscript m, in the first position, to denote the monetary policy shock ϵ_t^m.

We then proceed to calculate the next period values of the stochastic state

variables for all M draws, that is:

$$\left\{\left\{s_{t+1}^{i,m} = \rho_s s_t^i + \epsilon_{t+1}^{s,i,m}\right\}_{i=1}^{L}\right\}_{m=1}^{M}, \tag{106}$$

$$\left\{\zeta_{t+1}^{m} = \rho_\zeta \zeta_t + \epsilon_{t+1}^{\zeta,m}\right\}_{m=1}^{M}, \tag{107}$$

$$\left\{z_{t+1}^{m} = \rho_z z_t + \epsilon_{t+1}^{z,m}\right\}_{m=1}^{M}, \tag{108}$$

which then gives us the state variables for all M shock draws.

We can then use the NNs for the individual and aggregate policy functions to calculate the individual and aggregate control variables:

$$\left\{\left\{\left(H_{t+1}^{i,m}\right) = \psi_{NN}^{I}\left(\mathbb{S}_{t+1}^{i,m}, \mathbb{S}_{t+1}^{m}, \tilde{\Theta}|\bar{\Theta}\right)\right\}_{i=1}^{L}\right\}_{m=1}^{M}, \tag{109}$$

$$\left\{\begin{pmatrix}\Pi_{t+1}^{m} \\ \tilde{W}_{t+1}^{m}\end{pmatrix} = \psi_{NN}^{A}\left(\mathbb{S}_{t+1}^{m}, \tilde{\Theta}|\bar{\Theta}\right)\right\}_{m=1}^{M}. \tag{110}$$

The next step is to calculate the following (aggregate) variables for the period $t+1$ for all M draws:

$$\left\{a_{t+1}^{m} = \bar{a} exp(z_{t+1}^{m})\right\}_{m=1}^{M}, \tag{111}$$

$$\left\{N_{t+1}^{m} = \left(\frac{1}{L}\sum_{i=1}^{L} H_{t+1}^{i,m} \exp(s_{t+1}^{i,m})\right)\right\}_{m=1}^{M}, \tag{112}$$

$$\left\{\tilde{Y}_{t+1}^{m} = N_{t+1}^{m}\right\}_{m=1}^{M}, \tag{113}$$

$$\left\{\tilde{T}_{t+1}^{m} = \left(\frac{R_t}{\Pi_{t+1}^{m} a_{t+1}^{m}} - 1\right) D\right\}_{m=1}^{M}, \tag{114}$$

$$\left\{\tilde{\tau}_{t+1}^{m} = \frac{\frac{1}{L}\sum_{i=1}^{L}\left(\tilde{W}_{t+1}^{m} \exp(s_{t+1}^{i,m}) H_{t+1}^{i,m}\right) - \tilde{T}_{t+1}^{m}}{\frac{1}{L}\sum_{i=1}^{L}\left(\tilde{W}_{t+1}^{m} \exp(s_{t+1}^{i,m}) H_{t+1}^{i,m}\right)^{1-\gamma_\tau}}\right\}_{m=1}^{M}. \tag{115}$$

We proceed with calculating for each household their individual variables in

period $t+1$ for all M draws:

$$\left\{\left\{\tilde{C}^{i,m}_{t+1} = \left(\frac{\tilde{\tau}^m_{t+1}(1-\gamma_\tau)\exp(s^{i,m}_{t+1})\tilde{W}^m_{t+1}\left(\exp(s^{i,m}_{t+1})\tilde{W}^m_{t+1}H^{i,m}_{t+1}\right)^{-\gamma_\tau}}{\chi(H^{i,m}_{t+1})^\eta}\right)^{1/\sigma}\right\}^L_{i=1}\right\}^M_{m=1}, \tag{116}$$

$$\left\{\left\{\tilde{Div}^{i,m}_{t+1} = \left(\tilde{Y}^m_{t+1} - \tilde{W}^m_{t+1}N^m_{t+1}\right)\exp(s^{i,m}_{t+1})\right\}^L_{i=1}\right\}^M_{m=1}, \tag{117}$$

$$\left\{\left\{\tilde{\omega}^{i,m}_{t+1} = \tilde{\tau}^m_{t+1}\left(\tilde{W}^m_{t+1}\exp(s^{i,m}_{t+1})H^{i,m}_{t+1}\right)^{1-\gamma_\tau} + \frac{R_t}{\Pi^m_{t+1}a^m_{t+1}}\tilde{B}^i_t + \tilde{Div}^{i,m}_{t+1}\right\}^L_{i=1}\right\}^M_{m=1}, \tag{118}$$

$$\left\{\left\{\tilde{B}^{i,m}_{t+1} = \tilde{\omega}^{i,m}_{t+1} - \tilde{C}^{i,m}_{t+1}\right\}^L_{i=1}\right\}^M_{m=1}. \tag{119}$$

Aggregate consumption is given as:

$$\left\{\tilde{C}^m_{t+1} = \frac{1}{L}\sum_{i=1}^{L}\tilde{C}^{i,m}_{t+1}\right\}^M_{m=1}. \tag{120}$$

We need to ensure that the borrowing constraints for households $B^i_t \geq \underline{B}$ are satisfied (see also equation (46)).[32] It is convenient to define $\bar{\lambda}^i_t$ as follows:

$$\bar{\lambda}^i_t = 1 - \mu^i_t. \tag{121}$$

We calculate the multiplier for all agents L using their Euler equation:

$$\left\{\bar{\lambda}^i_t = \beta R_t \frac{1}{M}\sum_{m=1}^{M}\left[\left(\frac{\exp(\zeta^m_{t+1})}{\exp(\zeta_t)}\right)\left(\frac{\tilde{C}^i_t}{\tilde{C}^{i,m}_{t+1}}\right)^\sigma \frac{1}{\Pi^m_{t+1}a^m_{t+1}}\right]\right\}^L_{i=1}, \tag{122}$$

where we can now evaluate the expectations by taking the mean of the M draws.

Equipped with this object, we use the Fischer-Burmeister function Ψ^{FB}, a smooth way of representing the Kuhn-Tucker conditions, to ensure that these

[32] The Kuhn-Tucker conditions can be written as $\mu^i_t \geq 0,\ \left(\tilde{B}^i_t - \underline{B}\right) \geq 0,\ \mu^i_t \times \left(\tilde{B}^i_t - \underline{B}\right) = 0.$

conditions hold.[33] This can be written as

$$\Psi^{FB}\left(1-\bar{\lambda}_t^i, \tilde{B}_t^i - \underline{B}\right). \tag{123}$$

We can now calculate the error associated with the Fisher-Burmeister function:

$$\left\{L^{1,i} = \left(\Psi^{FB}\left(1-\bar{\lambda}_t^i, \tilde{B}_t^i - \underline{B}\right)\right)^2\right\}_{i=1}^{L}, \tag{124}$$

where $L^{1,i}$ is the squared error of agent i.

We also calculate the squared error for the New Keynesian Phillips curve, the resource constraint in period t and $t+1$, and market clearing for the bond in period t and $t+1$:

$$\begin{aligned} L^2 = \Bigg(&\left[\varphi\left(\frac{\Pi_t}{\Pi}-1\right)\frac{\Pi_t}{\Pi}\right] - (1-\epsilon) - \epsilon MC_t \\ &-\beta\varphi\frac{1}{M}\sum_{m=1}^{M}\left[\left(\frac{\exp(\zeta_{t+1}^m)}{\exp(\zeta_t)}\right)\left(\frac{\tilde{C}_{t+1}^m}{\tilde{C}_t}\right)^{-\sigma}\left(\frac{\Pi_{t+1}^m}{\Pi}-1\right)\frac{\Pi_{t+1}^m}{\Pi}\frac{\tilde{Y}_{t+1}^m}{\tilde{Y}_t}\right]\Bigg)^2, \end{aligned} \tag{125}$$

$$L^3 = \left(D - \frac{1}{L}\sum_{i=1}^{L} B_t^i\right)^2, \tag{126}$$

$$L^4 = \frac{1}{M}\sum_{m=1}^{M}\left(D - \frac{1}{L}\sum_{i=1}^{L} B_{t+1}^{i,m}\right)^2, \tag{127}$$

$$L^5 = \left(\tilde{Y}_t - \tilde{C}_t\right)^2, \tag{128}$$

$$L^6 = \frac{1}{M}\sum_{m=1}^{M}\left(\tilde{Y}_{t+1}^m - \tilde{C}_{t+1}^m\right)^2. \tag{129}$$

3. Repeat step 2 for each element in a batch B, which gives the following loss

[33] The Kuhn-Tucker conditions can be written as: $e \geq 0,\ f \geq 0,\ e \times f = 0$. The Fischer-Burmeister function representing them is defined as: $\Psi^{FB}(e,f) = e + f - \sqrt{e^2+f^2}$. The conditions are satisfied when $\Psi^{FB}(e,f) = 0$.

components:

$$\left\{\left\{L^{1,i,b}\right\}_{i=1}^{L}, L^{2,b}, L^{3,b}, L^{4,b}, L^{5,b}, L^{6,b}\right\}_{b=1}^{B}. \tag{130}$$

4. Define the loss function:

$$\phi^L = \frac{1}{B}\sum_{b=1}^{B}\left[\sum_{i=1}^{L}\alpha_1^i L_1^{1,i,b} + \alpha_2 L^{2,b} + \alpha_3 L^{3,b} + \alpha_4 L^{4,b} + \alpha_5 L^{5,b} + \alpha_6 L^{6,b}\right], \tag{131}$$

where $\{\alpha_1^i\}_{i=1}^{L}, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6$ determine the weights for the different equations.

5. Optimize the parameters of the NNs ψ_{NN}^I and ψ_{NN}^A to minimize the loss function ϕ^L with a stochastic gradient decent based optimizer.

6. Steps 2 - 5 are repeated N^{int} times to take several optimization steps. Redraw the shock realization in period $t+1$ for each internal optimization step.

7. Draw new parameters for each economy in the batch. This step is not required at each iteration.

8. Simulate each economy for T^{sim} periods using randomly drawn shocks. This creates then the state vector for the next iteration of the optimizer.

9. Steps 2 - 8 are repeated N^{iter} times until the NN converges on average to sufficiently low loss.

E.1 Deterministic steady state and neural network

We need the mapping from the parameters to the deterministic steady state values of the interest rate R and detrended output $\tilde{Y}$, as these objects enter the Taylor rule. To obtain this mapping, we solve for the deterministic steady state (DSS) of our HANK

model, in which agents face and experience idiosyncratic risk while aggregate risk is absent.

The DSS NN solves for the deterministic steady state values:

$$\begin{pmatrix} R \\ \tilde{Y} \end{pmatrix} = \psi_{NN}^{SS}\left(\tilde{\Theta}|\bar{\Theta}\right). \tag{132}$$

To solve for this NN, we also need to solve for the individual policy functions of the agents in the DSS.

$$\left\{\left(H_t^i\right) = \psi_{NN}^{I,DSS}\left(\mathbb{S}_t^i, \mathbb{S}, \tilde{\Theta}|\bar{\Theta}\right)\right\}_{i=1}^{L}, \tag{133}$$

where DSS enters the superscript of the NN for a clear distinction. Note that we let the deterministic steady state values of the aggregate state variables still enter this network. While this is unnecessary for solving for the DSS, it allows us to use this trained NN as a starting point when we solve for the economy with aggregate risk.

1. Set up the NNs to approximate the policy functions and initialize their weights:

   The NN $\psi_{NN}^{SS}\left(\tilde{\Theta}|\bar{\Theta}\right)$ for the steady state parameters:

   $$\begin{pmatrix} R \\ \tilde{Y} \end{pmatrix} = \psi_{NN}^{SS}\left(\tilde{\Theta}|\bar{\Theta}\right). \tag{134}$$

   The NN $\psi_{NN}^{I,DSS}\left(\mathbb{S}_t^i, \mathbb{S}, \tilde{\Theta}|\bar{\Theta}\right)$ for the individual policy functions without aggregate risk:

   $$\left\{\left(H_t^i\right) = \psi_{NN}^{I,DSS}\left(\mathbb{S}_t^i, \mathbb{S}, \tilde{\Theta}|\bar{\Theta}\right)\right\}_{i=1}^{L}. \tag{135}$$

2. Solve for all the current period individual variables. From the NN, we get a

current guess for the deterministic steady state values

$$\begin{pmatrix} R \\ \tilde{Y} \end{pmatrix} = \psi_{NN}^{SS}\left(\tilde{\Theta}|\bar{\Theta}\right), \tag{136}$$

and individual policy functions

$$\{H_t^i\}_{i=1}^L. \tag{137}$$

Note that inflation is at target in the DSS, the wage equals its value in the DSS without idiosyncratic risk, and the growth rate is at its mean. Thus, we know:

$$\Pi, \tilde{W}, g. \tag{138}$$

The next step is to calculate the following variables:

$$N_t = \left(\frac{1}{L}\sum_{i=1}^{L} H_t^i \exp(s_t^i)\right), \tag{139}$$

$$\tilde{Y}_t = N_t, \tag{140}$$

$$\tilde{T} = \left(\frac{R}{\Pi g} - 1\right) D, \tag{141}$$

$$\tilde{\tau}_t = \frac{\frac{1}{L}\sum_{i=1}^{L}\left(\tilde{W}\exp(s_t^i)H_t^i\right) - \tilde{T}}{\frac{1}{L}\sum_{i=1}^{L}\left(\tilde{W}\exp(s_t^i)H_t^i\right)^{1-\gamma_\tau}}, \tag{142}$$

$$MC = \tilde{W}. \tag{143}$$

Note that we use a subscript t here to indicate variables calculated based on individual shock realizations. We assume that L is sufficiently large so that specific shock realizations and asset level of households do not affect the steady state value.[34] Once the network successfully approximates the DSS, aggregate variables such as (detrended) output are the same in each period, e.g. $\tilde{Y}_t \approx \tilde{Y}, \forall t$

[34]Furthermore, the stochastic setup of the NN allows us to average over several hundred thousands of draws as we consider in each iteration different draws for the batch B.

We pursue calculating each household's individual variables:

$$\left\{ \tilde{C}_t^i = \left(\frac{\tilde{\tau}_t(1-\gamma_\tau)\exp(s_t^i)\tilde{W}\left(\exp(s_t^i)\tilde{W}H_t^i\right)^{-\gamma_\tau}}{\chi(H_t^i)^\eta} \right)^{1/\sigma} \right\}_{i=1}^{L}, \tag{144}$$

$$\left\{ \tilde{Div}_t^i = \left(\tilde{Y}_t - \tilde{W}N_t\right)\exp(s_t^i) \right\}_{i=1}^{L}, \tag{145}$$

$$\left\{ \tilde{\omega}_t^i = \tilde{\tau}_t\left(\tilde{W}\exp(s_t^i)H_t^i\right)^{1-\gamma_\tau} + \frac{R}{\Pi g}\tilde{B}_{t-1}^i + \tilde{Div}_t^i \right\}_{i=1}^{L}, \tag{146}$$

$$\left\{ \tilde{B}_t^i = \tilde{\omega}_t^i - \tilde{C}_t^i \right\}_{i=1}^{L}. \tag{147}$$

Aggregate consumption is given as:

$$\tilde{C}_t = \frac{1}{L}\sum_{i=1}^{L}\tilde{C}_t^i. \tag{148}$$

We use a Monte Carlo approach to evaluate the expectations, which requires randomly drawing M sets of next period shocks for the households (we also use the antithetic variate method here). The drawn shocks are given as:

$$\left\{ \left\{ \epsilon_{t+1}^{s,i,m} \right\}_{i=1}^{L} \right\}_{m=1}^{M}, \tag{149}$$

where the superscript m indicates which shock draw the next period value is associated with.

We then proceed to calculate the next period values of the stochastic state variables for all M draws, that is:

$$\left\{ \left\{ s_{t+1}^{i,m} = \rho_s s_t^i + \epsilon_{t+1}^{s,i,m} \right\}_{i=1}^{L} \right\}_{m=1}^{M}. \tag{150}$$

This gives us the household state variables for all M shock draws.

We can then use the NN for the individual policy functions in the DSS:

$$\left\{\left\{\left(H_{t+1}^{i,m}\right) = \psi_{NN}^{I,DSS}\left(\mathbb{S}_{t+1}^{i,m}, \mathbb{S}^m, \tilde{\Theta}|\bar{\Theta}\right)\right\}_{i=1}^{L}\right\}_{m=1}^{M}. \tag{151}$$

The next step is to calculate the following variables for the period $t+1$ for all M draws:

$$\left\{N_{t+1}^m = \left(\frac{1}{L}\sum_{i=1}^{L} H_{t+1}^{i,m}\exp(s_{t+1}^{i,m})\right)\right\}_{m=1}^{M}, \tag{152}$$

$$\left\{\tilde{Y}_{t+1}^m = N_{t+1}^m\right\}_{m=1}^{M}, \tag{153}$$

$$\left\{\tilde{\tau}_{t+1}^m = \frac{\frac{1}{L}\sum_{i=1}^{L}\left(\tilde{W}\exp(s_{t+1}^{i,m})H_{t+1}^{i,m}\right) - \tilde{T}}{\frac{1}{L}\sum_{i=1}^{L}\left(\tilde{W}\exp(s_{t+1}^{i,m})H_{t+1}^{i,m}\right)^{1-\gamma_\tau}}\right\}_{m=1}^{M}. \tag{154}$$

We proceed to calculate each household's individual variables for the period $t+1$ across all M draws:

$$\left\{\left\{\tilde{C}_{t+1}^{i,m} = \left(\frac{\tilde{\tau}_{t+1}^m(1-\gamma_\tau)\exp(s_{t+1}^{i,m})\tilde{W}\left(\exp(s_{t+1}^{i,m})\tilde{W}H_{t+1}^{i,m}\right)^{-\gamma_\tau}}{\chi(H_{t+1}^{i,m})^\eta}\right)^{1/\sigma}\right\}_{i=1}^{L}\right\}_{m=1}^{M}, \tag{155}$$

$$\left\{\left\{\tilde{Div}_{t+1}^{i,m} = \left(\tilde{Y}_{t+1}^m - \tilde{W}N_{t+1}^m\right)\exp(s_{t+1}^{i,m})\right\}_{i=1}^{L}\right\}_{m=1}^{M}, \tag{156}$$

$$\left\{\left\{\tilde{\omega}_{t+1}^{i,m} = \tilde{\tau}_{t+1}^m\left(\tilde{W}\exp(s_{t+1}^{i,m})H_{t+1}^{i,m}\right)^{1-\gamma_\tau} + \frac{R}{\Pi g}\tilde{B}_t^i + \tilde{Div}_{t+1}^{i,m}\right\}_{i=1}^{L}\right\}_{m=1}^{M}, \tag{157}$$

$$\left\{\left\{\tilde{B}_{t+1}^{i,m} = \tilde{\omega}_{t+1}^{i,m} - \tilde{C}_{t+1}^{i,m}\right\}_{i=1}^{L}\right\}_{m=1}^{M}. \tag{158}$$

Aggregate consumption is given as:

$$\left\{\tilde{C}_{t+1}^m = \frac{1}{L}\sum_{i=1}^{L}\tilde{C}_{t+1}^{i,m}\right\}_{m=1}^{M}. \tag{159}$$

We need to ensure that the borrowing constraint of the households $B_t^i \geq \underline{B}$ is satisfied. It is convenient to define $\bar{\lambda}_t^i$ as follows:

$$\bar{\lambda}_t^i = 1 - \mu_t^i. \tag{160}$$

We calculate the multiplier for all agents L using their Euler equation:

$$\left\{ \bar{\lambda}_t^i = \beta R \frac{1}{M} \sum_{m=1}^{M} \left[\left(\frac{\tilde{C}_t^i}{\tilde{C}_{t+1}^{i,m}} \right)^{\sigma} \frac{1}{\Pi g} \right] \right\}_{i=1}^{L}, \tag{161}$$

where we can now evaluate the expectations by taking the mean over the M draws.

Equipped with this object, we use the Fischer-Burmeister function Ψ^{FB} to enforce the Kuhn-Tucker conditions again. This can be written as:

$$\Psi^{FB}\left(1 - \bar{\lambda}_t^i, \tilde{B}_t^i - \underline{B}\right). \tag{162}$$

We can now calculate the error associated with the Fischer-Burmeister function:

$$\left\{ L^{1,DSS,i} = \left(\Psi^{FB}\left(1 - \bar{\lambda}_t^i, \tilde{B}_t^i - \underline{B}\right) \right)^2 \right\}_{i=1}^{L}, \tag{163}$$

where $L^{1,DSS,i}$ is the squared Euler error of agent i.

We also calculate the squared error for aggregate output, the resource constraint in period t and $t+1$, and market clearing for the bond in period t and $t+1$:

$$\begin{aligned}
L^{2,DSS} &= \left(\tilde{Y}_t - \tilde{Y} \right)^2, \\
L^{3,DSS} &= \left(D - \frac{1}{L} \sum_{i=1}^{L} B_t^i \right)^2, \qquad (164) \\
L^{4,DSS} &= \frac{1}{M} \sum_{m=1}^{M} \left(D - \frac{1}{L} \sum_{i=1}^{L} B_{t+1}^{i,m} \right)^2, \qquad (165) \\
L^{5,DSS} &= \left(\tilde{Y}_t - \tilde{C}_t \right)^2, \qquad (166)
\end{aligned}$$

$$L^{6,DSS} = \frac{1}{M} \sum_{m=1}^{M} \left(\tilde{Y}_{t+1}^{m} - \tilde{C}_{t+1}^{m} \right)^2 . \tag{167}$$

3. Repeat step 2 for each element in a batch B, which gives the following loss components for the entire batch:

$$\left\{ \left\{ L^{1,DSS,i,b} \right\}_{i=1}^{L}, L^{2,DSS,b}, L^{3,DSS,b}, L^{4,DSS,b}, L^{5,DSS,b}, L^{6,DSS,b} \right\}_{b=1}^{B} . \tag{168}$$

4. Define the loss function:

$$\phi^{L} = \frac{1}{B} \sum_{b=1}^{B} \left[\sum_{i=1}^{L} \alpha_1^{i,DSS} L_1^{1,DSS,i,b} + \sum_{j=2}^{6} \alpha_j^{DSS} L^{j,DSS,b} \right], \tag{169}$$

   where the weights for the equations are determined by $\left\{ \alpha_1^{i,DSS} \right\}_{i=1}^{L}$ and $\alpha_j^{DSS}, \forall j = 2, 3, 4, 5, 6$.

5. Optimize the parameters of the NNs ψ_{NN}^{SS} and $\psi_{NN}^{I,DSS}$ to minimize the loss function ϕ^{L} with a stochastic gradient decent based optimizer.

6. Steps 2 - 5 are repeated N^{int} times to take several optimization steps. Redraw the shock realization in period $t+1$ for each internal optimization step.

7. Draw new parameters for each economy in the batch. This step is not required at each iteration.

8. Simulate each economy for T^{sim} periods using randomly drawn shocks. This creates the state vector for the next iteration of the optimizer.

9. Steps 2 - 8 are repeated N^{iter} times until the NN converges on average to sufficiently low loss.

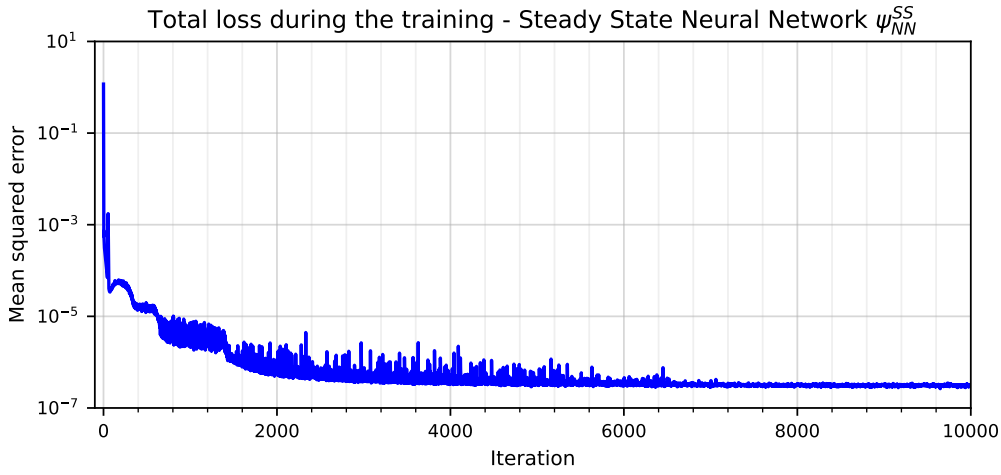

Figure 12: Convergence of the NN solution for the DSS of the HANK model. The figure shows the dynamics of the mean squared error, which is averaged across the equations that we minimize, associated with the deterministic steady state values of the nominal rate R and output Y. The vertical axis has a logarithmic scale.

E.2 Training of the neural networks and mean squared error

We are training our NN based model solution over the extended parameter space in two steps. The details of the training process and the mean squared error are shown.

Training of the DSS neural network. The NN to approximate output and nominal rate in the DSS is trained for 10,000 iterations. We conduct 15 optimization steps during each iteration step, including redrawing of the shocks, and use 100 Monte Carlo draws. After each iteration, the economy is simulated for 20 periods. The parameters are redrawn after every 40 iterations and are initialized with a longer simulation of 200 periods. Note that this step also requires training a temporary individual policy function, which provides the decision of households in the DSS. The convergence of the NN is shown in Figure 12, where the mean squared error averaged across the equations that we minimize is shown. At the end of the training, the NN approximates the DSS with a very high accuracy.

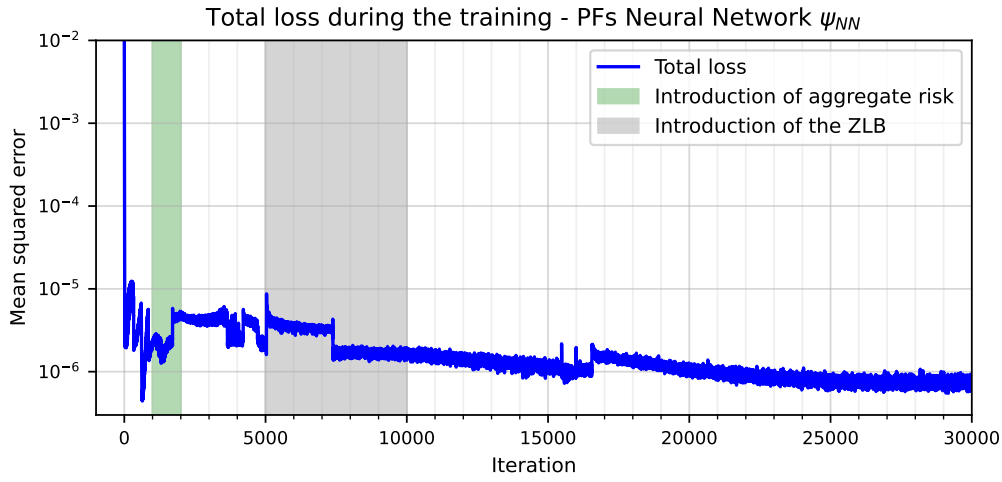

Figure 13: Convergence of the NN solution for the HANK model. The figure shows the dynamics of the mean squared error during the training of the extended individual and aggregate policy functions. The shaded areas indicate the periods in which we introduce aggregate risk and the ZLB. The vertical axis has a logarithmic scale.

Training of extended policy functions. We use 30,000 iterations to jointly train the NNs to approximate the extended individual and aggregate policy functions. We conduct 15 optimization steps during each iteration step, including redrawing of the shocks, and use 100 Monte Carlo draws. After each iteration, the economy is simulated for 20 periods. The parameters are redrawn after every 40 iterations and are initialized with a longer simulation of 200 period.[35] Figure 13 shows that the loss converges to an error in the magnitude of 10e-6. For better convergence, we slowly introduce aggregate risk between periods 1000 and 2000 by scaling the standard deviations of the aggregate shocks from a tiny value to the actual values. Similarly, we introduce the ZLB between periods 5000 and 1000 by slowly scaling the impact of the ZLB. In particular, we use the following functional form:

$$R_t = \max[R_t^N, 1] + a^{ZLB} \min[R_t^N - 1, 0]. \tag{170}$$

[35]During the first 1000 iterations, we have a shorter simulation horizon of 10 periods and 100 periods if parameters are redrawn.

The parameter a^{ZLB} is stepwise lowered from 1 to 0 to move from the economy without ZLB to the ZLB economy.