NATIONAL RESEARCH UNIVERSITY
HIGHER SCHOOL OF ECONOMICS
Faculty of Economic Sciences

Tatiana Sorokina

GRADUATE QUALIFICATION WORK - BACHELOR'S THESIS

Solving and Estimating Non-Linear HANK Models With Machine Learning

Qualification Economics 38.03.01
Educational program «Economics and Statistics»

Supervisor
Candidate of Sciences (PhD)
Nikolay Pilnik

Moscow 2023

Contents

Abstract

In recent years, Dynamic Stochastic General Equilibrium (DSGE) models have become a go-to tool for counterfactual and policy analysis in central banks. In practice, overly-complex models are simplified to a more tractable version that can be solved with fast local methods, as the closed-form or even approximate solutions become infeasible due to the "dimensionality curse." This simplification comes at the sacrifice of interesting features and loss of the empirical credibility of the model output. To address this problem, we develop a neural network algorithm for approximating and estimating the global solution in high-dimensional dynamic economic models. In our framework, the neural networks are efficiently trained using stochastic gradient descent to approximate policy and decision functions; moreover, the framework determines a foundation for a subsequent fast parameter estimation process as the neural network solves economic models for the whole parameter space.

We apply our algorithm to a simple consumption-saving problem with multiple shocks, a small-scale representative-agent New Keynesian DSGE model with monopolistic competition, and two HANK models with strong non-linearities and aggregate risks. The application of our framework has demonstrated that it can generate reliable and interpretable economic results for small-scale representative-agent models. However, it does not consistently produce accurate solution approximations for complex heterogeneous-agent models. To address the problem, our suggested solution involves utilising different sampling and simulation methods to increase the stability of the neural network solution.

1 Introduction

Modern central banks use complex DSGE models in decision-making analysis. Unlike empirical macroeconometric approaches, DSGE models have a micro-founded framework and are constructed using a systematic approach to macroeconomic modeling, rather than drawing on rigid assumptions. They explicitly describe technology, agents' preferences and information sets, and take an optimisation-based approach to structural equation development. These models rely on overall contemporary macroeconomic theory to explain and predict aggregate time series movements over the business cycle, which makes them relevant to the use of policymakers, researchers, and major market participants.

Most dynamic economic models do not admit a closed-form solution. Moreover, the presence of complex features makes it computationally infeasible to resolve the models due to: stochasticity, an exponential increase in the number of state spaces, strong non-linearities or kinks in the equilibrium functions, and irregular geometries of the ergodic set of states. As the number of state variables increases, classical solution methods such as perturbation, projection, and stochastic simulation can become infeasible or inadequate. Despite theoretical and technological advances to improve approximation accuracy and lower the computation time, in practice, most models are reduced to tractable approximations to deal with the "dimensionality curse" (Bellman 1958). This comes at the cost of losing interesting features in the model, such as: the impact of the zero lower bound, which is erased through linearisation - the transformation needed to solve the model via perturbation around the steady state; or financial frictions with occasionally binding constraints that induce kinks in the agent decision functions which cannot be approximated via standard functions in the projection method. Even the strategies designed to approximate the decision functions (e.g. Smolyak sparse grids) fail if the dimensionality of the state space exceeds ~ 20.

Our paper aims to build on existing solution algorithms of high-dimensional dynamic economic models with strong non-linearities and propose a tool for approximating the global solution with neural networks. The algirithm learns the agents' optimal decision rules in the evolving economy through the Euler optimisation problem. Moreover, it is adaptable for faster model parameter estimation as it solves the model for the entire parameter space and does not require repetitive solution during the estimation step.

For our algorithm, we choose the class of heterogeneous-agent (HA) economic models, as they represent a relevant framework for policy-analysis DSGE model formulation. In particular, they show augmented inflation costs in the models due to the distributional effects of monetary policy and resolution of the "risk-free rate puzzle" as a result of the uninsurable idiosyncratic uncertainty. Moreover, in the HA models, the agents' asymmetric response to redistributive effects of inflation strengthens the propagation of shocks, affecting aggregate macroeconomic variables as well as enhancing the cost of the business cycle fluctuations. Following these model implications, adopting the heterogeneous agent instead of representative-agent (RA) framework at central banks is considered crucial in literature (see, *e.g.* Algan et al. 2010; Dou, Lo, et al. 2020; Alves et al. 2022), which is why fast solution algorithms for the HA models are needed.

We find that our neural network framework formulation generates reliable and interpretable economic results for small-scale representative-agent models. However, it does not consistently produce accurate solution approximations for complex heterogeneous-agent models. To address the problem, we suggest enhancing the neural network training loss function. Firstly, by updating the sampling technique to include more precise definitions of ergodic spaces for generating parameter and state variables. And secondly, by accounting for more future periods with the Bellman value function or alternative integral approximations to improve the simulated approximation of the agent expectations. The mentioned improvements, along with the practical application of the neural network framework for the model estimation on real data, provide a clear path for further research.

The rest of the paper is organised as follows. In Section 2, we discuss modern methods for solving and estimating complex economic models that include agent heterogeneity. We outline effective techniques that we use in our algorithm later on. Additionally, the literature review summarises previous research on how Machine Learning (ML) techniques have been applied to solve and estimate economic models. Section 3 introduces the neural network framework that is used for the DSGE model solution and subsequent estimation. Section 4 describes two simple models - consumption-saving and small-scale representative-agent New Keynesian DSGE model with monopolistic competition - and casts them into our neural network framework to showcase its practical application. In Section 4, we also describe difficulties we have encountered while applying our framework to complex Heterogeneous-Agent New Keynesian (HANK) models in the style of Krusell and Smith (1998), propose our treatments to the encountered inconsistencies, and lay out the path for further research. At last, Section 5 concludes.

2 Literature Review

This section outlines the numerical methods commonly used to solve and estimate high-dimensional dynamic economic models. These findings will be used in our neural network framework formulation in Section 3. Particular attention in this review is drawn to the literature regarding the use of the Machine Learning methods in application to the high-dimensional models, as the vast surveys of the standard method and their adaptations for the high-dimensional problems already exist.[1]

The literature emphasises the need for effective and computationally cheap global methods regarding the model solution methods. Analysing the existing research, we conclude that combining simulation and projection approaches proves to be an effective way of overcoming the high-dimensional input problem and ensuring convergence to the global solution. We account for these findings by incorporating both techniques in our neural network framework in our algorithm.

Considering the existing model parameter estimation methods, we conclude that the techniques applied to the heterogeneous-agent models are similar to the representative-agent framework. The only difficulty is attaining a low-cost solution in the HA model setting. Thus, in Section 3, we focus on building a solution framework that would be able to incorporate most model parameter estimation techniques in an efficient manner.

2.1 Solution methods for the heterogeneous-agent models

Dynamic economic models, including the New Keynesian DSGE models considered in this paper, rarely admit closed-form solutions. Conventional numerical methods for deriving

[1] For reviews of numerical methods for the models with finite number of agents see, *e.g.*, L. Maliar and S. Maliar (2014) and Kollmann et al. (2011); for general reviews on the local and global solution methods see, *e.g.*, Fernández-Villaverde, Rubio-Ramírez, et al. (2016) and Brumm, Krause, et al. (2021); for the description of the dynamic model estimation methods see, *e.g.*, Fernández-Villaverde, Rubio-Ramírez, et al. (2016).

the solution include perturbation,[2] projection,[3] and stochastic simulation[4]methods. The dimensionality curse, however, renders these methods either infeasible (due to the need to iterate over an *n*-dimensional hypercube) or inaccurate (due to the stochasticity and irregularly-shaped ergodic states).

In practice, the dynamic economic models are usually solved in their steady-state form, which results in the loss of some specific features (such as binding constraints, which disappear in steady state when the decision or policy function is located either on the bound or non-bounded part, and the numerical approximation becomes a constant or a linear function, breaching the kink), resulting in a local solution. Thus, the need for efficient global solution methods is stressed in literature, by Dou, Fang, et al. (2022) and Fernández-Villaverde, Rubio-Ramírez, et al. (2016) among others. However, projection class global methods are limited to low-dimensional problems and can be rendered inefficient as they must solve the economy of the hyper-cubic domain by using grid methods (which might be wasteful since economies with irregularly shaped ergodic sets of states typically fill only a tiny fraction of the hypercube or cannot be approximated by a hypercube at all).[5] The grid-free, simulation-based global methods are more effective in this regard (see, *e.g.*, K. Judd et al. 2009, L. Maliar and S. Maliar 2014; Scheidegger and Bilionis 2017). Nevertheless, since Monte Carlo sampling is used as a basis for the algorithm, the practical convergence of these methods remains in question.

[2] Perturbation methods can solve models in their steady-state through Taylor expansions. Although, the accuracy of the perturbation methods is uncertain when there are strong non-linearities and kinks in the decision functions (see, e.g., a perturbation method by K. L. Judd and Guu (1993)).

[3] Projection methods base mostly on the tensor-product rules. Therefore, the computational cost of this class of methods increases exponentially with the number of state variables (as the reverse tensor-product becomes increasingly complex to compute) even though they remain fast and accurate in models of low-dimensionality (see, e.g., a projection method by K. Judd (1992)).

[4] Stochastic simulation methods combine iterative least-squares learning and Monte Carlo integration; stochastic sampling in this framework makes them applicable to high-dimensional problems. However, numerical instability of the least-squares learning and low accuracy of the Monte Carlo integration lowers the accuracy of the stochastic simulation methods (see, *e.g.* a stochastic simulation approach by Marcet and Lorenzoni (1998)).

[5] Using Smolyak sparse grids (see, *e.g.*, Krueger and Kubler (2004)) or adaptive sparse grids (see, *e.g.*, Brumm and Scheidegger (2017)) can alleviate the dimensionality curse. However, these methods still have to approximate the economy on a hypercube.

2.1.1 Treating high dimensionality in the heterogeneous-agent models

Aggregate uncertainty usually requires treating agents' characteristics as separate state variables in order to account for changes in their cross-sectional distribution. However, in the presence of the common (for HA models) assumption of a continuum of agents, the set of state variables for the individual functions becomes infinite, and even an approximation of an economy with a sufficiently large number of agents extensively augments the state space.

Several approaches have been developed to handle heterogeneity, including seminal Krusell and Smith (1998) and Den Haan (1996), which use simulation and projection methods: in both works, an infinite-dimensional object is summarised with a limited number of distribution moments (typically of the first or second order). Krusell and Smith (1998) specify an approximation to the aggregate law of motion for the moment function to describe the next period's prices in terms of the next period's value of the aggregate shock and the current period's moments. They then solve for the individual decision function in the usual way, treating the output of the approximated aggregate law of motion as a state variable. Rather than assuming the functional form of the aggregate law of motion, Den Haan (1996) directly approximates the individual decision function from simulations. In economies with borrowing constraints and missing insurance markets, these methods' "approximate aggregation" is an effective way of predicting period prices since the marginal propensity to save is very similar among agents of different incomes and wealth levels, except for the very poor. Due to the small number of very poor agents in such economies, wealth redistribution has little impact on aggregate savings, thereby having no effect on market prices. These methods, however, suffer from the dimensionality curse in relatively complex economies and, therefore, are confined to a limited number of assets and shock variables. Moreover, adopted simplifications cause information loss in the model, which results in the inability to identify model parameters for large models.

Local perturbation approaches used for the HA models are usually applied to the economies with no aggregate uncertainty, as they propose to perturb around the steady-state model solution. Recent papers that discuss how to deal with variable distributions across households include the following approaches: Winberry (2018), which uses low-dimensional differentiable functions to parametrise the infinite-dimensional vector distribution of heterogeneous firms; Bayer and Luetticke (2018), which solves linearised HA models under the assumption of a constant copula for cross-sectional household variable distribution; and Ragot (2019), which aggregates households by shock distribution instead of capital distribution to solve for optimal Ramsey policies in a HA model with aggregate shocks, and then applies simulation to perturb the model's steady state. It should be noted that, as with the representative-agent models, the perturbation method cannot be used for problems that involve nonlinear dynamics induced by idiosyncratic shocks or problems that do not approach the stationary equilibrium in a deterministic manner. Thereby, in the following sections, we focus on the computational economics literature on the global numerical methods instead of the local ones.

2.1.2 Machine Learning techniques in application to economic model solution

Numerical methods from Machine Learning are used as base algorithms to find optimal decision functions - even of a non-linear type - and for augmentation of the efficiency of the general methods in application to the HA framework. Scheidegger and Bilionis (2017) and Renner and Scheidegger (2018) propose a global solution simulation-based method which uses Gaussian Processes for approximating the decision functions; they also employ a Bayesian active learning algorithm for selecting the most information-dense data points for the function training. With this method, the hypercube where the economy lies can be determined even when the ergodic set of states has irregular geometries, which cannot be achieved with the grid-based projection method. Using Gaussian Processes also has the advantage of addressing the parameter uncertainty and providing decision function confidence intervals that consider the data's limitations. Gaussian Processes, however, have the drawback of not being capable of handling a large number of variables; this means that the benefits of simulation-based methods, such as no-cost input data, are not available in such algorithms. Consequently, the literature explores other techniques for function approximation that might be more suitable for high-dimensional problems.

2.1.3 Neural networks in application to economic model solution

One of the first implementations of neural networks for solving dynamic models was done by Duffy and McNelis (2001), who extended the Haan and Marcet (1990) simulation-based stochastic parameterised expectations algorithm to shallow neural networks. This approach was later implemented by Duarte (2018), who approximated the solution to the Bellman equation in optimal control problems in macro-finance; Fernández-Villaverde, Hurtado, et al. (2019) parametrised the aggregate low of motion from the cross-sectional distribution of assets in a modified continuous-time Krusell and Smith (1998) model; Valaitis and Villa (2021) used a framework similar to Duarte (2018) to solve the Faraglia et al. (2019) discrete-time optimal debt management problem with three maturities. Similarly to them, Hill et al. (2021) use a Bellman equation to approximate an individual decision function in a General Equilibrium Epi-Macro model with individual state stochasticity and rational expectations while taking into account past realisations of the general equilibrium over time. These papers demonstrate, in particular, that using neural networks rather than polynomials to approximate expectation functions can eliminate the problems associated with multicollinearity and high dimensionality of state spaces.

Furthermore, the Krusell and Smith (1998) algorithm has been reformulated by Han et al. (2021) to incorporate neural networks as base functions for the calculation of the generalised moments of the capital distribution. Following this, the moments are included as inputs into the individual decision function, which is approximated by using neural networks as well. The method shows similar accuracy to the one introduced by Fernández-Villaverde, Hurtado, et al. (2019); therefore, it can be concluded that one neural network can be sufficient to capture dynamics of both the aggregate low of motion and individual decision functions.

Another approach was developed independently by L. Maliar, S. Maliar, and Winant (2021) and Azinovic et al. (2019). In both papers, the entire model (in the form of residuals in the Euler equation for Azinovic et al. (2019) and, in addition to that, in the form of the lifetime reward and residuals in the Bellman equations for L. Maliar, S. Maliar, and Winant (2021)) is adapted into a neural network representation and solved using stochastic gradient descent (SGD) on simulated points. Azinovic et al. (2019) solve complex finite versions of the Krueger and Kubler (2004) OLG model, implementing the exact integration of the expectation functions. Building on this approach, Folini et al. (2021) alter the method by including the time as a state variable to solve for the transition path in the climate economics framework. By contrast, L. Maliar, S. Maliar, and Winant (2021) integrate stochastic processes with continuous transition density via the introduction of the All-in-One (AiO) integration parameter for model reduction and stabilisation of the Monte Carlo integration. They draw the state variables from ergodic sets to allow the neural networks to learn on the space where the “solution lives” and solve a Krusell and Smith (1998) model with 1000 agents (2001 state variables). In the follow-up paper, L. Maliar and S. Maliar (2020) adopt the algorithm from L. Maliar, S. Maliar, and Winant (2021) and solve a HANK model which includes savings through bonds and a ZLB on the nominal interest rate. Building upon the existing algorithm, L. Maliar and S. Maliar (2022) construct a classification deep learning method for modelling non-convex labour choices; Gorodnichenko et al. (2021) solve a version of a HANK model with individual and aggregate stochasticity; and Lepetyuk et al. (2020) solve a version of the "Terms of Trade Economic Model" (ToTEM) of the Bank of Canada with heterogeneous agents. Kase et al. (2022) extend the technique of L. Maliar, S. Maliar, and Winant (2021) by including the model parameters as state variables to simultaneously estimate the solutions for all parameter combinations in a HANK model with ZLB. All examples demonstrate that the proposed method can accurately approximate recursive equilibria in a high-dimensional state space with aggregate and individual uncertainty and strong non-linearities. The advantages of the L. Maliar, S. Maliar, and Winant (2021) and Azinovic et al. (2019) frameworks are that they provide a grid-free, global solution method to high-dimensional problems virtually at zero cost.

Firstly, it is possible because neural networks that approximate equilibrium functions are trained on simulated data to satisfy all equilibrium conditions directly, eliminating the need to solve sets of simultaneous nonlinear equations or optimisation problems to simulate the economy. Secondly, the training process allows for an effective parallelisation scheme, resulting in efficient computations.

2.2 Parameter estimation in the heterogeneous-agent models

Traditional frequentist (*i.e.* likelihood-based estimation techniques, Simulated Minimum Distance methods, Impulse-Response Function Matching methods, Generalised Method of Moments estimation) or Bayesian (*i.e.* Metropolis–Hastings algorithm, Sequential Monte Carlo techniques) estimation methods solve for the stationary model distribution and, then, implement perturbation techniques to align the aggregate dynamics around the stationary distribution with the data (see, *e.g.* Fernández-Villaverde, Rubio-Ramírez, et al. 2016; Auclert 2017). In the HA framework, the infinite dimension of the agent distribution has to be simplified to be easily estimated with standard methods. As discussed in Section 2.1.1, the infinite dimension of agent distribution is usually approximated with a simpler function form or described through an aggregate statistic. Otherwise, the model estimation of the heterogeneous-agent dynamic economic models does not differ from the representative-agent setting.

We note the following studies among recent papers on estimating model parameters in the HA models. In application to a HANK model with incomplete insurance, Challe et al. (2017) simplify the infinite-dimensional cross-sectional wealth distribution by approximating it with finite space and then recovering the distribution of the model parameters with Sequential Monte Carlo. A study conducted by Auclert (2017) used a HANK model of the Italian economy to evaluate how monetary policy affects consumption. The agent distribution was represented using a MA(∞) approach, and the model parameters were estimated by comparing the model with real moments of the data distribution using Bayesian inference. At last, in application to a continuous-time HANK model, Bayer and Luetticke (2018) use a full-information Markov Chain Monte Carlo to estimate the posterior distributions of the non-steady-state parameters conditional on individual agents' investment behaviour.

It should be noted that computational inefficiencies in the high-dimensional model solution translate into the inadequacy of the parameter estimation as the models have to be solved multiple times with different parameter combinations to arrive at a reasonable estimated value. Thereby, for the further development of our neural network framework, we consider approximating the model solution for the whole parameter space to augment the speed of further parameter estimation process.

2.2.1 Machine Learning for estimation of the model parameters

Compared to the literature on model solution, the literature on Machine Learning usage for the likelihood computation of the model fit with data is scarce. To our knowledge, only Fernández-Villaverde, Hurtado, et al. (2019) and Kase et al. (2022) address this issue. Fernández-Villaverde, Hurtado, et al. (2019) use neural networks to approximate the maximum likelihood function of the capital growth rate volatility parameter in a medium-scale continuous time HANK model with financial frictions. Afterwards, they maximise the obtained likelihood on a simple grid using a standard procedure. The advantage of using neural networks in their set-up is the ability to account for the microdata on wealth distribution to align the model production dynamics with the U.S. aggregate GDP growth time series. By contrast, Kase et al. (2022) introduce a secondary neural network that approximates the results of the particle filter likelihood estimations. To solve a medium-scale HANK model with aggregate and individual risks, they incorporate the model parameters into the neural network as pseudo-variables of the economy. Then, they use the particle filter a limited number of times to approximate the output with a surrogate neural network. Finally, they maximise the likelihood by applying stochastic gradient iteration over the whole parameter space. Their approach allows running the particle filter less frequently while still estimating the likelihood across the entire parameter space.

3 Algorithm description

This section describes an algorithm for solving the dynamic economic models in their non-linear specification based mainly on L. Maliar, S. Maliar, and Winant (2021) and Kase et al. (2022). Bypassing the linearisation step during solution allows accounting for such non-linearities as occasionally binding constraints or regime switching, which would be reduced to its steady state realisation in the case of linearisation. We formulate the algorithm in the way that would allow to estimate the model parameters in an efficient manner while solving the model only once for the whole parameter space.

3.1 The optimisation problem

We consider a widely used in the DSGE modelling class of discrete-time Markov dynamic economic models with time-invariant decision functions. The economies in these models can be represented as vectors of endogenous and exogenous state variables.

The *exogenous state* variable vector $\mathbb{M}_t \in \mathbb{R}_t^m$ follows a Markov process subject to i.i.d. exogenous shocks ε_t with transition function $M(\cdot)$ and the model parameters Θ:

$$\mathbb{M}_t = M(\mathbb{M}_{t-1}, \varepsilon_t | \Theta).$$

The exogenous state drives the vector of *endogenous states* $\mathbb{S}_t \in \mathbb{R}^s$ through its realisation and through the control variable $\mathbb{X}_t \in \mathbb{R}^x$ according to a transition function $S(\cdot)$:

$$\mathbb{S}_t = S\left(\mathbb{M}_t, \mathbb{S}_{t-1}, \mathbb{X}_{t-1}, \varepsilon_t | \Theta\right).$$

The control variable $\mathbb{X}_t$ characterises the agents' optimal policy choices. The agents observe the realisation of the state variables before making the decision. Therefore, the decision function can be represented as a mapping from the state variables to control variables:

$$\mathbb{X}_t = X(\mathbb{S}_t, \mathbb{M}_t|\Theta);$$

with x, s, and m being the number of the control, endogenous, and exogenous variables, respectively.

The solution of the model is the non-linear time-invariant function $X(\cdot)$ that maximises the agents' lifetime reward in the form:

$$\max_{\{\mathbb{S}_{t+1}, \mathbb{X}_t,\}_{t=0}^{\infty}} \sum_{\tau=t}^{\infty} \beta^{\tau-t} \mathbb{E}_{(\varepsilon_{t+1}, \varepsilon_{t+2}, \dots|\Theta)} \left[U(\mathbb{S}_\tau, \mathbb{M}_\tau, \mathbb{X}_\tau|\Theta) \right], \tag{1}$$

where $\beta \in [0,1)$ is the discount factor, $U(\cdot)$ is the agent's reward function, and $\mathbb{E}_{(\varepsilon_{t+1}, \varepsilon_{t+2}, \dots)}$ is the period t expectation of the future shocks realisations $(\varepsilon_{t+1}, \varepsilon_{t+2}, \dots)$ conditional on the initial state realisation $\mathbb{M}_t, \mathbb{S}_t$ and the model parameter vector Θ.

We derive the solution to the optimal control problem (1) using the Euler equation method. As shown by Santos (2000), the Euler equation residuals are a good measure of numerical solution accuracy. Therefore, this method can be considered a suitable candidate for the neural network loss function. Additionally, L. Maliar, S. Maliar, and Winant (2021) demonstrate that minimising the Euler equation residual objective yields the most stable results when solving economic models with neural networks, compared to direct lifetime-reward maximisation and Bellman equation residual minimisation approaches.

If the objective functions are differentiable, the solution could be characterised by a set of J first-order conditions (dropping the t indices where unnecessary):

$$\mathbb{E}_{\varepsilon|\Theta} \left[f_j (\mathbb{M}_t, \mathbb{S}_t, \mathbb{X}_t, \mathbb{M}_{t+1}, \mathbb{S}_{t+1}, \mathbb{X}_{t+1}) \right] = 0;$$
$$f_j : \mathbb{R}^m \times \mathbb{R}^s \times \mathbb{R}^x \times \mathbb{R}^m \times \mathbb{R}^s \times \mathbb{R}^x \to \mathbb{R},\ j = 1,\dots,J$$

If we consider function $\varphi(\cdot|\Theta)$ a numerical approximation of the decision rule $X(\cdot|\Theta)$, the squared residuals of the Euler equation (risk function), associated with it, would have the form:

$$\Xi(\varphi|\Theta) = \mathbb{E}_{(\mathbb{M}_t,\mathbb{S}_t)} \left\{ \sum_{j=1}^{J} v_j \left(\mathbb{E}_\varepsilon \left[f_j (\ \mathbb{M}_t, \mathbb{S}_t, \varphi(\mathbb{M}_t, \mathbb{S}_t|\Theta), \right. \right. \right. \qquad (2)$$
$$\left. \left. \left. \mathbb{M}_{t+1}, \mathbb{S}_{t+1}, \varphi(\mathbb{M}_{t+1}, \mathbb{S}_{t+1}|\Theta) \ ) \right) \right] \right\}^2 \to 0,$$

where v_j is the assigned weight of the j^{th} optimality condition in the empirical risk function. With respect to the expectation of the initial state value realisation $\mathbb{E}_{(\mathbb{M}_t,\mathbb{S}_t)}$ and the expectation $\mathbb{E}_\varepsilon$ of the $(t+1)$ period shock ε. The optimal decision rule function $\varphi(\cdot)$, thus, is the one that solves $\min_\varphi \Xi(\varphi|\Theta)$.

3.2 Accounting for heterogeneity in the framework

Although our approach can be used to solve dynamic economic models with representative agents (see Sections 4.1 and 4.2), this paper introduces a solution algorithm for heterogeneous-agent models. Heterogeneity is commonly modelled as a continuum of agents, implying that initial states and shocks are distributed infinitely. In order to address the infinite-state variables, we approximate the continuum using a sufficiently large but finite number of agents L as in L. Maliar, S. Maliar, and Winant (2021). Consequently, state variables and shocks can be rewritten in terms of idiosyncratic and aggregate components:

$$\mathbb{S}_t = \left\{ \left\{ \mathbb{S}_t^i \right\}_{i=1}^{L}, \mathbb{S}_t^A \right\}, \mathbb{M}_t = \left\{ \left\{ \mathbb{M}_t^i \right\}_{i=1}^{L}, \mathbb{M}_t^A \right\}, \text{ and } \varepsilon_t = \left\{ \left\{ \varepsilon_t^i \right\}_{i=1}^{L}, \varepsilon_t^A \right\}.$$

The optimal decision functions, thus, too, admit the divided form:

$$\mathbb{X}_t^i \sim \varphi^I(\mathbb{S}_t, \mathbb{M}_t, \mathbb{S}_t^i, \mathbb{M}_t^A|\Theta) \text{ and } \mathbb{X}_t^A \sim \varphi^A(\mathbb{S}_t, \mathbb{M}_t|\Theta);$$

with $\mathbb{X}_t^A$ and $\mathbb{X}_t^i$ being the aggregate policy and individual i's choice in the period t, and φ^A and φ^I being time-invariant decision functions - solution of the model. We consider that the agents differ only in individual state variable realisation, and the general individual function φ^I, conditional on the state variables, can be used for all individual agents.[6]

The transition of the state and control variables can, therefore, be rewritten in the HA framework as:

$$\begin{aligned}
&\mathbb{M}_t = \left\{\mathbb{M}_t^A, \{\mathbb{M}_t^i\}_{i=1}^L\right\} = M\left(\mathbb{M}_{t-1}^A, \{\mathbb{M}_{t-1}^i\}_{i=1}^L, \varepsilon_t^A, \{\varepsilon_t^i\}_{i=1}^L |\Theta\right);\\
&\mathbb{S}_t = \left\{\mathbb{S}_t^A, \{\mathbb{S}_t^i\}_{i=1}^L\right\} = S\left(\mathbb{S}_{t-1}^A, \{\mathbb{S}_{t-1}^i\}_{i=1}^L, \mathbb{M}_t^A, \{\mathbb{M}_t^i\}_{i=1}^L, \mathbb{X}_t^A, \{\mathbb{X}_t^i\}_{i=1}^L, \varepsilon_t^A, \{\varepsilon_t^i\}_{i=1}^L |\Theta\right);\\
&\mathbb{X}_t = \left\{\mathbb{X}_t^A, \{\mathbb{X}_t^i\}_{i=1}^L\right\} = \left\{\varphi^A(\mathbb{S}_t, \mathbb{M}_t|\Theta), \left\{\varphi^I(\mathbb{S}_t, \mathbb{M}_t, \mathbb{S}_t^i, \mathbb{M}_t^A|\Theta)\right\}_{i=1}^L\right\}.
\end{aligned}$$

The number of the decision function inputs, then, adds up to $s = s^i \times L + s^A$ endogenous state variables and $m = m^i \times L + m^A$ exogenous state variables. The number of the output decision or policy functions is $o = o^i \times L + o^A$. For each number n in (s,m,o), the notations n^i and n^A refer to the dimensionality of the individual and aggregate vectors, respectively.

3.3 Using the neural network to solve the model

While the general notation of the $\varphi(\cdot)$ decision rule approximation can adopt any functional form, we choose neural networks as the base function for the numerical solution. They are well-suited for solving high-dimensional models as they can handle many inputs without encountering the "dimensionality curse". This feature is especially helpful when augmenting the state space with individual state and shock variables as well as model parameters, which we have described in Section 3.2 and Section 3.4. While such space expansion would pose computational problems for classical numerical methods, the neural

[6] The difference between agent types could be accommodated into the neural network by adding the categorical variable of the agent's type. *E.g.*, it may be a natural choice for modelling the "categorical" heterogeneity types (such as comparing two types of consumers: Ricardian and non-Ricardian as in Gali et al. (2004); or two types of financial market participants as in Fernández-Villaverde, Hurtado, et al. (2019)) instead of continuous heterogeneity which is explored in this paper. However, this experiment is left for future research.

networks can handle large inputs because of their scalability. Moreover, due to their structure, neural networks can approximate highly complicated mathematical functions or mappings.[7] This ability to approximate even non-linear functions is crucial for models with aggregate and individual uncertainty.

The "aggregate" and "individual" neural networks map the inputs into the otput vector of the dimensionality $o = o^i \times L + o^A$ using non-linear transformations. The individual and aggregate decisions at the period t, then, are represented as:

$$\mathbb{X}_t^A = \varphi_{NN}^A(\mathbb{S}_t, \mathbb{M}_t | \Theta) \text{ and } \mathbb{X}_t^i = \varphi_{NN}^I(\mathbb{S}_t, \mathbb{M}_t, \mathbb{S}_t^i, \mathbb{M}_t^i | \Theta),$$

where φ_{NN}^A and φ_{NN}^I are the neural network approximations of the model solution for aggregate and individual decision functions respectively. Consequently, the entire vector of control variables has the form of:

$$\mathbb{X}_t = \left\{ \mathbb{X}_t^A, \{\mathbb{X}_t^i\}_{i=1}^L \right\} \sim \left\{ \varphi_{NN}^A(\mathbb{S}_t, \mathbb{M}_t | \Theta), \left\{ \varphi_{NN}^I(\mathbb{S}_t, \mathbb{M}_t, \mathbb{S}_t^i, \mathbb{M}_t^A | \Theta) \right\}_{i=1}^L \right\}.$$

3.4 Including model parameters as pseudo-state variables

Parameter estimation requires solving the model multiple times with different parameter sets. This may become computationally exhaustive with extensive models, and Kase et al. (2022) propose adding the parameter values to the model-solving neural network as inputs. Consequently, the neural network takes more time and iterations to converge but only has to be trained once to get the decision function for the entire parameter space. However, the benefit of this approach is that after the model is solved, the estimation step can be run

[7] The inclusion of the non-linear functions between fully connected layers makes the neural networks suited for approximating functions with kinks, discontinuities, discrete choices, and switching. Moreover, their design makes them linearly scalable, robust to ill-conditioning, and capable of model reduction. For a general overview of deep learning, see Goodfellow et al. (2016).

numerous times on the solution that is "conditioned" on a particular parameter combination to evaluate the corresponding economic dynamics.

We divide the model parameter set into sets of parameters for estimation and calibration:

$$\Theta = \left\{\tilde{\Theta}, \bar{\Theta}\right\},$$

where $\tilde{\Theta}$ is a set of parameters that must be estimated, and the $\bar{\Theta}$ is the set of calibrated parameters. The parameters to be estimated $\tilde{\Theta}$ are thus passed as inputs into the neural network, which treats them as pseudo-economic variables. As the calibrated parameters are not varied during the neural network training, the numerical solution will be "conditioned" on them. Using both sets of parameters, the decision functions can be parametrised with the neural network as:

$$\mathbb{X}_t = X(\mathbb{S}_t, \mathbb{M}_t | \Theta) \sim \varphi_{NN}(\mathbb{S}_t, \mathbb{M}_t, \tilde{\Theta} | \bar{\Theta});$$

and the total number of inputs is then calculated by adding up the number of exogenous state variables m, endogenous state variables s, and pseudo-state variables p. As a result, after training, the neural-network-based decision rules $\varphi_{NN}(\cdot)$ can can be applied to solve models with any parameter combination within the set parameter space.

3.5 The All-in-One expectation operator used to construct the neural network loss function

In literature, (2) is usually solved by iterating over the fixed grid of points in the relevant state and parameter spaces. However, for models with a large number of equilibrium optimality conditions and high-dimensional shocks, this approach becomes computationally infeasible as the grid is augmented exponentially with additional space vectors.[8]

Following L. Maliar, S. Maliar, and Winant (2021), we assume that the initial states and the second-period shock realisations are drawn from the given distribution. Then, under independent second-period draws of the exogenous shocks ε_1 and ε_2, the expectation of the product of the $f(\cdot)$ function under the random draws is $\mathbb{E}_{\varepsilon_1}[f(\varepsilon_1)] \cdot \mathbb{E}_{\varepsilon_2}[f(\varepsilon_2)] = \mathbb{E}_{(\varepsilon_1,\varepsilon_2)}[f(\varepsilon_1)f(\varepsilon_2)]$.

Therefore, the expectation operators $\mathbb{E}_{(\mathbb{M},\mathbb{S})}$ and $\mathbb{E}_{\varepsilon}$ can be merged, giving:

$$\begin{aligned}\Xi(\varphi|\Theta) = \mathbb{E}_{(\mathbb{M}_t,\mathbb{S}_t,\varepsilon_1,\varepsilon_2|\Theta)}\Big\{ \sum_{j=1}^{J} v_j \left[f_j(\mathbb{M}_t,\mathbb{S}_t,\mathbb{X}_t,\mathbb{M}_{t+1},\mathbb{S}_{t+1},\mathbb{X}_{t+1})\,|_{\varepsilon=\varepsilon_1}\right] \\ \cdot \left[f_j(\mathbb{M}_t,\mathbb{S}_t,\mathbb{X}_t,\mathbb{M}_{t+1},\mathbb{S}_{t+1},\mathbb{X}_{t+1})\,|_{\varepsilon=\varepsilon_2}\right]\Big\}.\end{aligned} \tag{3}$$

3.6 Training the neural networks for model solution

An important distinction of using neural networks for minimisation of the (3) is that the task is viewed as an ordinary regression instead of a numerical computational problem. In particular, the neural network inputs $\{\mathbb{M}_t,\mathbb{S}_t,\varepsilon,\tilde{\Theta}\}$ are viewed as state parameters of the economy and no distinction is drawn between them.

A general idea of training a neural network is to initiate the weights of the fully-connected layers (the layers are separated by a non-linear activation function to allow for the non-linearity in the solution approximation) and update them iteratively using the anti-gradient of the constructed loss function, which is defined in a unique way that corresponds to the problem. To build the loss function, we construct an empirical version of the (3) by simulating N economies with different draws of estimated parameters $\tilde{\Theta}$ and of the initial distribution of the state variables $\{\mathbb{M}_t,\mathbb{S}_t\}$. Further, we make two draws of

[8] For example, even sparse Smolyak grid method can fail if the dimensionality of the state space exceeds ~ 20. For an in-depth take on other solution methods for the high-demensional problems refer to L. Maliar and S. Maliar (2014).

the $(t+1)$ period exogenous shocks $\{\varepsilon_1,\varepsilon_2\}$. The economies are run one step forward to construct the empirical risk function:

$$\Xi_{NN}(\varphi_{NN}|\Theta) = \min_{\varphi_{NN}|\Theta} \frac{1}{N} \sum_{n=1}^{N} [\, \sum_{j=1}^{J} v_j \left[f_j(\mathbb{M}_{n,t}, \mathbb{S}_{n,t}, \mathbb{X}_{n,t}, \mathbb{M}_{n,t+1}, \mathbb{S}_{n,t+1}, \mathbb{X}_{n,t+1}) \,|_{\varepsilon=\varepsilon_1} \right] \cdot \left[f_j(\mathbb{M}_{n,t}, \mathbb{S}_{n,t}, \mathbb{X}_{n,t}, \mathbb{M}_{n,t+1}, \mathbb{S}_{n,t+1}, \mathbb{X}_{n,t+1}) \,|_{\varepsilon=\varepsilon_2} \right] \,], \tag{4}$$

where $\mathbb{X}_t$ is the neural network approximation of the control variable vector $\mathbb{X}_t \sim \varphi_{NN}(\mathbb{S}_t, \mathbb{M}_t|\Theta)$.

The equilibrium conditions for the (4) are constructed using the standard Euler-equation approach; their number is minimised to be in line with the number of the neural network outputs. It should be noted that in the HA setup, the set of equilibrium equations consists of both aggregate and individual optimality and equilibrium conditions and transition functions. Thus, the number of the Euler equation is minimised to correspond to $o = o^i \times L + o^A$. Certain model types imply additional constraints on the aggregate or individual decision functions: *e.g.* binding constraints on borrowing or market equilibrium conditions. If that is the case for the model, the constraints, adapted to assume a differentiable form, are included in the neural network loss function in addition to the Euler equation residuals.

3.7 Stochastic solution domain

The initial values of the state variables (both endogenous and exogenous) are drawn randomly before each iteration. As argued in K. Judd et al. (2009) and L. Maliar and S. Maliar (2014), simulation methods may improve the solution convergence for the high-dimensional models as they solve the model only in the relevant - ergodic - space. Following the takes from simulation methodology, we draw the state variables from the corresponding ergodic distributions, which reduces the problem space of an n-dimensional hypercube to an n-dimensional hypersphere (n being the numbers of state variables).[9]

[9] Another approach might include adjustment of the active learning space as in Renner and Scheidegger (2018) or in Kase et al. (2022).

3.8 Parameter space

We restrict each estimated parameter to lie within specific bounds in order to specify the parameter space, where we suppose the economy lies:

$$\tilde{\Theta} = \left\{ \left[\tilde{\theta}_1^{lower}, \tilde{\theta}_1^{upper}\right], \left[\tilde{\theta}_2^{lower}, \tilde{\theta}_2^{upper}\right], \dots, \left[\tilde{\theta}_p^{lower}, \tilde{\theta}_p^{upper}\right] \right\},$$

where $\tilde{\theta}_i^{lower}$ and $\tilde{\theta}_i^{upper}$ are lower and upper bounds of the i^{th} estimated parameter. Parameters for each iteration of the algorithm are drawn randomly from the state space. Then, after the economy is simulated one step forward and the neural network loss is computed and fed-forward to adjust the weights in the fully-connected layers, the parameters $\tilde{\Theta}$ are redrawn for the new iteration. As a result, in each batch of parallel-running economies, the parameters differ significantly, which allows us to attain the neural network solution for the whole parameter space.

We have also explored the approach of Kase et al. (2022) of simulating and training the neural network forward for T periods, but it did not result is a significant increase of the solution accuracy while slowing down the neural network training significantly; thus, we opted out of using this approach.

4 Numerical analysis of dynamic economic models

In this section, we compare the behaviour of the neural network solution algorithm for several problems. First, we consider a simple consumption-saving model with two control variables and four structural shocks to show the general principle of incorporating a dynamic economic model with multiple shocks and estimated parameters into the framework. Next, we move to a New Keynesian DSGE model with monopolistic competition, three control variables and one shock to assess the neural network framework performance on a still simple model, but one that produces relevant macroeconomic results. At last, explain the difficulties we have encountered while applying our framework to the heterogeneous-agent models and propose possible treatments.

4.1 Consumption-saving problem

Optimisation problem. To demonstrate the performance of the algorithm, we first consider a simple one-agent consumption-saving problem of choosing $\{c_\tau, w_{\tau+1}\}_{\tau=t}^{\infty}$ to maximise a discounted utility $u(c_t)$ subject to a transition equation and a borrowing constraint:

$$\begin{aligned}
&\max_{\{c_t, w_{t+1}\}_{t=0}^{\infty}} \mathbb{E}_t \left[\sum_{\tau=t}^{\infty} B_\tau \beta^{\tau-t} u(c_\tau) \right] \\
&\text{s.t. } w_{t+1} = (w_t - c_t) \cdot \bar{R} \cdot R_{t+1} + Y_{t+1}, \\
&c_t \leq w_t, \text{ where:}
\end{aligned} \tag{5}$$

- w_t is the agent's wealth at the beginning of the period t and c_t is their consumption during the period t;
- $u(c_t)$ is a Cobb-Douglas utility function $u(c_t) = \frac{1}{1-\gamma}\left(c_t^{1-\gamma} - 1\right)$;
- $\beta \in [0,1)$ is a discount factor;
- the logarithm of the real return on investment $r_t = \ln(R_t)$ follows an AR(1) processes $r_{t+1} = \rho_r r_t + \sigma_r \varepsilon_t^r$, and $\bar{R}$ is the target (steady state) interest rate;

- the production is exogenous and dependent on the transitory and permanent production components such that the logarithm of production can be represented as $\ln Y_t = p_t + q_t$, with both transitory and permanent production components following the AR(1) process respectively $q_{t+1} = \rho_q q_t + \sigma_q \varepsilon_t^q$ and $p_{t+1} = \rho_p p_t + \sigma_p \varepsilon_t^p$;
- the logarithm of the preference shock $b_t = \ln(B_t)$ follows an AR(1) process $b_{t+1} = \rho_b b_t + \sigma_b \varepsilon_t^b$;
- the exogenous shocks $\varepsilon_t^r, \varepsilon_t^y$, and ε_t^b assume a standard normal distribution $\sim N(0,1)$.

The Lagrangian of the problem:

$$\begin{aligned}\mathscr{L} = \mathbb{E}_t \left[\sum_{\tau=t}^{\infty} B_\tau \beta^\tau u(c_\tau) \right] &+ \mathbb{E}_t \sum_{\tau=t}^{\infty} \mu_\tau (w_\tau - c_\tau) \\ &+ \mathbb{E}_t \sum_{\tau=t}^{\infty} \lambda_\tau \left(w_{\tau+1} - (w_\tau - c_\tau) \cdot \bar{R} \cdot R_{\tau+1} - Y_{\tau+1}\right)\end{aligned}$$

where λ_τ, μ_τ are the Lagrange multipliers associated with the constraints $w_{\tau+1} = (w_\tau - c_\tau) \cdot \bar{R} \cdot R_{\tau+1} + Y_{\tau+1}$ and $c_\tau \leq w_\tau$, respectively. According to the Karush-Kuhn-Tucker (KKT) conditions for this Lagrangian, the combination of the primary and dual feasibility with the complementary slackness (substituting the Lagrange mutipliers with the gradients of the Lagrangian in respect to the control variables), define the following system of the solution conditions:

$$\begin{cases} a \equiv w_t - c_t \geq 0, \\ b \equiv u'(c_t) - \beta \bar{R} \mathbb{E}_t \left[u'(c_{t+1}) \frac{B_{t+1} \cdot R_{t+1}}{B_t} \right] \geq 0, \\ a \cdot b = 0. \end{cases}$$

The system can be described using the function $\min\{a,b\} = 0$, which is not differentiable and, thus, unsuitable for the neural network loss function. To address this problem, we adopt the smoothened version of the KKT conditions in the form of the Fischer-Burmeister function:

$$F_{FB}(a,b) = a + b - \sqrt{a^2 + b^2} = 0.$$

Moreover, for a convenient definition of the neural network, we represent the components a,b in a unit-free form, which makes the model approach the consumption to the available wealth endowment to avoid falling into local extreme values or running into the lower bound[10]:

$$\begin{aligned} A &\equiv \frac{c_t}{w_t},\ 0 \le A \le 1 \\ B &\equiv \beta \bar{R} \mathbb{E}_\varepsilon \left[\frac{u'(c_{t+1})}{u'(c_t)} \cdot \frac{B_{t+1} \cdot R_{t+1}}{B_t} \right], \end{aligned}$$

This gives us the following Fischer-Burmeister function:

$$F_{FB}(1-A, 1-B) = (1-A) + (1-B) - \sqrt{(1-A)^2 + (1-B)^2}.$$

Notably, we have switched the expectation operator notation from $\mathbb{E}_t$ to $\mathbb{E}_\varepsilon$ as the only stochastic component in the Euler equation in respect to the period t is the vector of exogenous shocks ε.

Neural network parametrisation. Using the estimation framework described in Section 3, we define the state and the control variables of the model as:

[10] As demonstrated in the practical applications, an explicit definition of the approximate agent behaviour is necessary, in order to prevent the neural network from converging to suboptimal local values.

$$\mathbb{S}_t = \{w_t\}\,;\ \mathbb{M}_t = \{r_t,p_t,q_t,b_t\}\,;\mathbb{X}_t = \{c_t,w_{t+1}\}\,.$$

The set of calibrated parameters of the model is empty and the set of the estimated parameters consists of all model parameters - for demonstration purposes:

$$\bar{\Theta} = \{\}\,;\ \tilde{\Theta} = \left\{\beta,\gamma,\bar{r},\rho_r,\sigma_r,\rho_p,\sigma_p,\rho_q,\sigma_q,\rho_b,\sigma_b\right\}.$$

The Fischer-Burmeister-function form of the KKT conditions and the stationarity of the problem allows us to set constraints directly through the parametrisation of the neural network $\xi(\mathbb{S}_t,\mathbb{M}_t,\tilde{\Theta})$. Namely, we set the outputs of the neural network as a sigmoid function for the A component of the KKT condition and an exponent for the component B:

$$\begin{pmatrix} A \\ B \end{pmatrix} = \begin{pmatrix} \frac{1}{1+\exp(-\xi_1(\mathbb{S}_t,\mathbb{M}_t,\tilde{\Theta}))} \\ \exp(\xi_2(\mathbb{S}_t,\mathbb{M}_t,\tilde{\Theta})) \end{pmatrix},$$

with $\xi_1(\cdot)$ and $\xi_2(\cdot)$ being the first and second output vectors of the neural network, respectively. The inputs of the network are normalised to ensure the stability of the numerical approximation: the exogenous variables $\mathbb{M}_t$ are normalised to two ergodic standard deviations of the form $\sigma^e = \frac{\sigma}{\sqrt{1-\rho^2}}$, and the endogenous state variable $\mathbb{S}_t$ is set to lie within the minimum and maximum bound of its uniform distribution.

We parametrise the neural network using three fully-connected layers with 128 neurons each; the non-linear function used between the fully-connected layers is *SiLU* (Elfwing et al. 2017).[11] The optimiser used for the neural network training is Adam (Kingma and Ba

[11] Azinovic et al. (2019) have shown that the popular in most deep learning problems *ReLU* activation function does not approximate well mostly-smooth functions with kinks and flat areas; the experiments have also shown that the *SiLU* activation function shows better convergence and smoothness of the output than a *ReLU* or a sigmoid function.

2017), and the optimal learning rate was found to be 10^{-3}. All computations for this paper are conducted using the PyTorch library (Paszke et al. 2019) on the *Google Colaboratory* (2023) GPU engine.

Training the neural network. The objective for the neural network is minimising the sum of the Euler equation residuals under the AiO expectation parameter (4) with two realisations of shocks, and the squared Fischer-Burmeister function for the complementary slackness condition:

$$\begin{aligned}
&Loss(\xi,\Theta) = \mathbb{E}_{\mathbb{S}_t,\mathbb{M}_t,\varepsilon_1,\varepsilon_2}\left[\underbrace{F_1\left(\mathbb{S}_t,\mathbb{M}_t,\varepsilon_1,\Theta\right)F_1\left(\mathbb{S}_t,\mathbb{M}_t,\varepsilon_2,\Theta\right)+v\left(F_2\left(\mathbb{S}_t,\mathbb{M}_t\right)\right)^2}_{\varphi_{NN}(\xi,\mathbb{S}_t,\mathbb{M}_t,\varepsilon_1,\varepsilon_2,\Theta)}\right];\\
&F_1(\mathbb{S}_t,\mathbb{M}_t,\varepsilon,\Theta) = \beta\bar{R}\mathbb{E}_\varepsilon\left[\frac{u'\left(c_{t+1}\right)}{u'(c_t)}\cdot\frac{B_{t+1}\cdot R_{t+1}}{B_t}\right]-B;\\
&F_2\left(\mathbb{S}_t,\mathbb{M}_t,\Theta\right) = F^{FB}(1-A,1-B),
\end{aligned}\tag{6}$$

where v is the weight of the complementary slackness component in the loss function. The state variable transition needed to proceed from the period t to $(t+1)$ in the Euler equation $F_1(\mathbb{S}_t,\mathbb{M}_t,\varepsilon,\Theta)$ under the corresponding shock realisation is complacent with (5). At last, the neural network updates the outputs to solve: $Loss(\xi,\Theta) \to \min\limits_{\{\xi|\Theta\}}$, where ξ is a neural network, used to derive all decision rules approximations φ_{NN}.

The parameters $\tilde{\Theta}$, used as inputs in the neural network, are generated on the uniformly-distributed space,[12] the lower and upper bounds for each parameter are presented in Table 1. For simplicity, the initial value of the only endogenous state variable w_t is drawn from the uniform distribution $w_t \sim U[0.1;4]$. The logarithms of the exogenous state variables $\{r_t,p_t,q_t,b_t\}$ are drawn from the ergodic distribution: for the AR(1) processes with normal shocks, the ergodic space can be characterised as a normal distribution with a standard

[12] In this paper, we opt for the uniform distribution to select parameters from the relevant parameter space for each economy. However, with certain types of models, it might be beneficial to use more suitable prior parameter distributions.

deviation, adjusted for the autoregressive component: $\{r_t,p_t,q_t,b_t\} \sim N(0,\frac{\sigma}{\sqrt{1-\rho^2}})$. The exogenous shocks $\varepsilon = \{\varepsilon^r,\varepsilon^p,\varepsilon^q,\varepsilon^b\}$ are drawn from the standard normal distribution.

Table 1: Estimated parameters of the consumption-saving problem

Parameter	Parameter name	Lower bound	Upper bound	Calibrated value
β	Discounting factor	0.92	0.99	0.9
γ	Preference parameter in the Cobb-Douglas utility function	1.8	2.2	2.0
$\bar{R}$	Target interest rate	1.0	1.1	1.04
ρ_r	Persistence of the interest rate shock	0.19	0.21	0.2
σ_r	Std. of the interest rate shock	0.0005	0.0015	0.001
ρ_p	Persistence of the permanent production component shock	0.92	0.97	0.999
σ_p	Std. of the permanent production component shock	0.00005	0.00015	0.0001
ρ_q	Persistence of the transitory production component shock	0.92	0.99	0.9
σ_q	Std. of the transitory production component shock	0.0005	0.0015	0.001
ρ_b	Persistence of the preference shock	0.18	0.22	0.2
σ_b	Std. of the preference shock	0.0005	0.0015	0.001

In order to train the network, we run 10,000 iterations, each iteration processing a batch of 500 economies using randomly drawn parameters $\tilde{\Theta}$ and state variables w_t from their respective spaces. We attempted to implement the approach suggested by Kase et al. (2022), which involved running economy simulations for T time steps and updating the loss function after each step to train the neural network on a state distribution that closely resembled the ergodic state for a specific parameter set. However, this method did not result in better approximations and significantly increased computation time. Therefore, we decided to draw new sets of parameters $\tilde{\Theta}$, state variables, and shocks $\{\mathbb{S}_t,\mathbb{M}_t,\varepsilon_1,\varepsilon_2\}$ before each iteration.

Figure 1, represents the progression of the Euler equation residuals. According to Santos (2000), it is a reliable indicator of the solution function approximation accuracy and can be used to compare different approximations. Therefore, we have used them to assess the accuracy of our neural network solution algorithm.[13] In order to demonstrate that the

[13] It is important to note that the Euler equation residuals can only be compared among different approximation strategies applied to the same model, not between different models.

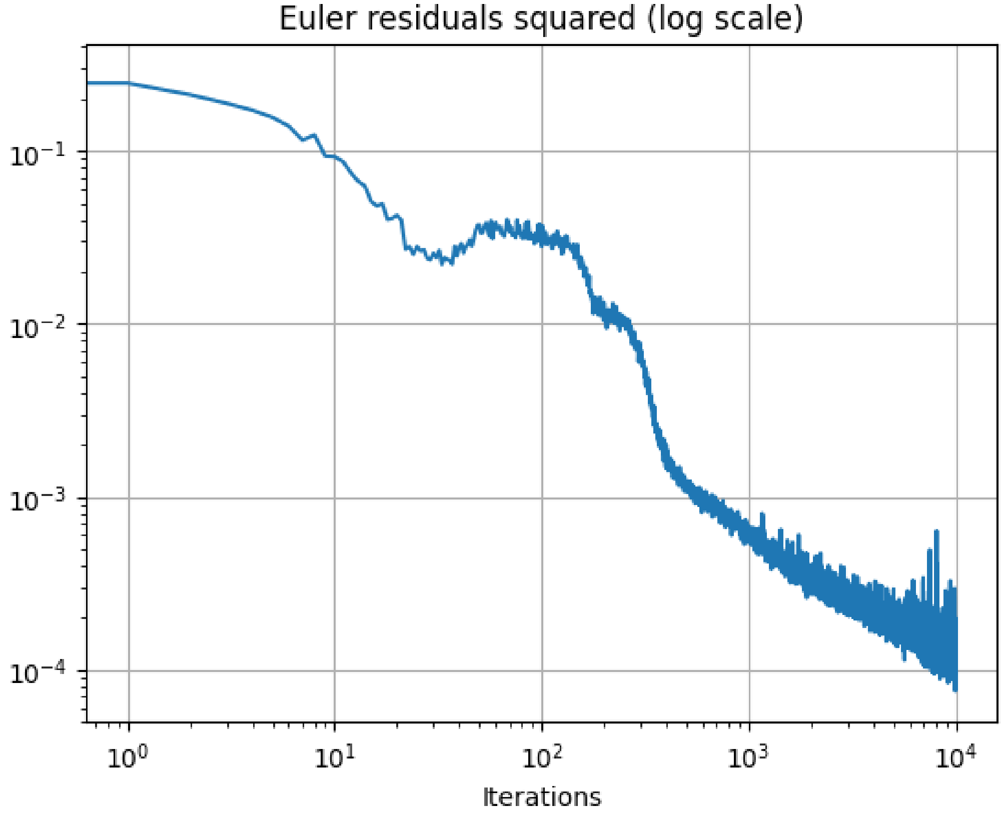

Figure 1: Progression of the residuals of the Euler equation during the neural network training for the consumption-saving model, computed using the standard method as in Santos (2000), without the AiO (see Section 3.5) expectation parameter.

inclusion of the parameters $\tilde{\Theta}$ as inputs of the neural network does not impact the accuracy of the solution in a simple consumption-saving model, we have compared the residuals obtained using our methodology with those obtained using the algorithm developed by L. Maliar, S. Maliar, and Winant (2021) on the same model. The comparison of the Euler residuals for 10,000 generated economies with the exact values of the estimated parameters $\tilde{\Theta}$ has shown that the difference between the accuracy of our method and the one presented in L. Maliar, S. Maliar, and Winant (2021) is statistically insignificant, which proves the reliability of our technique for small problems.

To prove that the neural network retrieves a meaningful approximation of the decision function, we asses the elasticity of the consumption over the varying parameters, holding

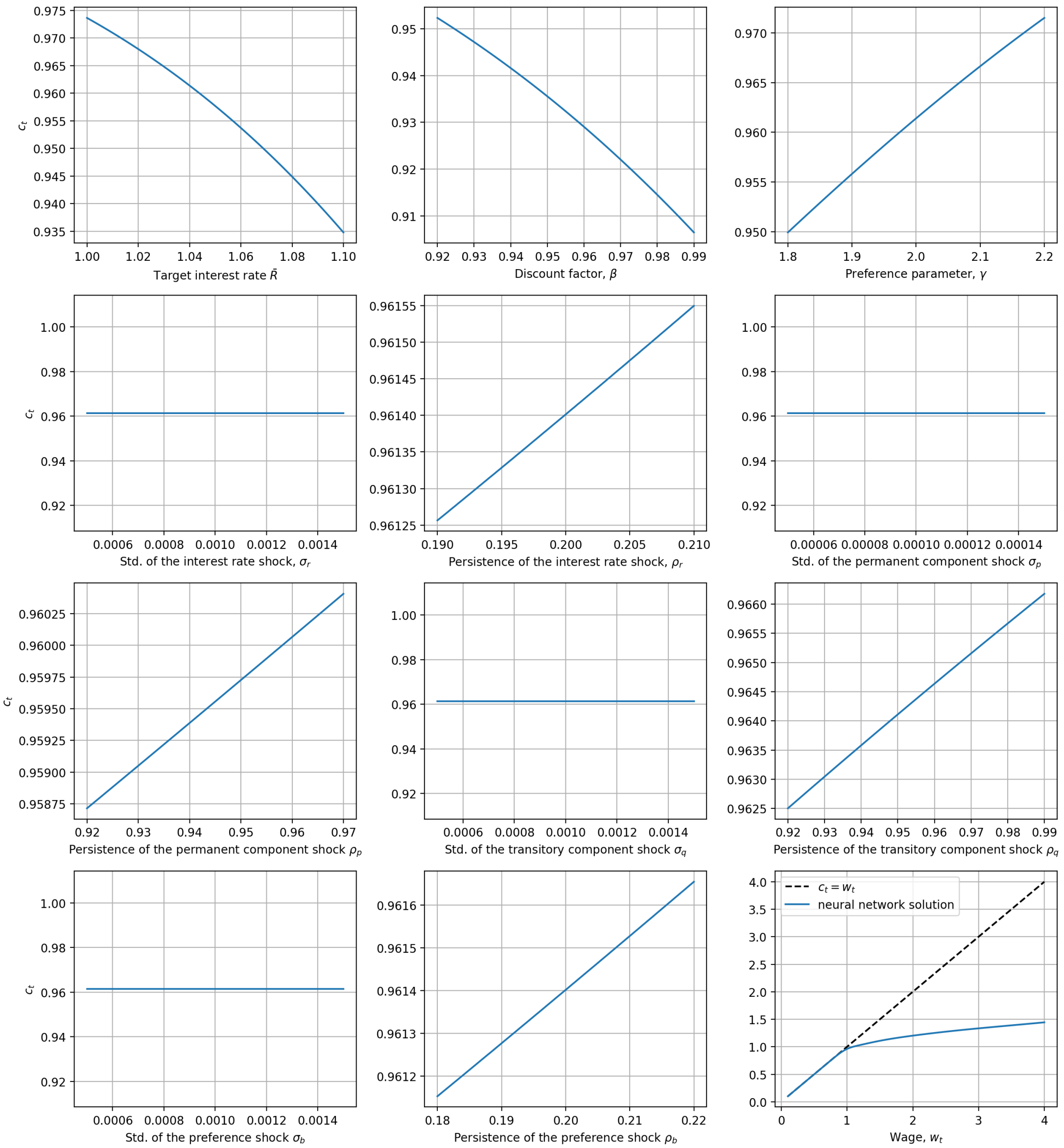

Figure 2: The graph illustrates changes in the decision function for consumption in a simple consumption-saving model based on the structural parameter value, while the remaining parameters are determined by the calibrated value (see the last column of Table 1); all parameters vary within a range from their minimum to maximum values. The initial wealth w_t is set to one, and exogenous shocks ε are set to zero. The last graph shows the neural network solution for the consumption rule in a steady state (with exogenous shocks equal to zero).

other parameters and the initial wealth distribution equal. The data depicted in Figure 2 shows that the neural network output is not affected by changes in the standard deviations of shocks but does respond positively to the greater shock persistence. This is because the problem is stationary, and higher standard deviations do not affect the expected shock realisation, which is the most crucial part of the shock in simple models like consumption-saving. On the other hand, the higher predictability of the next period's shock outcome explains the positive response. The agent's utility function has a positive elasticity over the intertemporal consumption preference parameter γ within the described values, corresponding with the positive angle on the graph; the relationship between the discount factor or the target interest rate and the current consumption is negative as the higher values of $\bar{R}$ and β augment "price" of the current period's consumption.

The graphic analysis of the solution has shown that the outputs of the neural network are interpretable, which addresses the common concern of neural networks being an intractable solvers. Nevertheless, it has to be noted that the solution depends highly on the initial value of the neural network weights, the neural network architecture, the bounds of the parameters $\tilde{\Theta}$, and the formulation of the loss function (for example, on the weight v of the complementary slackness component of the neural network loss function). The appropriate combination of these factors has to be found to achieve a stable solution without falling into the local minimum. For complex models with a large number of local minima, it is a disadvantage as it has to be verified every time that the solution has converged to its global state.

4.2 Standard NK DSGE model

To look at the algorithm performance for a more complex representative agent problem, we consider a New Keynesian DSGE model with monopolistic competition based on Dixit and Stiglitz (1977).[14] The model considers an economy with a continuum of differentiated intermediary goods sold in a market with monopolistic competition. The only final consumption good produced according to a Constant Elasticity of Substitution (CES) function

[14] The model used for the numerical analysis in this section is based on the lecture notes from the course on Business Cycles by Oxana Malakhovskaya (2021).

by a representative firm is sold in a perfectly competitive market. The model presented here deviates from the basic consumption-saving model also by including the individual control variable of labour supply (represented as L_t) and introducing a second agent in the economy - the firm.

Households. The representative household chooses control variables $\mathbb{X}_t = \{C_t, L_t, K_{t+1}\}$ to maximise their discounted utility on an infinite horizon; they are the owners of the labour and capital in the economy. Their budget constraint also contains profits of the intermediary firms. The capital of the previous period is subject to depreciation and can be augmented with the current period investments:

$$\begin{aligned}
&\max_{\{C_t, L_t, K_{t+1}\}_{t=0}^{\infty}} E_t \sum_{\tau=t}^{\infty} \beta^{\tau-t} u(C_\tau, 1-L_\tau) \\
&\text{s.t. } P_t(C_t + K_{t+1} - (1-\delta)K_t) = P_t(w_t L_t + r_t^k K_t) + \int_0^1 \Pi_{j,t} dj \\
&C_t + I_t \leq Y_t
\end{aligned}$$

- K_t is the capital at the beginning of the period t with the real return r_t^k; L_t is their labour supply with the real wage w_t; C_t is the consumption of a representative agent; and I_t is the amount of the produced final good that was invested into the next period's capital;

- the agent's utility is an additively separable in time form of the Cobb-Douglas function: $u(C_t, 1-L_t) = \gamma \ln C_t + (1-\gamma)\ln(1-L_t)$, with γ being the normalisation constant;

- β is the discount factor $\beta \in [0;1)$;

- P_t is the price of the final consumption good and Π_t is the profit of a representative firm.

Final good production. The final good is produced by a representative firm in a perfectly competitive market. A representative firm chooses the final good price P_t to maximise its profit using the production technology in a Dixit-Stiglitz aggregation form with the elasticity of the intermediary goods substitution parameter ζ[15]:

$$\max_{Y_{j,t}} \Pi_t = P_t Y_t - \int_0^1 P_{j,t} Y_{j,t} dj$$

$$Y_t = \int_0^1 \left(Y_{j,t}^{\frac{\zeta-1}{\zeta}} dj \right)^{\frac{\zeta}{\zeta-1}}, \text{ where}$$

- P_t and Y_t are the price and the supply of the final good, respectively;
- $P_{j,t}$ and $Y_{j,t}$ are the price of the j^{th} intermediary good and the demand for it, respectively.

Intermediate good production. An intermediate firm j produces the j^{th} intermediate good choosing the labour and capital $\{L_{j,t}, K_{j,t}\}$ in accordance with a standard Cobb-Douglas production function with output elasticity of capital α and total factor productivity A_t subject to technology shocks. The firm j aims to minimise the production costs and chooses the price of the j^{th} intermediate good $P_{j,t}$ to maximise its profit:

$$\min_{L_{j,t},K_{j,t}} Cost = W_t L_{j,t} + R_t K_{j,t}$$

$$\max_{P_{j,t}} \Pi_{j,t} = P_{j,t} Y_{j,t} - W_t L_{j,t} R_t K_{j,t}$$

$$\text{s.t } Y_{j,t} = A_t K_{j,t}^{\alpha} L_{j,t}^{(1-\alpha)}$$

$$\ln A_t = (1-\rho_A)\bar{A} + \rho_A \ln A_{t-1} + \sigma_A \varepsilon_t^A, \text{ where}$$

[15] With $\zeta \to \infty$, the problem goes to the basic Neo-Classic dynamic model formulation with perfect competition and no intermediate firms.

- R_t^k and W_t are the nominal return on capital and the nominal wage, respectively;
- ρ_A and σ_A are the persistence and the standard deviation of the total factor productivity shock; $\bar{A}^{(1-\rho_A)}$ is the determined value of the total factor productivity; and $\varepsilon_t^A \sim N(0,1)$ is the exogenous shock of the total factor productivity logarithm.

Equilibrium conditions. The monopolistic competitive equilibrium in this economy can be defined by the set of equations.

- Intertemporal consumption substitution: $1-\mu_t = \beta \mathbb{E}_t \frac{C_t}{C_{t+1}}(1+r_{t+1}^k - \delta)$, where μ_t is a normalised Lagrange multiplier corresponding to the constraint $C_t - I_t \leq Y_t$.
- Consumption-labour substitution: $w_t = \frac{1-\gamma}{\gamma} \cdot \frac{C_t}{1-L_t}$.
- Marginal real cost of capital: $r_t^k = \alpha \frac{\zeta - 1}{\zeta} \cdot \frac{Y_t}{K_t}$.
- Marginal real cost of labour: $w_t = (1-\alpha)\frac{\zeta-1}{\zeta} \cdot \frac{Y_t}{L_t}$.
- Production function for the intermediate good: $Y_t = A_t K_t^\alpha L_t^{(1-\alpha)}$.
- The capital transition from the household problem: $K_{t+1} = (1-\delta)K_t + I_t$.
- Economy budget constraint: $Y_t = C_t + I_t$.
- Transition function for the TFP: $\ln A_t = (1-\rho_A)\bar{A} + \rho_A \ln A_{t-1} + \varepsilon_t^A$.

Neural network parametrisation. In this model, the state variables and the control variables consist of:

$$\mathbb{S}_t = \{K_t\}\,;\ \mathbb{M}_t = \{A_t\}\,; \mathbb{X}_t = \left\{L_t, Y_t, C_t, K_{t+1}, w_t, r_t^k\right\}.$$

The only component of the shock vector is the TFP shock $\varepsilon = \{\varepsilon^A\}$.

The set of the estimated model parameters includes only arguments for the total factor productivity shock, while other parameters are calibrated according to the last column of Table 2.

$$\bar{\Theta} = \left\{\beta, \gamma, \alpha, \delta, \zeta, \bar{A}\right\}; \; \tilde{\Theta} = \left\{\rho_A, \sigma_A\right\}.$$

The capital endowment for the period t is drawn from the uniform distribution $K_t \sim Uniform[10^{-8}; 50]$; the TFP shock of the period t is drawn from the ergodic normal distribution $\varepsilon_t^A \sim N(0; \frac{\sigma_A}{\sqrt{1-\rho_A}})$ so that the initial value of the TFP follows $A_t = \exp(\varepsilon_t^A) \cdot \bar{A}^{(1-\rho_A)}$; and the TFP shocks for the AiO expectation parameter are drawn from the normal distribution $\varepsilon_1^A, \varepsilon_2^A \sim N(0; \sigma_A)$. We do not assume any particular form of the $\tilde{\Theta}$ parameter distribution; therefore, we generate these parameters for each draw from the uniform distribution with the bounds presented in the third and fourth columns of Table 2.

Table 2: Estimated parameters of a simple New Keynesian model with monopolistic competition

Parameter	Parameter name	Lower bound	Upper bound	Calibrated value
β	Discounting factor	-	-	0.99
γ	Normalisation constant in the utility function	-	-	0.4
α	Output elasticity of capital in the production function	-	-	0.35
δ	Depreciation rate	-	-	0.025
ζ	Elasticity of the intermediary good substitution	-	-	2.0
$\bar{A}$	Determined component of the total factor productivity shock	-	-	1.0
ρ_A	Persistence of the total factor productivity shock	0.9	0.99	0.95
σ_A	Std. of the total factor productivity shock	0.005	0.015	0.01

The neural network approximates the decision on the household's labour supply L_t, which is parametrised to lie within 0 and 1, and the Euler equation parameter $(1 - \mu_t)$, which is defined to be non-negative:

$$\begin{pmatrix} L_t \\ 1-mu_t \end{pmatrix} = \begin{pmatrix} \frac{1}{1+\exp(-\xi_1(\mathbb{S}_t,\mathbb{M}_t,\varepsilon_1,\varepsilon_2,\Theta))} \\ \exp(\xi_2(\mathbb{S}_t,\mathbb{M}_t,\varepsilon_1,\varepsilon_2,\Theta)) \end{pmatrix},$$

The other control variables $\{Y_t,w_t,r_t^k,C_t,K_{t+1}\}$ are computed according to the model's equilibrium conditions.

We derive the KKT conditions and the Euler equation for this model in the same manner we did in Section 5. These two constraints define the loss function in the following way.

$$\begin{aligned} Loss(\xi,\Theta) &= \mathbb{E}_{\mathbb{S}_t,\mathbb{M}_t,\varepsilon_1,\varepsilon_2}\left[\underbrace{F_1(\mathbb{S}_t,\mathbb{M}_t,\varepsilon_1,\Theta)\,F_1(\mathbb{S}_t,\mathbb{M}_t,\varepsilon_2,\Theta)+v\,(F_2(\mathbb{S}_t,\mathbb{M}_t))^2}_{\varphi_{NN}(\xi,\mathbb{S}_t,\mathbb{M}_t,\varepsilon_1,\varepsilon_2,\Theta)}\right]; \\ F_1(\mathbb{S}_t,\mathbb{M}_t,\varepsilon,\Theta) &= \beta\bar{R}\mathbb{E}_\varepsilon\left[\frac{C_t}{C_{t+1}}\cdot(1+r_t^k-\delta)\right]-(1-\mu_t); \\ F_2(\mathbb{S}_t,\mathbb{M}_t,\Theta) &= F^{FB}\left(1-(1-\mu_t),1-\frac{C_t}{Y_t}\right), \end{aligned}$$

The neural network's architecture, the optimisation techniques for its training, and the procedure of the model parameters and initial state variable generation correspond to that of the consumption-saving problem. The neural network was trained for 20,000 iterations (for the progression of the Euler equation residuals refer to Appendix A.1); simulating the economy for T periods forward to train the network on the ergodic state variable space also did not improve the algorithm convergence.

Results of the model solution. The functions for the neural network approximations φ_{NN} of the control variables $\mathbb{X}_t$, in respect to the period t^{th} period capital endowment K_t are depicted in the Figure 3. It should be noted that consumption grows with the increase of the capital endowment, while the labour supply slumps, as is complacent with the general economic theory. The behaviour of the real factor prices is also predictable - it falls with

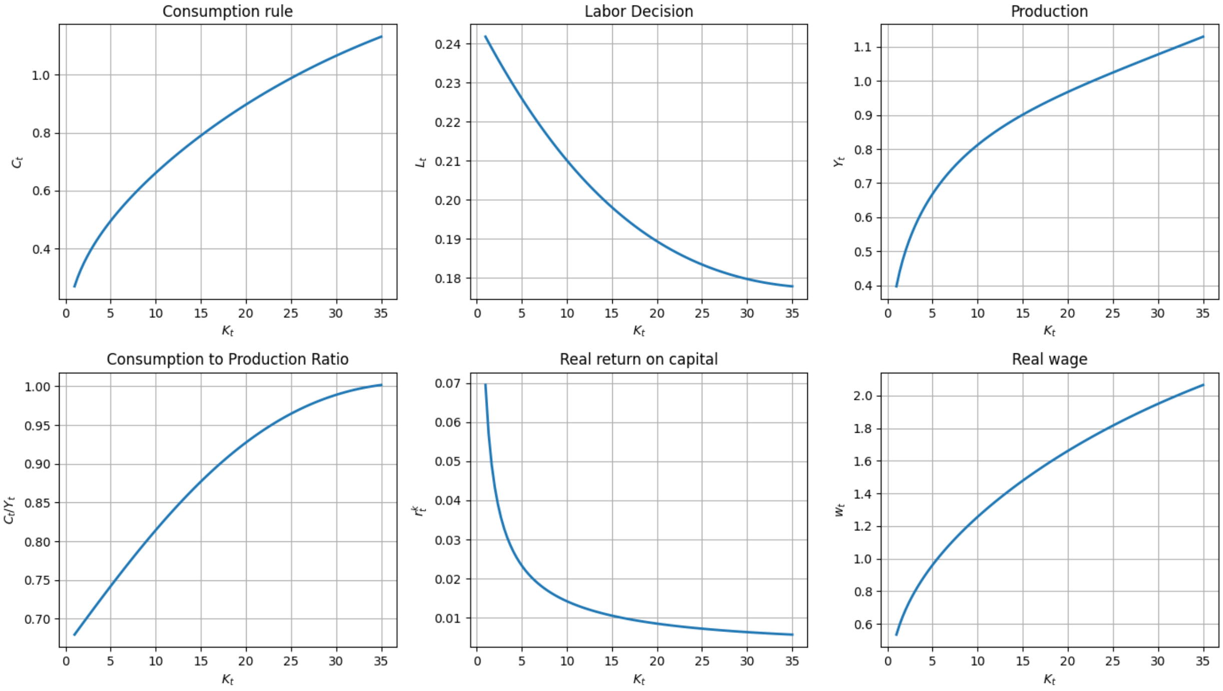

Figure 3: Neural network approximations of the control variable $\mathbb{X}_t \sim \varphi_{NN}$ behaviour in a simple New Keynesian DSGE model with monopolistic competition depending on the initial capital endowment K_t, *ceteris paribus*. The graphs represent the output L_t of the neural network $\xi(\mathbb{S}_{\approx - \nVdash}, \mathbb{M}_{\approx - \nVdash}, \tilde{\Theta})$ and the other control variables, derived through the equilibrium conditions. The TFP shock is set to zero, the estimated and the calibrated parameters are set in accordance with the last column of Table 2.

the greater factor endowment and rises in the presence of the factor scarcity. As the model is simple, there exists only one optimal pair of the labour supply L_t and capital endowment K_t; thus, the values of the real factor prices w_t and r_t^k are also unique.

Nevertheless, it should be noted that consumption reaches the constraint of $C_t = Y_t$ after surpassing the initial endowment $K_t \approx 35$ - the capital in this scenario decreases gradually with each following economy step after the period t until it reaches a certain value around $K_t \approx 6.7$. It should be noted that under the same values of the estimated parameters of the TFP shock, the economy converges to a certain state (for further details see Appendix A.2). This feature can also be traced through the impulse-response functions in Figure 4: following a positive TFP shock, the economy returns to its steady state after a finite amount of periods.

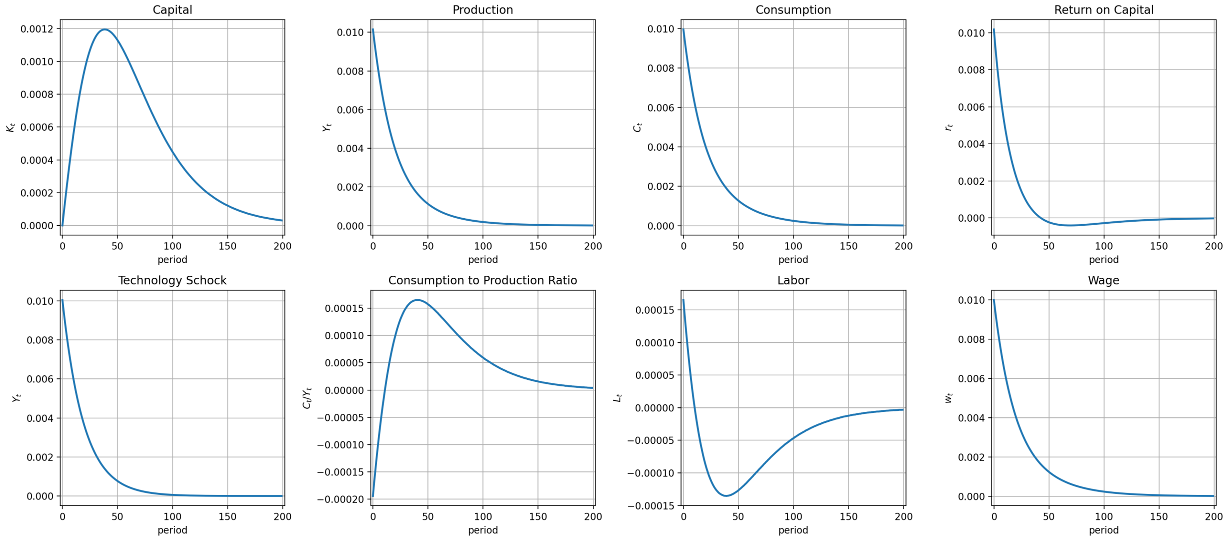

Figure 4: The impulse-response functions (IRFs) for the state variables $\{K_t, A_t\}$ (first column) and control variables $\{L_t, C_t, Y_t, w_t, r_t^k\}$. The IRFs are derived in a manner of Swarbrick (2021) by simulating the 10,000 economies with a randomly distributed capital endowment for 200 periods and subtracting the average trajectory of the economies with an absent shock from the average trajectory of the same economies with a 1 std. shock present in the 1st period. The capital endowments K_t are distributed uniformly $K_t \sim Uniform[10^{-8}; 50]$; the estimated parameter values of the TFP shock are fixed at $\tilde{\Theta} = \{\rho_A, \sigma_A\} = \{0.95, 0.01\}$; other model parameters are calibrated according to the last column of Table 2.

To sum up, the neural network solution to a small-scale representative-agent New Keynesian DSGE model is smooth; the decision rules and the IRFs have a predictable (for this economy type) form that does not hold any inconsistencies. That proves the applicability of our neural network algorithm to the small RA DSGE models.

4.3 Remarks on the neural network framework applicability for the HA economic models

This paper aims to create a neural network framework for non-linear models with heterogeneous agents and aggregate risks. Our next step is to test the algorithm's performance on this type of framework. To compare our approach with existing studies, we applied it to two models: the L. Maliar, S. Maliar, and Winant (2021) version of the Krusell and Smith (1998) model with agents differing only in productivity and capital, and the Kase et al. (2022) HANK model, which includes heterogeneous agents with varying preferences, productivity, and employment, but with the same fundamental parameters and aggregate policy variables such as gross tax and wages.

Our neural network model solution framework has shown the following results. Firstly, the solutions for the HANK models were not reproducible. While in the simple consumption-saving and RANK models the neural network solution stabilised at a particular form only after 5,000 iterations and recreated it with various neural network weights initialisations, the solution of the HANK model did not produce a constant shape even after 100,000 iterations. Secondly, unstable solutions provided counterintuitive economic results in a simulation exercise. After a few iterations, the real return on investment and inflation increased tenfold. At the same time, aggregate consumption, labour demand, and consequently, aggregate production decreased to zero, while individual bond holdings converged to a constant value for all households. It was a recurring result after the neural network and the HA model reformulations, which signals that the neural network framework in its current specification is not suitable for solving the considered type of problem.

According to the Machine Learning literature concerning the first problem, the surface of our neural network training loss function held considerable local extrema due to non-linearities in individual and aggregate decision functions, and the stochastic gradient descent algorithm of the neural network training converged to suboptimal values. Common treatments for this problem include: using more iterations for the neural network training; initialising the neural network weights with different values and comparing the solutions; using several optimisers and learning rates for the neural network weights update; changing the neural network architecture; adjusting the training loss function. Training the neural network for more iterations did not improve the result as even complex HANK models are considered relatively easy compared to the deep-learning field high-dimensional problems in computer vision, sound detection, natural language processing, etc. Initialising the neural network weights with various values prevented us from identifying the best solution because the outcomes varied in each training session. Change of the used optimiser[16] resulted in the augmented computation time, but did not affect the solution accuracy.[17] Changing the model architecture[18] affected the smoothness of the solution and did not yield any results on its stability. Adjusting the loss function of the neural network proved to be the most efficient way of affecting the solution convergence, on which we elaborate further.

The second problem is related to the aggregate uncertainty effect on the market equilibrium. The combination of idiosyncratic shocks and individual decisions converged the economy to a zero-production state. To solve this issue, it might be helpful to include market clearing conditions in the neural network's loss function and adjust the weights of equilibrium conditions. Another solution to the second problem might be using Bellman instead of Euler equation residuals in the neural network loss function, as this formulation compares not only two consecutive periods but a T period progression of the agent's discounted utility.

Adding to the previous remarks, the AiO expectation parameter, which allows us to approximate the integrals over the state space, depends extensively on the defined ergodic area. The state uncertainty has increased by introducing the model parameters as economic pseudo-variables, resulting in a larger and irregularly-shaped model state. Even neural networks cannot converge on the correct solution in a vast domain, but this feature is augmented by the incompatibility of some of the parameter (and state variable) combinations: in case of the incompatible combination, the problem simply does not have a solution, but the neural network adapts the weights in the fully-connected layers in an attempt to solve is, which renders the training ineffective. In this case, a criterion has to be developed to sort the solvable model parameter combinations; it would allow training

[16] We attempted to use the standard Stochastic Gradient Descent, RMSprop, Adam, Adagrad, and AdamW from the PyTorch library.

[17] It should be noted that the average of squared Euler residuals of the intertemporal individual agent problem stabilised at 10^{-5} for the best solutions and at 10^{-2} for the worst and descended gradually for each training session.

[18] We have adjusted the number of the fully-connected layers, experimented with the activation functions, and attempted to switch fully-connected layers to the convolutional ones (Yamashita et al. 2018).

the neural network only in the relevant space. Considering the incompatible state variable values, the neural network framework needs a more strict definition of the ergodic state. A possible implementation of which might be adaptive learning space as in Renner and Scheidegger (2018).

5 Conclusion

This paper examines the current use of Machine Learning for solving and estimating high-dimensional models. Since parameter estimation methods are typically similar for both representative-agent and heterogeneous-agent frameworks, our research mainly focuses on defining the algorithm for solving DSGE models and evaluating its performance on several model types. Based on the recent advances in the field, we formulate a neural network framework for the DSGE model solution. Our framework is learning the agents' optimal decision rules in the evolving economy as we reformulate the problem following the Euler equation method. The proposed algorithm also significantly reduces the time needed for parameter estimation as the model only requires a single solution.

Next, we estimate the performance of the neural network framework on a simple consumption-saving model, a New Keynesian model with monopolistic competition, and then evaluate our algorithm using two HANK models in the style of Krusell and Smith (1998). The proposed framework's application step has demonstrated that it can generate reliable and understandable economic results for small-scale representative-agent models. However, it may not consistently produce accurate results when approximating solutions for complex heterogeneous-agent models.

Concluding, our neural network framework in its current state is not suitable for the solution of the large complex models, and additional research is needed to achieve more accurate results in future. We particularly stress the need to analyse the approximation techniques of agents' expectations through simulation algorithms, to examine the explicit definition of the ergodic space for the economic state variables, and to conduct the experiments with problem formulation formats to further enhance our approach.

6 Bibliography

Algan, Yann, Olivier Allais, Wouter Den Haan, and Pontus Rendahl (2010). “Solving and Simulating Models with Heterogeneous Agents and Aggregate Uncertainty”. In: *Sciences Po publications*. Number: info:hdl:2441/5lc99v0teo898rban7bgs9reua Publisher: Sciences Po.

Alves, Felipe, Christian Bustamante, Xing Guo, Katya Kartashova, Soyoung Lee, Thomas Michael Pugh, Kurt See, Yaz Terajima, and Alexander Ueberfeldt (2022). “Heterogeneity and Monetary Policy: A Thematic Review”. In: *Discussion Papers*. Number: 2022-2 Publisher: Bank of Canada.

Auclert, Adrien (2017). *Monetary Policy and the Redistribution Channel*.

Azinovic, Marlon, Luca Gaegauf, and Simon Scheidegger (2019). *Deep Equilibrium Nets*. Rochester, NY.

Bayer, Christian and Ralph Luetticke (2018). *Solving Heterogeneous Agent Models in Discrete Time with Many Idiosyncratic States by Perturbation Methods*. Rochester, NY.

Bellman, Richard (1958). “Dynamic programming and stochastic control processes”. In: *Information and Control* 1.3, pp. 228–239.

Brumm, Johannes, Christopher Krause, Andreas Schaab, and Simon Scheidegger (2021). *Sparse Grids for Dynamic Economic Models*. Rochester, NY.

Brumm, Johannes and Simon Scheidegger (2017). *Using Adaptive Sparse Grids to Solve High-Dimensional Dynamic Models*. Rochester, NY.

Challe, Edouard, Julien Matheron, Xavier Ragot, and Juan F. Rubio-Ramirez (2017). “Precautionary saving and aggregate demand”. In: *Quantitative Economics* 8.2. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.3982/QE714, pp. 435–478.

Den Haan, Wouter J. (1996). “Heterogeneity, Aggregate Uncertainty, and the Short-Term Interest Rate”. In: *Journal of Business & Economic Statistics* 14.4. Publisher: [American Statistical Association, Taylor & Francis, Ltd.], pp. 399–411.

Dixit, Avinash K. and Joseph E. Stiglitz (1977). “Monopolistic Competition and Optimum Product Diversity”. In: *The American Economic Review* 67.3. Publisher: American Economic Association, pp. 297–308.

Dou, Winston Wei, Xiang Fang, Andrew W. Lo, and Harald Uhlig (2022). *Macro-Finance Models with Nonlinear Dynamics*. Rochester, NY.

Dou, Winston Wei, Andrew W. Lo, Ameya Muley, and Harald Uhlig (2020). *Macroeconomic Models for Monetary Policy: A Critical Review from a Finance Perspective*. Rochester, NY.

Duarte, Victor (2018). *Machine Learning for Continuous-Time Economics*. Rochester, NY.

Duffy, John and Paul D McNelis (2001). “Approximating and simulating the stochastic growth model: Parameterized expectations, neural networks, and the genetic algorithm”. In: *Journal of Economic Dynamics and Control* 25.9, pp. 1273–1303.

Elfwing, Stefan, Eiji Uchibe, and Kenji Doya (2017). *Sigmoid-Weighted Linear Units for Neural Network Function Approximation in Reinforcement Learning*. arXiv: 1702.03118[cs].

Faraglia, Elisa, Albert Marcet, Rigas Oikonomou, and Andrew Scott (2019). “Government Debt Management: The Long and the Short of It”. In: *Review of Economic Studies* 86.6. Publisher: Oxford University Press, pp. 2554–2604.

Fernández-Villaverde, J., S. Hurtado, and G. Nuño (2019). *Financial Frictions and the Wealth Distribution*.

Fernández-Villaverde, J., J. F. Rubio-Ramírez, and F. Schorfheide (2016). “Chapter 9 - Solution and Estimation Methods for DSGE Models”. In: *Handbook of Macroeconomics*. Ed. by John B. Taylor and Harald Uhlig. Vol. 2. Elsevier, pp. 527–724.

Folini, Doris, Felix Kubler, Aleksandra Malova, and Simon Scheidegger (2021). *The Climate in Climate Economics*. Rochester, NY.

Gali, Jordi, J. David Lopez-Salido, and Javier Valles (2004). *Rule-of-Thumb Consumers and the Design of Interest Rate Rules*.

Google Colaboratory (2023). URL: https://colab.research.google.com/.

Gorodnichenko, Yuriy, Lilia Maliar, Serguei Maliar, and Christopher Naubert (2021). “Household Savings and Monetary Policy under Individual and Aggregate Stochastic Volatility”. In: *CEPR Discussion Papers*. Number: 15614 Publisher: C.E.P.R. Discussion Papers.

Haan, Wouter J. den and Albert Marcet (1990). “Solving the Stochastic Growth Model by Parameterizing Expectations”. In: *Journal of Business & Economic Statistics* 8.1. Publisher: [American Statistical Association, Taylor & Francis, Ltd.], pp. 31–34.

Han, Jiequn, Yucheng Yang, and Weinan E (2021). *DeepHAM: A Global Solution Method for Heterogeneous Agent Models with Aggregate Shocks*. Rochester, NY.

Hill, Edward, Marco Bardoscia, and Arthur Turrell (2021). *Solving Heterogeneous General Equilibrium Economic Models with Deep Reinforcement Learning*. arXiv: `2103.16977[cs,econ,q-fin,stat]`.

Judd, Kenneth (1992). “Projection methods for solving aggregate growth models”. In: *Journal of Economic Theory* 58.2. Publisher: Elsevier, pp. 410–452.

Judd, Kenneth, Lilia Maliar, and Serguei Maliar (2009). *Numerically Stable Stochastic Simulation Approaches for Solving Dynamic Economic Models*.

Judd, Kenneth L. and Sy-Ming Guu (1993). “Perturbation Solution Methods for Economic Growth Models”. In: *Economic and Financial Modeling with Mathematica®*. Ed. by Hal R. Varian. New York, NY: Springer, pp. 80–103.

Kase, Hanno, Leonardo Melosi, and Matthias Rottner (2022). *Estimating Nonlinear Heterogeneous Agents Models with Neural Networks*. Rochester, NY.

Kingma, Diederik P. and Jimmy Ba (2017). *Adam: A Method for Stochastic Optimization*. arXiv: `1412.6980[cs]`.

Kollmann, Robert, Serguei Maliar, Benjamin A. Malin, and Paul Pichler (2011). “Comparison of solutions to the multi-country Real Business Cycle model”. In: *Journal of Economic Dynamics and Control* 35.2. Publisher: Elsevier, pp. 186–202.

Krueger, D. and F. Kubler (2004). “Computing equilibrium in OLG models with stochastic production”. In: *Journal of Economic Dynamics and Control* 28.7. Publisher: Elsevier, pp. 1411–1436.

Krusell, Per and Jr. Smith Anthony A. (1998). “Income and Wealth Heterogeneity in the Macroeconomy”. In: *Journal of Political Economy* 106.5. Publisher: The University of Chicago Press, pp. 867–896.

Lepetyuk, Vadym, Lilia Maliar, and Serguei Maliar (2020). “When the U.S. catches a cold, Canada sneezes: A lower-bound tale told by deep learning”. In: *Journal of Economic Dynamics and Control* 117, p. 103926.

Malakhovskaya, Oxana (2021). *"Business Cycles"*. Undergraduate course, Higher School of Ecoonmics, Moscow.

Maliar, Lilia and Serguei Maliar (2014). “Chapter 7 - Numerical Methods for Large-Scale Dynamic Economic Models”. In: *Handbook of Computational Economics*. Ed. by Karl Schmedders and Kenneth L. Judd. Vol. 3. Handbook of Computational Economics Vol. 3. Elsevier, pp. 325–477.

— (2022). “Deep learning classification: Modeling discrete labor choice”. In: *Journal of Economic Dynamics and Control* 135, p. 104295.

— (2020). *Deep Learning: Solving HANC and HANK Models in the Absence of Krusell-Smith Aggregation*. Rochester, NY.

Maliar, Lilia, Serguei Maliar, and Pablo Winant (2021). “Deep learning for solving dynamic economic models.” In: *Journal of Monetary Economics* 122, pp. 76–101.

Marcet, Albert and Guido Lorenzoni (1998). *Parameterized expectations approach; Some practical issues*. Economics Working Paper. Department of Economics and Business, Universitat Pompeu Fabra.

Paszke, Adam, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala (2019). “PyTorch: An Imperative Style, High-Performance Deep Learning Library”. In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox, and R. Garnett. Curran Associates, Inc., pp. 8024–8035.

Ragot, Xavier (2019). “Managing Inequality over the Business Cycles: Optimal Policies with Heterogeneous Agents and Aggregate Shocks”. In: *2019 Meeting Papers*. Number: 1090 Publisher: Society for Economic Dynamics.

Renner, Philipp and Simon Scheidegger (2018). *Machine Learning for Dynamic Incentive Problems*. Rochester, NY.

Santos, Manuel S. (2000). “Accuracy of Numerical Solutions Using the Euler Equation Residuals”. In: *Econometrica* 68.6. Publisher: [Wiley, Econometric Society], pp. 1377–1402.

Scheidegger, Simon and Ilias Bilionis (2017). *Machine Learning for High-Dimensional Dynamic Stochastic Economies*. Rochester, NY.

Swarbrick, Jonathan (2021). *Occasionally Binding Constraints in Large Models: A Review of Solution Methods*. Number: 2021-5 Publisher: Bank of Canada.

Valaitis, Vytautas and Alessandro T. Villa (2021). *A Machine Learning Projection Method for Macro-Finance Models*. Rochester, NY.

Winberry, Thomas (2018). “A method for solving and estimating heterogeneous agent macro models”. In: *Quantitative Economics* 9.3, pp. 1123–1151.

Yamashita, Rikiya, Mizuho Nishio, Richard Kinh Gian Do, and Kaori Togashi (2018). “Convolutional neural networks: an overview and application in radiology”. In: *Insights Imaging* 9.4. Number: 4 Publisher: SpringerOpen, pp. 611–629.

A Additional materials for a simple Representative Agent New Keynesian DSGE model with monopolistic competition

A.1 Euler residuals of a simple RANK DSGE model

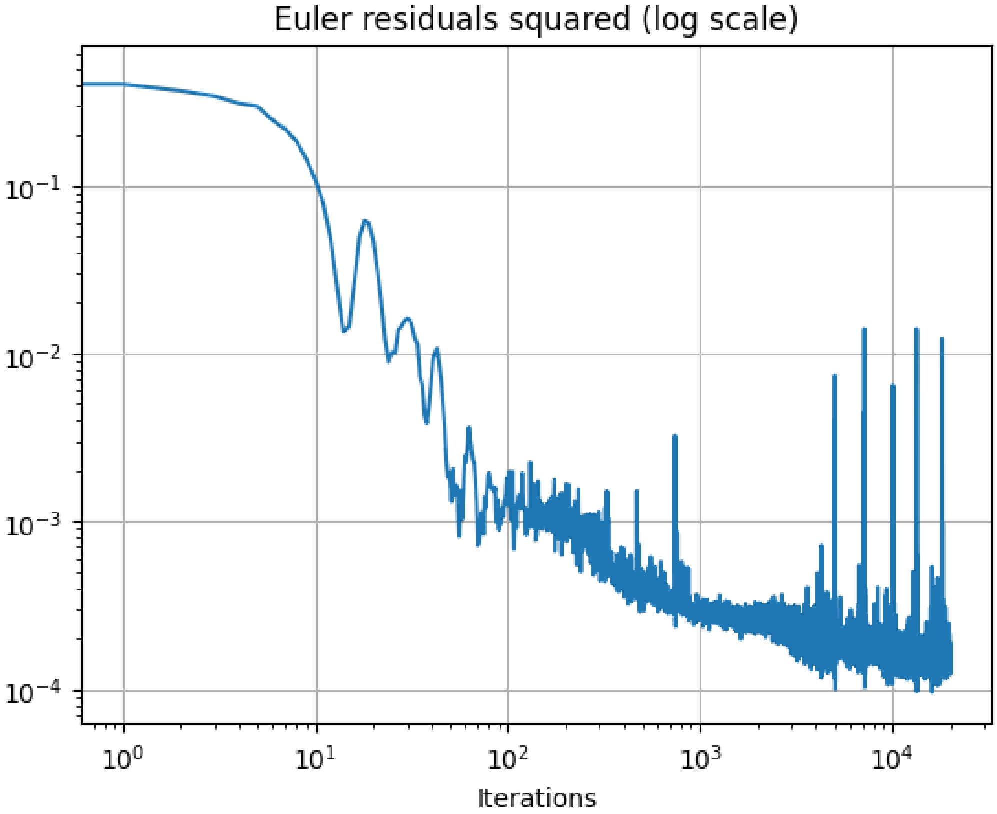

Figure 5: Progression of the residuals of the Euler equation during the neural network training for the New Keynesian DSGE model with monopolistic competition, computed using the standard method as in Santos (2000), without the AiO (see Section 3.5) expectation parameter.

A.2 Simulation of the economies in a simple RANK DSGE model

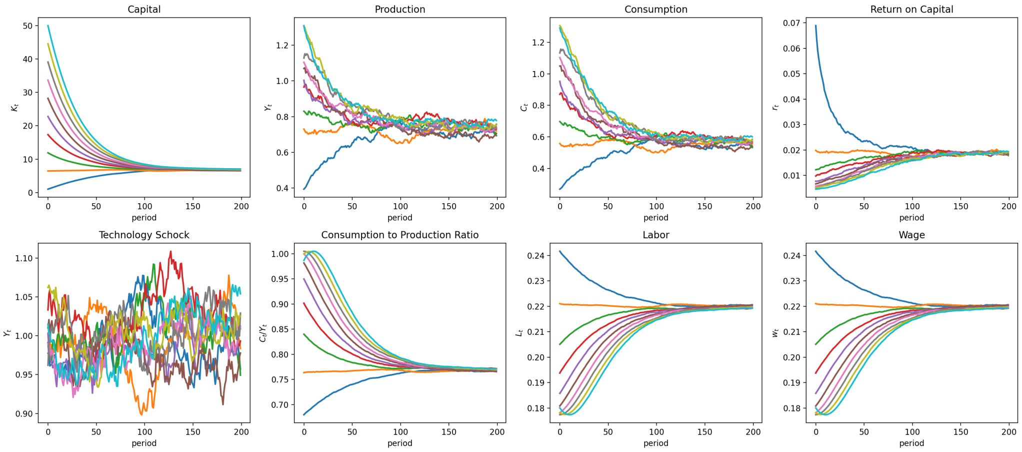

Figure 6: Simulation of the economies in the New Keynesian DSGE model with monopolistic competition solved in the neural network framework. The capital endowments K_t are distributed uniformly $K_t \sim Uniform[10^{-8};50]$; the estimated parameter values of the TFP shock are fixed at $\tilde{\Theta} = \{\rho_A, \sigma_A\} = \{0.95, 0.01\}$; other model parameters are calibrated according to the last column of Table 2. The first column presents the state variables, $\{K_t, A_t\}$; the control variables $\{L_t, C_t, Y_t, w_t, r_t^k\}$ fill the rest of the graph.

A.3 Decision rule elasticity on the estimated model parameter values in a simple RANK DSGE model

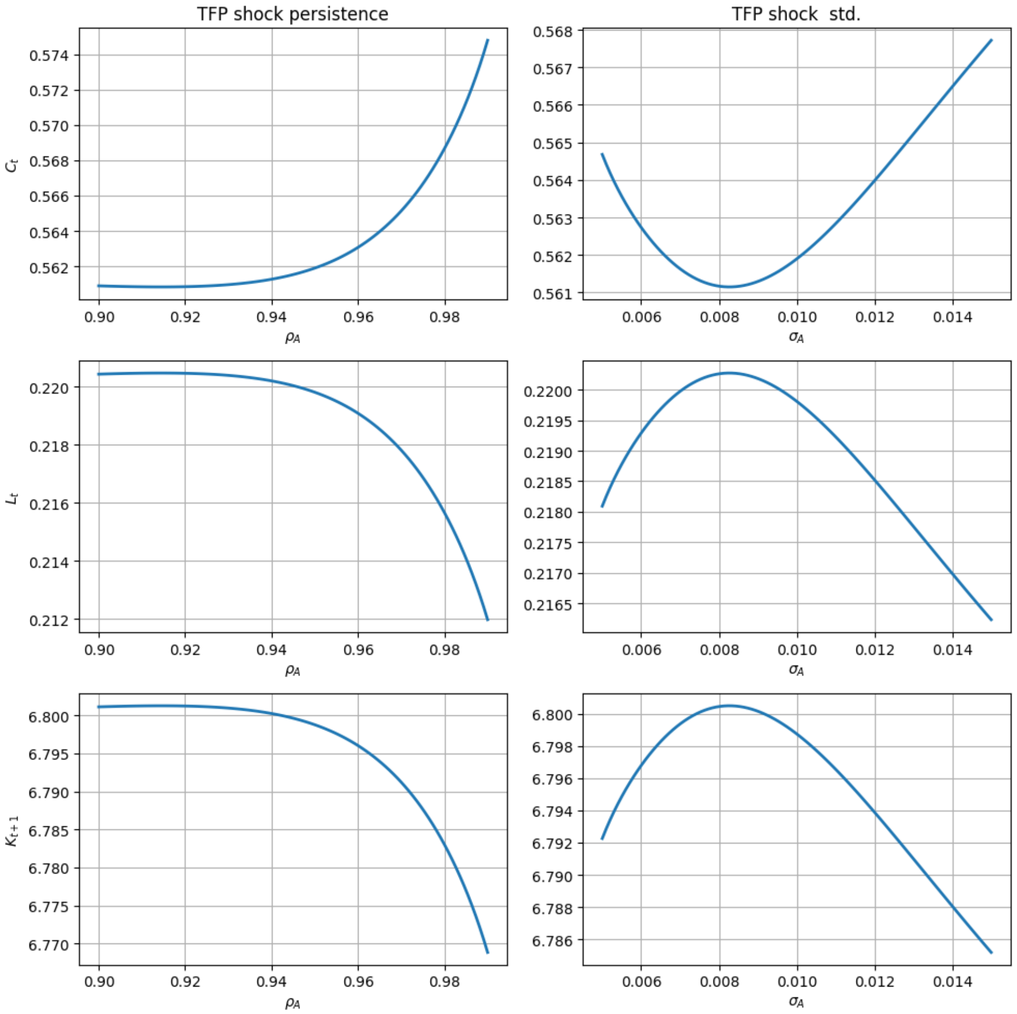

Figure 7: The graph illustrates changes in the decision function for consumption C_t, household labour supply L_t, and the next period capital K_{t+1} in a small-scale New Keynesian DSGE model based on the structural parameter value, while the remaining parameters are determined by the calibrated value (see the last column of Table 2); all parameters vary within a range from their minimum to maximum values. The initial capital endowment is fixed at $K_t = 6.8$ (all economies converge around this value), and exogenous shocks ε are set to zero.