BANK OF CANADA
BANQUE DU CANADA

Staff Working Paper/Document de travail du personnel—2025-3

Last updated: January 16, 2025

Differentiable, Filter Free Bayesian Estimation of DSGE Models Using Mixture Density Networks

by Christopher Naubert

Canadian Economic Analysis Department
Bank of Canada
CNaubert@bankofcanada.ca

DOI: https://doi.org/10.34989/swp-2025-3 | **ISSN** 1701-9397

Abstract

I develop a methodology for Bayesian estimation of globally solved, non-linear macroeconomic models. A novel feature of my method is the use of a mixture density network to approximate the distribution of initial states. I use the methodology to estimate a medium-scale, two-agent New Keynesian model with irreversible investment and a zero lower bound on nominal interest rates. Using simulated data, I show that the method is able to recover the "true" parameters when using the mixture density network approximation of the initial state distribution. This contrasts with the case when the initial states are set to their steady-state values.

Topics: Business fluctuations and cycles; Economic models
JEL codes: C61, C63, E37, E47

Résumé

J'élabore une méthode d'estimation bayésienne pour les modèles macroéconomiques non linéaires résolus globalement. Cette méthode innove notamment par l'utilisation d'un réseau de densité de mélange comme approximation de la distribution des états initiaux. Je l'applique pour estimer un modèle néokeynésien à moyenne échelle et à deux agents en situation d'investissement irréversible où les taux d'intérêt nominaux sont soumis à la borne du zéro. À l'aide de données simulées, je montre que cette méthode permet de récupérer les « vrais » paramètres lorsqu'on utilise le réseau de densité de mélange comme approximation de la distribution des états initiaux. Cette situation contraste avec celle où les états initiaux sont paramétrés à leurs valeurs de régime permanent.

Sujets : Cycles et fluctuations économiques; Modèles économiques
Codes JEL : C61, C63, E37, E47

1 Introduction

Over the past two decades, the zero lower bound constraint has become an important limitation on central bank policy responses around the world. In more recent years, central banks have been faced with the challenge of choosing a path for raising the nominal interest rate away from the zero lower bound. Many models used by central banks, however, rely on first-order approximations of agent's optimal policies around the non-stochastic steady-state. Standard first-order approximation methods cannot handle occasionally binding constraints such as the zero lower bound. Additionally, because the first-order approximate policies are linear, one cannot use these models to assess the state- and scale-dependent effects of central bank policy. Therefore, these methods are not well suited to compare the effect of a 25 basis point increase versus a 50 basis point increase in the policy rate, for example.

Given that one needs global, non-linear solutions to address many of the most pressing questions central bankers face, I develop a novel methodology for Bayesian estimation of globally solved, non-linear macroeconomic models. My methodology addresses three of the difficulties one typically faces when estimating non-linear models using Markov chain Monte Carlo. First, my method generates a differentiable target density allowing one to use Hamiltonian Monte Carlo (HMC) to generate samples. HMC tends to result in lower correlation between samples, meaning one needs fewer samples to reach a target effective sample size. Second, I avoid repeatedly solving for policies during the estimation step by solving for the policies as functions of the states and economic parameters as in Kase, Melosi, and Rottner (2022). Finally, I avoid non-linear filtering by generating samples for the joint density of the parameters, innovations to the exogenous states, and initial states conditional on the data as in Childers et al. (2022). One then uses the portion of the samples pertaining to the parameters for posterior inference.

Evaluating the target density requires evaluating the density of the initial states. In linearized models with Gaussian innovations, the initial state density is a multivariate normal density. In non-linear models, one generally cannot express the initial state density in closed form. I circumvent this problem using, to the best of my knowledge, an approach novel to the literature. I first decompose the density into two components: the density of the initial exogenous states conditional on the economic parameters and the density of the initial endogenous states conditional on the exogenous states and economic parameters. In many models, only the second density needs to be approximated as the first can be expressed in closed form.

The key contribution of this paper is the use of mixture density networks (Bishop 1994) to approximate the conditional density of the initial endogenous states. Mixture density networks provide a flexible approximation to a conditional density. A mixture density network uses a neural network to parameterize a mixture distribution. The inputs to the network are the conditioning variables, and the outputs are the parameters of the mixture components. The network is differentiable with respect to its inputs making the approximation

of the initial state density differentiable. The differentiability condition is needed for one to use HMC.

I use the methodology to solve and estimate two models. The first model, the baseline real business cycle model, serves as an illustrative example of how to use the methodology in practice. The second model is a medium-scale, two-agent New Keynesian model with a zero lower bound on the nominal interest rate, irreversible capital investment, sticky prices, sticky wages, and three aggregate shocks. In the main quantitative application, I estimate 17 parameters of the model on simulated data. I show that my method is able to adequately recover the parameters used to generate the simulated data. Moreover, I show that a variant of my method, which assumes that the initial state values are equal to their steady-state values instead of being sampled from the mixture density network, fails to recover any of the parameters used to generate the data. This result emphasizes the importance of the mixture density network in the overall performance of the methodology.

As a secondary contribution, I show how economists can incorporate neural network training strategies from the deep reinforcement learning literature to improve training stability. In particular, I adopt two aspects of the deep deterministic policy gradient algorithm (Lillicrap et al. 2016) that have been shown to improve training stability: using one parameter vector to compute current choices and a second parameter vector to compute future choices and slow updating of the parameter vector used to compute future choices. Incorporating these aspects into the training procedure both improves training stability and reduces the testing error.

Related Literature

This paper contributes to the literature on Bayesian estimation of dynamic stochastic general equilibrium (DSGE) models. Herbst and Schorfheide (2015) provide a textbook treatment of Bayesian estimation. For a survey of recent developments in Bayesian estimation of DSGE models, see Fernández-Villaverde and Guerrón-Quintana (2021). The method I propose is most closely related to Childers et al. (2022) and Kase, Melosi, and Rottner (2022). As in Childers et al. (2022), I use the joint likelihood in the estimation step instead of using a filter. The models estimated in their paper, however, are solved using first- or second-order perturbation methods. In contrast, I construct global, non-linear solutions to the models. Therefore, the methodology presented in this paper can be used to estimate a wider class of models such as those with stochastic volatility and occasionally binding constraints.

Kase, Melosi, and Rottner (2022) also propose a method for estimating globally solved, non-linear models. In Kase, Melosi, and Rottner (2022), the authors approximate the likelihood function with a neural network. They sample a set of economic parameters and use a particle filter to compute the likelihood values on this set. Then, the authors train a neural network on the set of sampled economic parameters and likelihood values. The approximated likelihood is used in the estimation. In their approach, each time one receives new

data, one has to retrain the neural network approximating the likelihood. In models with high dimensional parameter spaces, retraining is computationally expensive due to the large number of samples of parameters one needs to cover a sufficient portion of the parameter space. In my approach, I avoid the use of a particle filter by using the joint likelihood. In turn, I do not need to re-fit any approximating functions when I receive new data.

Outline

The remainder of the paper is as follows. Section 2 provides a generic description of the types of models that can be solved and estimated using the methodology I develop in this paper. Section 3 describes the estimation setup. In Section 4, I explain how I approximate the optimal policy functions. Section 5 describes how I use mixture density networks to approximate the initial state distribution. In Section 6, I demonstrate how to use the methodology by way of two examples. Section 7 concludes. Additional details are included in the Appendix.

2 Economic Model

An economic model is a collection of discounted Markov decision problems. A solution to an economic model is then a collection of optimal policy functions. In economic models of interest, the optimal policy functions cannot be characterized in closed form. Instead, one needs to approximate the optimal policy functions. The optimizing behavior of the agents in the model delivers a set of conditions that the optimal policies satisfy. Solving an economic model, therefore, implies finding a collection of approximate optimal policies that satisfy the set of equilibrium conditions within some degree of numerical tolerance.

The optimal action of an agent depends on the state of the economy. Additionally, the optimal policies depend on the parameters of the economic model. I use $x \in \mathbb{R}^{N_x}$ to denote the current state of the economy and $u \in \mathbb{R}^{N_u}$ to denote the economic parameters. Each agent in an economic model solves a discounted Markov decision problem. I use the subscript i to denote an agent's identity. Agent i's flow payoff function is denoted by $r_i(\cdot,\cdot,\cdot)$. Agent i's policy function is given by $\Gamma_i(\cdot,\cdot)$ with the optimal policy given by $\Gamma_i^*(\cdot,\cdot)$. The collection of actions implied by agent i's policy is given by a_i. For each agent i, the action space is continuous. Denote the set of feasible actions for agent i as $\Phi_i(\cdot,\cdot;\cdot,\cdot)$. Agent i's discount factor is denoted by $\beta_i \in [0,1)$. The state transition function is given by $\mathcal{T}(\cdot,\cdot,\cdot;\cdot,\cdot)$. The economy is subject to innovations, denoted by ε, with distribution G. I denote the conditional expectation operator by $E[\cdot|\cdot]$. The value

function of agent i is denoted as $V_i(\cdot,\cdot)$. The problem of agent i is

$$V_i(x,u) = \max_{\widehat{\Gamma}_i} \{r_i(a_i,x,u) + \beta_i E[V_i(x',u)|x,u]\} \tag{1}$$

$$a_i = \widehat{\Gamma}_i(x,u) \in \Phi_i\left(x,u;\widehat{\Gamma}_i,\left\{\Gamma_j^*\right\}_{j\neq i}\right) \tag{2}$$

$$x' = \mathcal{T}\left(x,u,\varepsilon';\widehat{\Gamma}_i,\left\{\Gamma_j^*\right\}_{j\neq i}\right) \qquad \varepsilon' \sim G \tag{3}$$

To enforce market clearing, I assume that there is an agent called "Nature". Nature has $\beta_{\text{Nature}} = 0$ and chooses pricing functions to maximize a constant value. The set of feasible actions for Nature is restricted to the prices that are consistent with market clearing.

I assume the model has a unique solution and that the optimal policies are characterized by first-order necessary and sufficient conditions. Therefore, in the implementation, instead of solving for value functions, I solve for the policy functions that satisfy the equilibrium conditions.

Given the optimal policies and the state transition functions, estimating the model reduces to estimating a non-linear state space model with unknown parameters. The following section describes the estimation procedure.

3 Estimation

In this paper, I focus on posterior inference using a Bayesian approach. In macroeconomic models, one typically cannot directly sample from the posterior distribution of the parameters. Instead, one uses Markov chain Monte Carlo (MCMC) or sequential Monte Carlo to generate samples from a target density that one knows how to evaluate but not directly sample from. My methodology addresses three of the difficulties one typically faces when estimating non-linear models using Monte Carlo methods. First, to evaluate the target density, one needs to compute the likelihood. With non-linear state transitions, one cannot use the Kalman filter to compute the likelihood. Instead, researchers typically use a particle filter to numerically approximate the likelihood. The numerical approximation, however, tends to be computationally costly. The second difficulty arises because of the fact that the optimal policies depend on the economic parameters. Evaluating the target density requires knowing the policies for the given configuration of the parameters. Traditional estimation approaches re-solve for the optimal policies at each step of the Markov chain. Solving for global, non-linear solutions, however, also tends to be computationally demanding. Finally, the samples generated using MCMC are correlated. Therefore, if one wants N independent samples, one may need to generate $M \gg N$ samples.

Section 3.1 shows how to avoid the filtering step by sampling from the joint likelihood as in Childers et al. (2022). One needs to evaluate the approximate optimal policies and the conditional distribution of initial states to compute the joint likelihood. Section 4 shows how I approximate the optimal policies following the

approach presented in Kase, Melosi, and Rottner (2022). Using their approach, I avoid the need to re-solve for the policies at each step of the Markov chain. In Section 5, I show how I approximate the conditional distribution of initial states. Both the approximate policies and the approximate conditional distribution of initial states are differentiable with respect to the economic parameters. Therefore, one can use HMC to generate samples. HMC tends to generate samples with lower correlation than those generated by random walk Metropolis-Hastings. Therefore, when sampling with HMC, one needs fewer total samples to reach a target effective sample size. In other words, HMC can reduce the computational time needed to generate N independent samples.

3.1 Joint Likelihood

Let $z^T \in \mathbb{R}^{N_z \times T}$ denote the history of observations (the data) up to and including time T, $x^T \in \mathbb{R}^{N_x \times T}$ denote the history of states generated by the economic model, $\varepsilon^T \in \mathbb{R}^{N_\varepsilon \times T}$ denote the series of innovations, and $u \in \mathbb{R}^{N_u}$ denote the economic parameters one is estimating. Denote the conditional density function as $p(\cdot|\cdot)$. The density of the parameters, conditional on the observations, can be decomposed as follows:

$$p\left(u|z^T\right) = \int p\left(u, x^T|z^T\right) dx^T = \int\int p\left(u, \varepsilon^T, x_0|z^T\right) d\varepsilon^T dx_0 \tag{4}$$

One obtains the posterior, $p\left(u|z^T\right)$, by integrating the joint likelihood, $p\left(u, \varepsilon^T, x_0|z^T\right)$, over the space of innovations and initial states. In practice, this means that one can generate samples of the parameters by generating samples from the joint likelihood and discarding the portion of the samples pertaining to the innovations and initial states. The log of the joint likelihood can be further decomposed into $2+2T$ densities and a constant:

$$\ln p\left(u, \varepsilon^T, x_0|z^T\right) = \ln p(u) + \sum_{t=1}^{T} \ln p(\varepsilon_t|u) + \sum_{t=1}^{T} \ln p\left(z_t|\varepsilon^t, x_0, u\right) + \ln p(x_0|u) + C \tag{5}$$

From the decomposition, one can see that all the component terms of the joint likelihood can be computed without the use of a filter. Given a sample of the innovations, a sample of the initial states, and the state transition functions, one can simulate the model forward to recover the hidden states. In the conventional approach, when one does not directly sample the innovations, one needs a filter to recover the hidden states.

The first term in the decomposition, $\ln p(u)$, is the log prior. The researcher specifies this density. Additionally, one can evaluate this density without knowledge of the model solution. The component terms of the first summation, $p(\varepsilon_t|u)$, are the densities of the innovations to the exogenous states at time t conditional on the economic parameters. The researcher also specifies these densities and can evaluate the densities without knowledge of the model solution.

The remaining terms depend on the model solution. The component terms of the second summation, $p(z_t|\varepsilon^t, x_0, u)$, link the model-generated states to the observations. These terms are computable given the

model policy functions and specifications for the measurement equations. Let $z_{i,t}$ denote the observable i (e.g. consumption) at time t and $\Gamma_i^*(\cdot,\cdot)$ denote the true optimal policy function for variable i. Then, if the measurement errors are additive and independently distributed $\mathcal{N}(0,\sigma_{i,me})$, the conditional density for $z_{i,t}$ is

$$p\left(z_{i,t}|\varepsilon^t,x_0,u\right)=\mathcal{N}\left(z_{i,t}|\Gamma_i^*\left(x_t,u\right),\sigma_{i,me}\right) \tag{6}$$

To evaluate the density given in Equation (6), one needs an approximation of the optimal policy function for variable i. Additionally, in order to use HMC, the approximate policy needs to be differentiable with respect to u. I discuss how I approximate the policies in Section 4.1.

In most cases, there will not be an analytical expression for the fourth term, $\ln p\left(x_0|u\right)$. This term is the density of the initial state conditional on the parameters. Note that this term can be further decomposed into the density of the initial exogenous states, x_0^{exo}, conditional on the parameters and the density of the initial endogenous states, x_0^{endo}, conditional on the exogenous states and parameters:

$$\ln p\left(x_0|u\right)=\ln p\left(x_0^{endo}|x_0^{exo},u\right)+\ln p\left(x_0^{exo}|u\right) \tag{7}$$

I assume that the initial states are drawn from the stationary distribution. Given this assumption, the benefit of the decomposition is that the stationary distribution of the exogenous states conditional on the parameters, $p\left(x_0^{exo}|u\right)$, has a closed form in many models. Therefore, one reduces the dimensionality of the distribution that needs to be approximated. In Section 5, I discuss how I approximate the distribution of the initial endogenous states conditional on the exogenous states and parameters.

4 Policy Functions

In this section, I first present the policy function approximation objective. I then provide additional details on the implementation.

4.1 Approximating Policy Functions

To estimate the model using the HMC sampler, the model solution needs to be differentiable with respect to the estimated economic parameters. As in Kase, Melosi, and Rottner (2022), I treat the estimated economic parameters as pseudo-state variables. Let $x\in\mathbb{R}^{N_x}$ denote the current state, $u\in\mathbb{R}^{N_u}$ denote the current pseudo-state, and $\Gamma_i\left(x,u;\gamma_i\right)$ denote the parameterized policy for control $i\in\mathcal{M}$ with parameter vector $\gamma_i\in\mathbb{R}^{N_{\Gamma_i}}$.

I solve for the policies using projection methods. For a textbook treatment of projection methods, see Judd (1998), Chapter 11. Denote the true optimal policy function as $\Gamma_i^*\left(x,u\right)$ and the set of optimal polices

as $\{\Gamma_i^*(x,u)\}$. Additionally, denote the mapping of policies to equilibrium condition residuals by $\mathfrak{R}[\cdot]$. By definition of an optimal policy,

$$\mathfrak{R}[\{\Gamma_i^*(x,u)\}] = \vec{0}$$

for all x and u. Traditionally, projection methods solve for the policies under a fixed set of economic parameters. The projection method objective is to minimize the norm of the equilibrium condition residuals over a set of states, $\mathsf{X}(u)$. Therefore, the traditional approach to solving a model using projection methods can be formulated as

$$\{\gamma_i(u)\} = \underset{\{\hat{\gamma}_i\}}{\operatorname{argmin}} \int_{\mathsf{X}(u)} ||\mathfrak{R}[\{\Gamma_i(x,u;\hat{\gamma}_i)\}]|| d\mu(x) \tag{8}$$

where $||\cdot||$ denotes a norm. Note that the set of states can depend on u in a non-trivial manner. The reason for this is that regions of the state space that have high density with respect to the stationary distribution under one configuration of the economic parameters may have low density under another configuration. Allowing the set of states to vary with u, therefore, allows one to improve accuracy on the higher-density regions of the stationary distribution with a tradeoff of lower accuracy on the lower-density regions.

The problem as presented above highlights two shortcomings of the traditional approach to projection methods. First, each time one changes the values of the economic parameters, u, one needs to re-solve the minimization problem. Given that one needs to evaluate the policies on many different values of u during estimation, the re-solving step can make estimation practically infeasible. Second, the policies are non-differentiable with respect to the economic parameters. Therefore, one cannot use gradient-based sampling procedures or maximizers during estimation.

The alternative approach used in this paper retains the central idea used in traditional projection methods and follows the approach used in Kase, Melosi, and Rottner (2022). The one modification made to the standard projection method is that, instead of minimizing the equilibrium condition residual function over the set of states, the function is minimized over the set of states and the set of economic parameters (pseudo-states), U. The optimization problem for the alternative approach is

$$\{\gamma_i\} = \underset{\{\hat{\gamma}_i\}}{\operatorname{argmin}} \int_{\mathsf{U}} \int_{\mathsf{X}(u)} ||\mathfrak{R}[\{\Gamma_i(x,u;\hat{\gamma}_i)\}]|| d\mu(x)\, d\nu(u) \tag{9}$$

Note that the set of minimizers in Equation (9) are not functions of u. That is, one solves for one parameter vector for each approximate policy and uses this parameter vector for all values of the states and pseudo-states. Additionally, as long as the approximating function is differentiable with respect to u, then the approximate policies are also differentiable with respect to u. Therefore, using the alternative approach, one can use gradient-based sampling procedures or maximizers during estimation.

In the applications presented in this paper, I approximate the policy functions using multi-layer neural

networks.[1] Many recent papers, including Fernández-Villaverde et al. (2020); Maliar, Maliar, and Winant (2021); Azinovic, Gaegauf, and Scheidegger (2022); and Kase, Melosi, and Rottner (2022), use neural networks to approximate policy functions. As in those papers, I train the neural networks to minimize the errors in the equilibrium conditions (individual optimality conditions and market clearing conditions). I provide additional details of the training procedure in Section 4.3.

Neural networks are particularly well suited for the methodology presented in this paper. First, they can handle high dimensional inputs. Given that the input to the policy function includes both the states and the economic parameters one is estimating, the approximating functions need to be able to handle high dimensional spaces. Second, neural networks can approximate complicated functions. Even if the policies are linear for a given set of parameters, the policies may be non-linear functions of the parameters. Therefore, one needs to use an approximating function that is sufficiently flexible. Third, the estimation requires differentiability of the policies with respect to the economic parameters. There are many modern open source software packages that can compute the derivatives of the output of a neural network with respect to its inputs using automatic differentiation.

4.2 Initializing the Neural Network Parameters

The way in which the neural network parameters are initialized can have a non-trivial effect on the performance of the training procedure. I use data generated by a first-order approximation of the policies derived from a surrogate model to initialize the weights of the neural networks.

The initialization procedure is as follows: I first sample a set of economic parameters. Then, following Villemot (2011), I use first-order perturbation techniques to construct the approximate policies. Using the first-order approximate policies, I simulate the model forward for T_{init} periods. I store the set of economic parameters, simulated states, and simulated controls. I repeat this process N_{init} times to generate the training data set.

Note that the surrogate model used to generate the data for the initialization may be different than the model one is trying to solve and estimate. For example, the true model the researcher is interested in may have occasionally binding constraints. Standard perturbation methods cannot handle these constraints. Since the data generated by the surrogate model is only used to initialize the neural network parameters, however, this is not a problem.

Given the training data set, the initialization procedure takes the form of a standard supervised learning problem. Denote the set of variables that one is approximating policy functions for as y and the value of variable i at time t as $y_{i,t}$ with $i \in \mathcal{M}$. For concreteness, assume that one is using a training batch size of N_{batch}. At each training step, I draw a set of batch indices, $\mathcal{J}$, with generic element $j \in \{1, ..., N_{init}\}$. In

1. In the application, I use `Flux.jl` to train the neural networks (Innes et al. 2018).

other words, the js denote which set of economic parameters I use in the training input. For each index in the batch, I draw a time step $t_j \in \{1, ..., T_{init}\}$. Therefore, the inputs to the neural networks are $\left\{\left(x_{t_j,j}, u_j\right)\right\}_{j\in\mathcal{J}}$ and the targets are $\left\{y_{t_j,j}\right\}_{j\in\mathcal{J}}$. Using the mean squared error objective, the optimization problem is

$$\min_{\{\hat{\gamma}_i\}} \frac{1}{N_{batch}} \sum_{j\in\mathcal{J}} \frac{1}{|\mathcal{M}|} \sum_{i\in\mathcal{M}} \left(y_{i,t_j,j} - \Gamma_i\left(x_{t_j,j}, u_j; \hat{\gamma}_i\right)\right)^2 \tag{10}$$

After computing the objective on the current batch, I update the neural network parameters, draw a new batch of indices, and repeat the process. I update the neural network parameters using a standard updating scheme such as ADAM.

4.3 Deep Deterministic Policy Gradient

Neural networks can be unstable. I adopt two approaches deep reinforcement learning researchers use to improve the stability of the training procedure. Specifically, the approach I use is based on the deep deterministic policy gradient (DDPG) algorithm (Lillicrap et al. 2016). First, within the residual function, I use one set of neural network parameters to compute choices in the current period and another set of neural network parameters to compute choices for the future period. Denote the parameters of the network used to compute current choices as $\gamma^{\mathbb{C}}$ and the parameters of the network used to compute future choices as $\gamma^{\mathbb{F}}$. In the asset Euler equations, for example, this means that I compute current consumption, c, and future consumption, c', as

$$c = \Gamma_c\left(x, u; \gamma^{\mathbb{C}}\right) \tag{11}$$

$$c' = \Gamma_c\left(x', u; \gamma^{\mathbb{F}}\right) \tag{12}$$

At each training step, I only take the gradient of the residual function with respect to the parameters of the neural network for current choices, $\gamma^{\mathbb{C}}$. I update $\gamma^{\mathbb{C}}$ using a standard updating scheme such as ADAM.

The second approach I adopt from DDPG is the slow updating of the network parameters for future choices. I form the iteration j network parameters for the future choices network as a convex combination of the network parameters for the future choices network at iteration $j-1$ and the new network parameters for current choices. The hyperparameter τ^{γ} controls the updating speed. The step j updates are given by

$$\gamma_j^{\mathbb{C}} = \texttt{GradientDescent}\left(\nabla_{\gamma^{\mathbb{C}}}\mathfrak{R}, \gamma_{j-1}^{\mathbb{C}}\right) \tag{13}$$

$$\gamma_j^{\mathbb{F}} = \tau^{\gamma}\gamma_j^{\mathbb{C}} + \left(1-\tau^{\gamma}\right)\gamma_{j-1}^{\mathbb{F}} \tag{14}$$

Note that the approach implies that agents use one policy to compute current choices and a different policy to compute future choices in the training procedure. The theoretical model, however, assumes that the two policies are the same. Therefore, when testing the accuracy of the solution, I use the same parameter

vector $\gamma^{\mathbb{C}}$ for both current and future choices. I assess the accuracy of the policies by simulating the model for T_{test} periods and checking the errors in the equilibrium conditions along the simulation.

Maliar, Maliar, and Winant (2021) and Kase, Melosi, and Rottner (2022) use the same neural network parameters to compute current and future choices in their training procedure. I found that this approach often led to unstable training. Additionally, even when the training was stable, and the training error was sufficiently small, the use of two parameter vectors resulted in smaller errors when simulating the model.

4.4 Generating Training Data

In practice, one faces a tradeoff between policy accuracy in those states that occur more frequently in equilibrium and those that occur less frequently. Without the optimal policies, however, one cannot construct the stationary distribution of states to know which states have high density and which have low density. In models with few state variables, one can construct a product grid around the steady-state and use these points as the training states. There are two drawbacks to this approach. First, as is well known, the number of grid points grows exponentially with the dimensionality of the state space. Second, all points on the grid are treated equally in terms of desired accuracy. With a fixed neural network architecture, this means that one trades off higher accuracy at those points with high density under the stationary distribution for improved accuracy on those points with low density.

Instead of training the policies on a fixed set of points, I select the training points at each training step from a replay buffer (Lin 1992). I initially populate the replay buffer with points randomly selected from a distribution centered at the steady-state. At training step s, I draw a batch of parameters of size N_{batch}. Starting from points near each economy's steady-state, I simulate the model for T_{sim} periods. I then randomly select t_{select} points from the simulation and add the points to the replay buffer. These t_{select} points replace the oldest t_{select} points in the replay buffer using a "first in, first out" strategy.

Since the training points at step s are selected randomly from the replay buffer and most of the points in the replay buffer at step s were generated by a policy function that differs from the policy function being trained at step s, the algorithm is an off-policy algorithm. Therefore, the training points may not come from high-density regions of the stationary distribution generated by the current policy. As training progresses and the policy function converges to the optimal policy, however, the points in the replay buffer become representative of the stationary distribution of the optimal policy.

5 Initial State Density

When estimating the model, one needs to specify a density for the initial states. A natural candidate for the density is the stationary density. If the model is linear, and the innovations are Gaussian, then the

stationary distribution of the state variables is a multivariate normal distribution conditional on the economic parameters. In non-linear models or models with non-Gaussian innovations, this is no longer the case. One could assume that the economy always starts at steady-state. Childers et al. (2022) use this approach, for example. In doing so, one imposes a tight prior on parameter combinations that are consistent with the implied steady-state. In Section 6.2, I show how this approach can greatly distort posterior inference. Another alternative would be to assume the stationary distribution is a joint normal distribution even if the true model is non-linear or has non-Gaussian innovations, as is done in Dynare (Adjemian et al. 2022). The method I propose assumes that the initial states are drawn from the stationary density and approximates the stationary density using a flexible functional form.

5.1 Mixture Density Network Approximation

As noted in Section 3, in many economic models one only needs to approximate the stationary density of endogenous states conditional on the exogenous states and economic parameters, $p\left(x_0^{endo}|x_0^{exo},u\right)$. To approximate $p\left(x_0^{endo}|x_0^{exo},u\right)$, I use a mixture density network (Bishop 1994). A mixture density network combines a neural network with a mixture distribution to approximate a conditional distribution. The neural network takes the conditioning variables in as inputs and outputs the parameters of the component distributions and the mixing weights.

In this paper, I use multivariate normal distributions as the component distributions. Let N_{mix} denote the number of mixture components. Denote the neural network that produces the parameters of the mixture distribution as Λ, the mean vector for component i as ζ_{μ_i}, the covariance matrix for component i as ζ_{Σ_i}, and the mixing weight for component i as ζ_{π_i}. Evaluating the network on (x^{exo},u) with neural network parameters λ generates the set of mean vectors, covariance matrices, and mixing weights.

$$\{\zeta_{\mu_i},\zeta_{\Sigma_i},\zeta_{\pi_i}\}_{i=1}^{N_{mix}} \leftarrow \Lambda\left(x_0^{exo},u;\lambda\right) \tag{15}$$

The approximate density is then given by

$$p\left(x_0^{endo}|x_0^{exo},u\right) = \sum_{i=1}^{N_{mix}} \zeta_{\pi_i}\mathcal{N}\left(x_0^{endo}|\zeta_{\mu_i},\zeta_{\Sigma_i}\right) \tag{16}$$

I use the expectation maximization (EM) algorithm to fit the parameters of the mixture density network. The algorithm follows the approach presented in Delong, Lindholm, and Wüthrich (2021). At training iteration j, I use the neural network parameters form iteration $j-1$ to construct the posterior probabilities. Denote the posterior probability mass function for mixture component k at iteration $j-1$ as $\wp_k^{j-1}(\cdot)$. Note that one computes the iteration $j-1$ posterior probabilities after updating the neural network parameters

in iteration $j-1$. The posterior probability that x_0^{endo} is generated by component k is given by

$$\wp_k^{j-1}\left(x_0^{endo}\right) = \frac{\zeta_{\pi_k}^{j-1}\mathcal{N}\left(x_0^{endo}|\zeta_{\mu_k}^{j-1},\zeta_{\Sigma_k}^{j-1}\right)}{\sum_{i=1}^{N_{mix}}\zeta_{\pi_i}^{j-1}\mathcal{N}\left(x_0^{endo}|\zeta_{\mu_i}^{j-1},\zeta_{\Sigma_i}^{j-1}\right)} \tag{17}$$

Given the posterior probabilities, I update the neural network parameters so as to maximize a lower bound of the likelihood. The objective function, denoted by $Q(\cdot)$, at iteration j is given by

$$Q(\lambda) = \sum_{n=1}^{N_{batch}}\sum_{i=1}^{N_{mix}}\wp_i^{j-1}\left(x_{n,0}^{endo}\right)\left[\log\left(\zeta_{\pi_i}\right)+\log\left(\mathcal{N}\left(x_{n,0}^{endo}|\zeta_{\mu_i},\zeta_{\Sigma_i}\right)\right)\right] \tag{18}$$

The posterior probabilities are treated as constants in the objective function. Therefore, the gradient of $\wp_i^{j-1}(\cdot)$ with respect to the neural network parameters does not enter the updating equation. I update λ using a standard updating scheme such as ADAM.

$$\lambda_j = \texttt{GradientDescent}\left(-\nabla_\lambda Q, \lambda_{j-1}\right) \tag{19}$$

5.2 Discussion

Section 6.1 illustrates how to use the methodology in practice. In that section, I provide details on how to construct each term of the joint likelihood using the baseline real business cycle model as an example. Additionally, in Section 6.2 I test my methodology on a more challenging example using simulated data. The model is a medium-scale, two-agent New Keynesian model with 17 estimated parameters. The results show that the method proposed in this section is able to recover the parameters used to generate the simulated data.

In the implementation, all derivatives are computed using a standard automatic differentiation software package. Therefore, one does not need to write custom automatic differentiation rules or derive analytical derivatives to use the methodology. In particular, I use `ReverseDiff.jl` to compute the derivatives using reverse mode automatic differentiation. Reverse mode automatic differentiation tends to be more efficient than forward mode automatic differentiation in high dimensional settings.[2] The implementation samples the economic parameters, initial states, and innovations. Therefore, even if one is estimating few economic parameters in a model with few state variables, the size of the space one is sampling from is large due to the innovations.

When $N_{mix} > 1$, one can marginalize out the mixture selection parameter when computing the joint likelihood. In this case, Equation (5) becomes

$$\ln p\left(u,\epsilon^T,x_0|z^T\right) = \ln p(u) + \sum_{t=1}^{T}\ln p\left(\epsilon_t|u\right) + \ln\left(\sum_{i=1}^{N_{mix}}\exp\left(\ln\left(\zeta_{\pi_i}\right)+\sum_{t=1}^{T}\ln p\left(z_t|\zeta_{\mu_i},\zeta_{\Sigma_i},\epsilon^t,x_0,u\right)\right)\right) +$$
$$\ln p\left(x_0^{endo}|x_0^{exo},u\right) + \ln p\left(x_0^{exo}|u\right) + C \tag{20}$$

2. See Rackauckas (2022) for a discussion.

In practice, this means that, at each sampling step, one needs to simulate the model N_{mix} times. Each simulation starts from a different initial state.

6 Examples

I now illustrate my methodology using two examples. The first example is the standard real business cycle model. I use this example to demonstrate how to implement the methodology in practice and show that the method is able to recover the parameters used to generate the simulated data. The second example is a medium-scale two-agent New Keynesian model. I use this example to show that the method continues to work well as model complexity increases. Additionally, to highlight the importance of the mixture density network approximation of the initial state distribution, I compare the posterior estimation results from my methodology with the results from a methodology that assumes the initial states are equal to their steady-state values.

6.1 Real Business Cycle Model

The representative agent has utility defined over consumption, c. The utility function is iso-elastic with the inverse elasticity of intertemporal substitution given by η. The agent can save via capital, k. Capital depreciates at a rate δ. Total factor productivity, z, follows an $AR\left(1\right)$ process. The innovations to productivity are normally distributed with mean zero and standard deviation σ. As in Childers et al. (2022), the parameters to be estimated are the capital share of output, α; the discount factor, β; and the persistence of the productivity shock, ρ. Therefore, in this application, $u \triangleq (\alpha, \beta, \rho)$. Let $E\left[\cdot|\cdot\right]$ denote the conditional expectation operator. The planner's problem is

$$V\left(k_{-1}, z; u\right) = \max_{c} \frac{c^{1-\eta}}{1-\eta} + \beta E\left[V\left(k, z'; u\right)|k, z\right]$$

$$k = Ae^{z+\mu_z}\left(k_{-1}\right)^{\alpha} + \left(1-\delta\right)k_{-1} - c \tag{21}$$

$$z' = \rho z + \sigma\varepsilon' \qquad \varepsilon \sim \mathcal{N}\left(0,1\right) \tag{22}$$

where A is a scaling factor chosen so that steady-state output is equalized across parameterizations and μ_z is a normalizing constant so that $E\left[e^{z+\mu_z}\right] = 1$.

In this application, I approximate the consumption policy function using a neural network:

$$c = \Gamma_c\left(k_{-1}, z, u; \gamma_c\right) \tag{23}$$

I use the output of the approximate policy and the capital transition equation given in Equation (21) to compute end-of-period capital, k. The neural network is trained to minimize the error in the capital Euler

equation. The error is written as a percentage of consumption. The mapping of the policy to the equilibrium condition is given by

$$\mathfrak{R}\left[\left\{\Gamma_c\left(k_{-1},z,u;\gamma_c\right)\right\}\right]=L_{ee}=\frac{\left(\frac{1}{\beta E\left[(c')^{-\eta}\left(\alpha A e^{z'+\mu_z}k^{\alpha-1}+(1-\delta)\right)|k,z\right]}\right)^{-\frac{1}{\eta}}}{c}-1 \tag{24}$$

Once the policy is sufficiently accurate, one next needs to train the neural network that parameterizes the stationary distribution. As discussed in Section 3, only the distribution of the endogenous states, conditional on the exogenous states and estimated parameters, needs to be approximated. The distribution of the exogenous states, conditional on the estimated parameters, has a closed-form expression. In the example here, the distribution of z, conditional on (α,β,ρ), is

$$p\left(z|\alpha,\beta,\rho\right)=\mathcal{N}\left(z|0,\frac{\sigma}{\sqrt{1-\rho^2}}\right) \tag{25}$$

The neural network that produces the parameters for the approximate distribution takes as inputs the exogenous states and estimated parameters, (z,α,β,ρ). In this application, the approximate distribution is a normal distribution.[3] Therefore, the network has two outputs: the mean and the standard deviation:

$$\zeta_\mu,\zeta_\Sigma\leftarrow\Lambda\left(z,\alpha,\beta,\rho;\lambda\right) \tag{26}$$

$$p\left(k_{-1}|z,\alpha,\beta,\rho\right)=\mathcal{N}\left(k_{-1}|\zeta_\mu,\zeta_\Sigma\right) \tag{27}$$

Note that with a single mixture component, the objective function in Equation (18) reduces to the log-likelihood function. Even with a single mixture component, the mixture density network remains sufficiently flexible. Using a single mixture component does impose joint normality on the initial endogenous states. The mean vector and covariance matrix of this joint distribution, however, vary with the initial exogenous state.

For this example, the distribution of the innovation to the exogenous state variable at time t, conditional on the estimated parameters, is a standard normal distribution:

$$p\left(\varepsilon|\alpha,\beta,\rho\right)=\mathcal{N}\left(\varepsilon|0,1\right) \tag{28}$$

In the estimation, I use the log consumption series and investment series as observables. I assume both series are observed with independent, normally distributed measurement error. The measurement errors are mean zero. I denote the standard deviation of the measurement error for observable j as $\sigma_{j,me}$. The conditional densities of the observables are

$$p\left(\ln c_t|\varepsilon^t,k_0,z_0,\alpha,\beta,\rho\right)=\mathcal{N}\left(\ln c_t|\Gamma_c\left(k_{t-1},z_t,\alpha,\beta,\rho;\gamma_c\right),\sigma_{c,me}\right) \tag{29}$$

$$p\left(i_t|\varepsilon^t,k_0,z_0,\alpha,\beta,\rho\right)=\mathcal{N}\left(i_t|Ae^{z+\mu_z}\left(k_{t-1}\right)^\alpha-\Gamma_c\left(k_{t-1},z_t,\alpha,\beta,\rho;\gamma_c\right),\sigma_{i,me}\right) \tag{30}$$

3. I also experimented with two and three mixture components. In both cases, the network assigned a mixing weight of unity to one distribution and zero to the other(s).

To test the method, I simulate the model under a set of "true" values. Instead of directly estimating β, I follow the literature and estimate β_{draw} and compute β as

$$\beta = \frac{1}{\frac{\beta_{draw}}{100} + 1} \tag{31}$$

To generate the true data, I set α equal to 0.32, β_{draw} equal to 0.20, and ρ equal to 0.65. The list of parameters I hold fixed during the estimation is given in the first column of Table 1. The values of the parameters are given in the second column. I sample the true initial states from the mixture density network. Instead of directly sampling z_0 and k_{-1}, I sample $\widetilde{z}_0$ and $\widetilde{k}_{-1}$, which are both distributed $\mathcal{N}(0,1)$. I then compute z_0 and k_{-1} as follows:

$$z_0 = \widetilde{z}_0 \left(\frac{\sigma}{\sqrt{1-\rho^2}} \right) \tag{32}$$

$$\zeta_\mu, \zeta_\Sigma \leftarrow \Lambda\left(z_0, \alpha, \beta, \rho; \lambda\right) \tag{33}$$

$$k_{-1} = \zeta_\mu + \widetilde{k}_0 \zeta_\Sigma \tag{34}$$

Given the true parameters, initial conditions, and history of innovations, I simulate the model for 120 periods to generate the data. I then use `Turing.jl` to estimate the parameters using the No-U-Turn Sampler (Hoffman and Gelman 2014). I generate 7,500 samples with a 2,500 sample warm-up period. I set the standard deviations of the measurement errors equal to 20% of the standard deviation of the observable series. Table 2 presents the true values, prior shape, prior means, prior standard deviations, prior truncation bounds, posterior means, posterior 95% high-density intervals, and bulk effective sample size as a percentage of total samples for the estimated parameters. Figures 1 to 3 show kernel density plots and trace plots for the estimated parameters.

Table 1: Fixed Parameters.

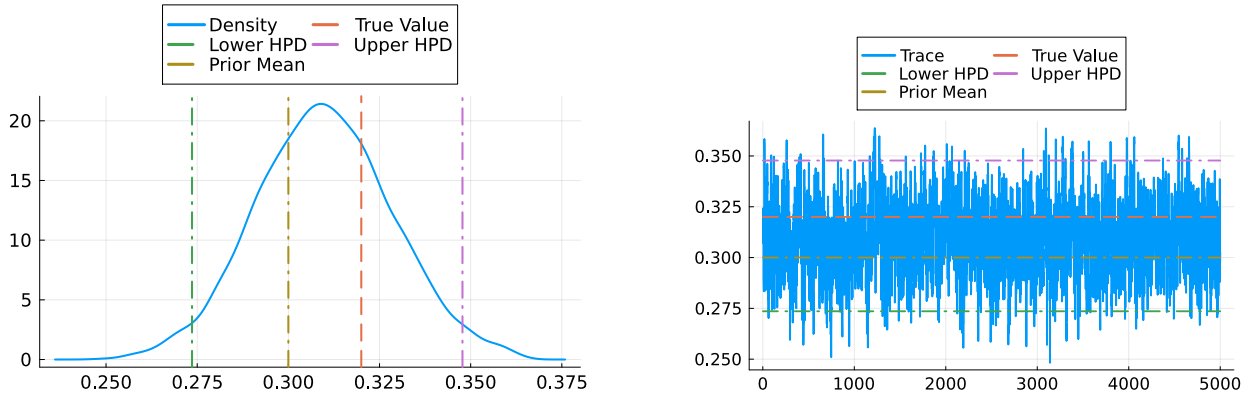
Parameter	Value	Description
δ	0.025	Capital depreciation rate
η	2.0	Inverse elasticity of intertemporal substitution
σ	0.01	Standard deviation of TFP shock innovations
y^*	1.0	Steady-state output

Notes: The table presents the parameters that are held fixed in the estimation procedure.

Table 2: True values, prior shapes, prior means, prior standard deviations, prior truncation bounds, posterior means, 95% high-density intervals, and bulk effective sample size percentage for the estimated parameters and initial states.

Parameter	True Value	Prior Shape	Prior Mean	Prior St. Dev.	Prior Bounds	Posterior Mean	95% HDI	Bulk-ESS %
α	0.3200	Normal	0.30	0.025	[0.2356,0.3644]	0.3098	[0.2736,0.3477]	13.9959
β_{draw}	0.2000	Gamma	0.25	0.1	[0.0663,0.5812]	0.2023	[0.0876,0.3244]	17.9016
ρ	0.6500	Beta	0.50	0.2	[0.0679,0.9321]	0.6280	[0.5656,0.6852]	9.2843
$\widetilde{z}_0$	1.2231	Normal	0.00	1.0	[-2.0,2.0]	1.1877	[0.8466,1.5298]	23.0313
$\widetilde{k}_0$	-0.2941	Normal	0.00	1.0	[-2.0,2.0]	-0.2743	[-0.4141,-0.1390]	23.9731

Notes: The table presents the parameter values used to generate the simulated data and the prior means in columns two and three, respectively. Column four presents the posterior mean. Column five presents the 95% high posterior density intervals.

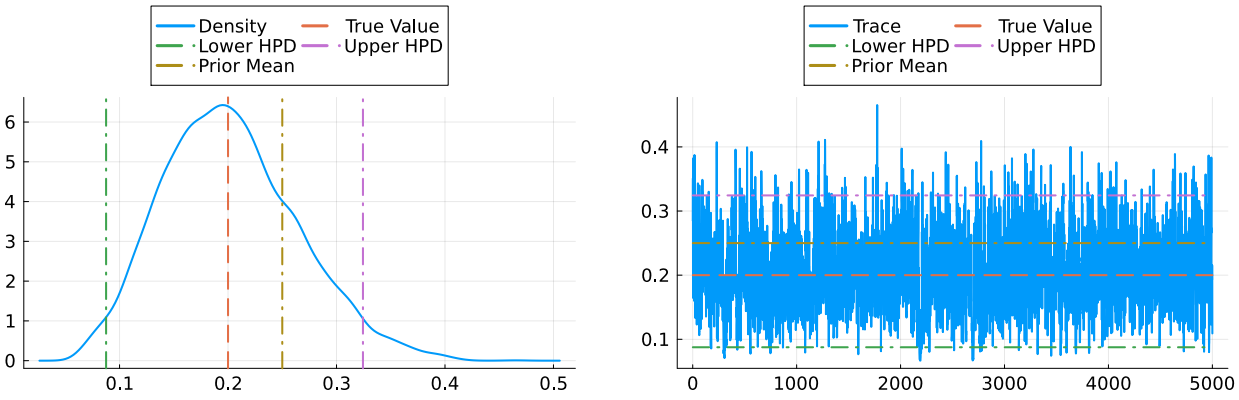


(a) Posterior Density

(b) Trace Plot

Figure 1: Estimation Results: α

Notes: Figure 1a shows the kernel density plot of the posterior distribution for α. Figure 1b shows the trace plot of the posterior sample. In both figures, the gold line is the prior mean, the orange line is the true value, the green line is the lower bound of the 95% high posterior density interval, and the purple line is the upper bound of the 95% high posterior density interval.

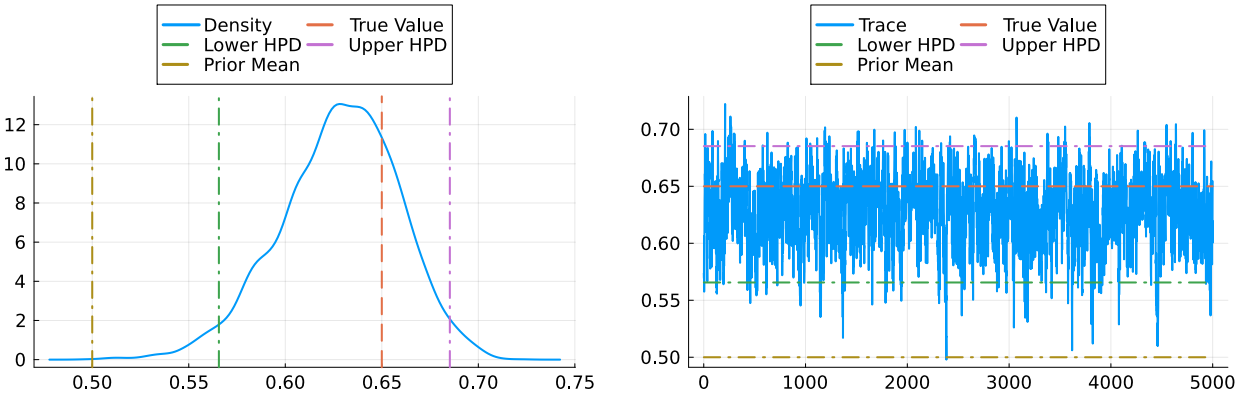


(a) Posterior Density

(b) Trace Plot

Figure 2: Estimation Results: β_{draw}

Notes: Figure 2a shows the kernel density plot of the posterior distribution for β_{draw}. Figure 2b shows the trace plot of the posterior sample. In both figures, the gold line is the prior mean, the orange line is the true value, the green line is the lower bound of the 95% high posterior density interval, and the purple line is the upper bound of the 95% high posterior density interval.

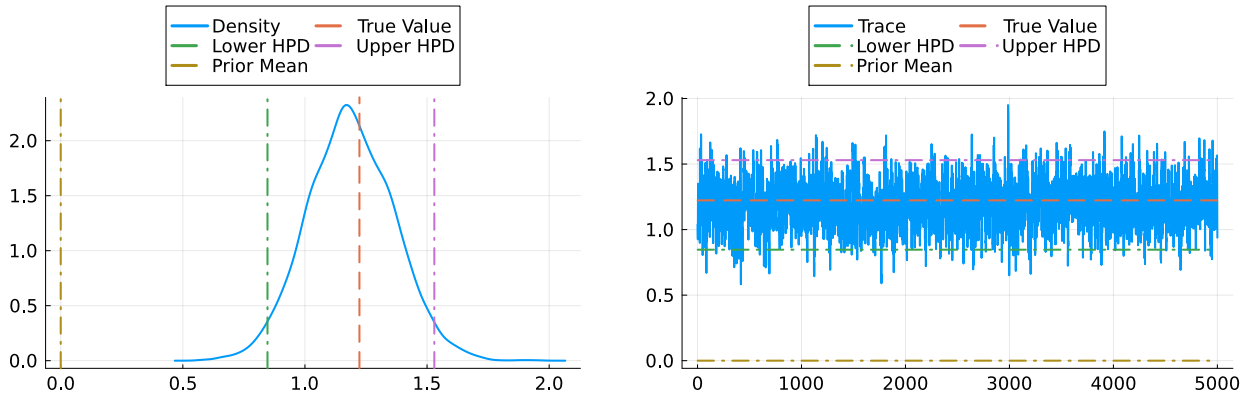


(a) Posterior Density

(b) Trace Plot

Figure 3: Estimation Results: ρ

Notes: Figure 3a shows the kernel density plot of the posterior distribution for ρ. Figure 3b shows the trace plot of the posterior sample. In both figures, the gold line is the prior mean, the orange line is the true value, the green line is the lower bound of the 95% high posterior density interval, and the purple line is the upper bound of the 95% high posterior density interval.

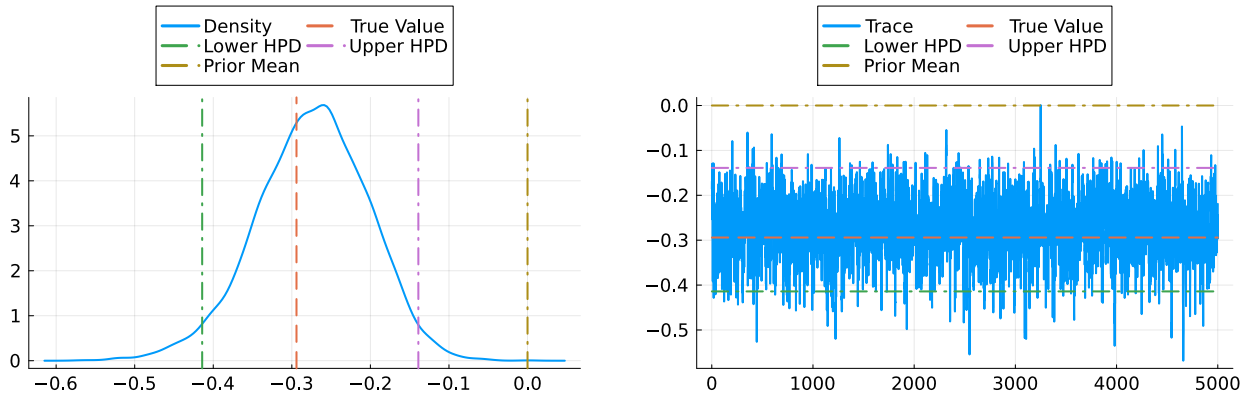


(a) Posterior Density

(b) Trace Plot

Figure 4: Estimation Results: $\widetilde{z}_0$

Notes: Figure 4a shows the kernel density plot of the posterior distribution for $\widetilde{z}_0$. Figure 4b shows the trace plot of the posterior sample. In both figures, the gold line is the prior mean, the orange line is the true value, the green line is the lower bound of the 95% high posterior density interval, and the purple line is the upper bound of the 95% high posterior density interval.



(a) Posterior Density

(b) Trace Plot

Figure 5: Estimation Results: $\widetilde{k}_{-1}$

Notes: Figure 5a shows the kernel density plot of the posterior distribution for $\widetilde{k}_{-1}$. Figure 5b shows the trace plot of the posterior sample. In both figures, the gold line is the prior mean, the orange line is the true value, the green line is the lower bound of the 95% high posterior density interval, and the purple line is the upper bound of the 95% high posterior density interval.

The table and figures show that the method is able to adequately recover the true values of the parameters and initial states. As shown in column eight of Table 2, all of the high-density intervals contain the true

values. Additionally, column nine highlights the efficiency of the method. The bulk effective sample size as a percentage of the sample size ranges from 9% to 23%. The trace plots (Figure 1b, Figure 2b, Figure 3b, Figure 4b, and Figure 5b) reinforce this point. The samples display very little auto-correlation and there are no clear signs of non-stationarity.

6.2 Medium-Scale Two-Agent New Keynesian Model

I now consider a medium-scale, two-agent New Keynesian model. I present the full details of the model in Section A.1. The model is populated by two types of households, labor unions, monopolistically competitive firms, a government, and a monetary authority. As in Bilbiie (2008), only a fraction of households, called "savers", have access to asset markets. The mass of savers is denoted by ω_s. Savers participate in the market for one-period nominal bonds and physical capital. Physical capital is subject to an adjustment cost. Additionally, capital investment is irreversible. The remaining households, called "hand-to-mouth" households, are excluded from asset markets. Their mass is given by $\omega_h = 1 - \omega_s$. I introduce nominal rigidities by assuming that the labor unions face a utility cost when adjusting wages, and firms face a price adjustment cost when adjusting prices. The monetary authority faces a zero lower bound on the nominal interest rate. Finally, I assume that there are shocks to the labor dis-utility scale, to total factor productivity, and to the policy rate. The model features four endogenous state variables and three exogenous state variables. I estimate 17 of the model parameters.

6.2.1 Simulated Data

I use six series as observables in the estimation procedure. The six series are the same as those used in Smets and Wouters (2007) and are commonly used as observables in medium-scale New Keynesian models. Specifically, I use the deviation of inflation from steady-state inflation, the deviation of the nominal interest rate from the steady-state nominal interest rate, the change in log output, the change in log investment, the change in log wages and the deviation of log hours from steady-state log hours.

To facilitate computation of the likelihood, I assume that the data are observed with independent, normally distributed measurement error. The measurement error for observable j in period t is denoted by $\epsilon_{j,me,t}$. The measurement error distributions all have zero mean. I set the standard deviations of the measurement errors equal to 5% of the standard deviation of the observable series. I denote the standard

deviation of the measurement error for observable j as $\sigma_{j,me}$. The measurement equations are

$$\text{Inflation}_t - \overline{\text{Inflation}} = \pi_t - \pi^* + \sigma_{\pi,me}\epsilon_{\pi,me,t} \qquad \epsilon_{\pi,me,t} \sim \mathcal{N}(0,1) \tag{35}$$

$$\text{Fed Funds}_t - \overline{\text{Fed Funds}} = r_t - r^* + \sigma_{r,me}\epsilon_{r,me,t} \qquad \epsilon_{r,me,t} \sim \mathcal{N}(0,1) \tag{36}$$

$$\Delta(\ln \text{RGDP})_t - \overline{\Delta(\ln \text{RGDP})} = \ln y_t - \ln y_{t-1} + \sigma_{y,me}\epsilon_{y,me,t} \qquad \epsilon_{y,me,t} \sim \mathcal{N}(0,1) \tag{37}$$

$$\Delta(\ln \text{Invest.})_t - \overline{\Delta(\ln \text{Invest.})} = \ln i_t^k - \ln i_{t-1}^k + \sigma_{i^k,me}\epsilon_{i^k,me,t} \qquad \epsilon_{i^k,me,t} \sim \mathcal{N}(0,1) \tag{38}$$

$$\Delta(\ln \text{Wages})_t - \overline{\Delta(\ln \text{Wages})} = \ln w_t - \ln w_{t-1} + \sigma_{w,me}\epsilon_{w,me,t} \qquad \epsilon_{w,me,t} \sim \mathcal{N}(0,1) \tag{39}$$

$$\Delta(\ln \text{Hours})_t - \overline{\Delta(\ln \text{Hours})} = \ln n_t - \ln n^* + \sigma_{n,me}\epsilon_{n,me,t} \qquad \epsilon_{n,me,t} \sim \mathcal{N}(0,1) \tag{40}$$

I generate the simulated observables as follows: First I draw a set of estimated parameters from a uniform distribution where the lower and upper bounds of the marginals are set to the lower and upper bounds for the parameters used in training the policy functions. I then draw the initial exogenous states from their stationary distribution. Since the labor dis-utility shock and the total factor productivity shock follow AR(1) processes with normally distributed innovations, and the monetary policy shocks are independent over time and identically distributed normal random variables, the stationary density of the exogenous states is a multivariate normal density. The mean vector and covariance matrix are fully determined once the estimated parameters are known. I input the estimated parameters and initial exogenous states into the mixture density network. I use the output of the mixture density network to construct the density of the initial endogenous states conditional on the exogenous states and estimated parameters. Given the conditional density, I generate a sample of the initial endogenous states. Next, I generate a sample of the innovations. Finally, given the parameters, initial conditions, and innovations, I use the approximate policy functions to simulate the model forward and construct the data set.

Note that, with this procedure for generating the true data, the true initial states are generally different from their steady-state values. I use this approach instead of assuming that the true data are generated by an economy that starts at its steady state since assuming that the initial states are drawn from the stationary distribution aligns more closely with reality than assuming that the initial states are equal to their steady-state values. If one were to assume that the initial states are equal to their steady-state values, then the posterior estimates for those parameters that affect the steady-state will be highly sensitive to the start date of the data sample.

The 17 parameters I estimate are the persistence of the total factor productivity shock (ρ_a), the persistence of the policy rate (ρ_m), the persistence of the labor dis-utility shock (ρ_n), the standard deviation of the innovations to the productivity shock (σ_a), the standard deviation of the monetary policy innovations (σ_m), the standard deviation of the innovations to the labor dis-utility shock (σ_n), the saver's discount factor (β), the inverse elasticity of intertemporal substitution (η), the inverse Frisch labor supply elasticity (φ), the capital share of output (α), the (inverse) scale of the price adjustment cost (κ_p), the (inverse) scale of the

wage adjustment cost (κ_w), the elasticity of substitution between goods (ϑ_p), the elasticity of substitution between labor services (ϑ_w), the responsiveness of the central bank to deviations of output from steady-state output (ϕ_y), the responsiveness of the central bank to deviations of inflation from steady-state inflation (ϕ_π), and the scale of the investment adjustment cost (κ_{i^k}). As in Section 6.1, I estimate β_{draw} and use Equation (31) to compute β. I hold the following parameters fixed throughout the estimation: the steady-state price inflation rate (π^*), the steady-state wage inflation rate (π_w^*), the steady-state labor supply (N^*), the steady-state level of output (y^*), the capital depreciation rate (δ), the mass of hand-to-mouth households (ω_h), and the supply of government bonds (b^s). The prior distributions for these parameters are given in Table 3. Table 4 provides the parameters that are held constant in the estimation. I derive the remaining parameters and steady-state values from the equilibrium conditions.

Table 3: Prior shapes, means, standard deviations, and truncation bounds.

Parameter	Prior Shape	Prior Mean	Prior Standard Deviation	Prior Bounds
ρ_a	Beta	0.75	0.2	[0.5,0.95]
ρ_m	Beta	0.75	0.2	[0.5,0.9]
ρ_r	Beta	0.75	0.2	[0.5,0.95]
σ_a	Normal	0.01	0.025	[0.005,0.015]
σ_m	Normal	0.003	0.01	[0.001,0.005]
σ_n	Normal	0.03	0.025	[0.005,0.05]
β_{draw}	Gamma	0.25	0.1	[0.077,0.539]
η	Normal	2.00	0.5	[1.05,3.0]
φ	Normal	1.50	0.25	[1.05,2.05]
α	Normal	0.33	0.025	[0.255,0.405]
κ_p	Normal	0.10	0.025	[0.05,0.2]
κ_w	Normal	0.10	0.025	[0.05,0.2]
ϑ_p	Normal	11.00	2.0	[5.0,17.0]
ϑ_w	Normal	11.00	2.0	[5.0,17.0]
ϕ_y	Normal	0.12	0.05	[0.02,0.22]
ϕ_π	Normal	2.00	0.25	[1.5,2.0]
κ_{i^k}	Normal	6.00	2.0	[2.0,10.0]

Table 4: Fixed parameters.

Parameter	Value	Description
π^*	$(1.02)^{\frac{1}{4}} - 1.0$	Quarterly price inflation target
π_w^*	$(1.02)^{\frac{1}{4}} - 1.0$	Quarterly wage inflation target
N^*	1.0	Steady-state labor supply
y^*	1.0	Steady-state output
δ	0.025	Capital depreciation rate
ω_h	0.21	Mass of hand-to-mouth households
b^s	0.0	Government bond supply

Notes: The table presents the parameters that are held fixed in the estimation procedure.

I simulate the model for 120 periods to generate the data. I set the standard deviations of the measurement errors equal to 5% of the standard deviation of the observable series. I use `Turing.jl` to estimate the parameters using the No-U-Turn Sampler. I generate 10,000 samples with a 10,000 sample warm-up period. Table 5 presents the posterior estimation results. The fourth and fifth columns present the posterior mean and 95% high-density intervals for the method where the initial states are drawn from a mixture density network. The sixth and seventh columns present analogous results for the case when the initial states are set to their steady-state values.

Table 5: True values, prior means, posterior means, and 95% high-density intervals for the estimated parameters.

Parameter	True Value	Prior Mean	MDN		SS	
			Posterior Mean	95% HDI	Posterior Mean	95% HDI
ρ_a	0.7320	0.75	0.7234	[0.7093,0.7374]	0.9122	[0.9101,0.9145]
ρ_m	0.7984	0.75	0.7831	[0.7672,0.7992]	0.8983	[0.8953,0.9000]
ρ_r	0.6148	0.75	0.6181	[0.5903,0.6456]	0.5008	[0.5000,0.5023]
σ_a	0.0079	0.01	0.0080	[0.0076,0.0084]	0.0096	[0.0094,0.0099]
σ_m	0.0037	0.003	0.0035	[0.0034,0.0037]	0.0050	[0.0050,0.0050]
σ_n	0.0170	0.03	0.0167	[0.0131,0.0202]	0.0499	[0.0497,0.0500]
β_{draw}	0.3258	0.25	0.3302	[0.3125,0.3475]	0.5380	[0.5359,0.5391]
η	2.0337	2.00	1.9612	[1.8430,2.0816]	2.4073	[2.3379,2.4777]
φ	1.5600	1.50	1.5128	[1.3779,1.6403]	1.9665	[1.9189,2.0115]
α	0.3534	0.33	0.3527	[0.3471,0.3582]	0.3834	[0.3812,0.3856]
κ_p	0.0994	0.10	0.1067	[0.0972,0.1154]	0.0506	[0.0500,0.0517]
κ_w	0.1553	0.10	0.1528	[0.1349,0.1713]	0.1992	[0.1975,0.2000]
ϑ_p	13.5703	11.00	13.6334	[13.0474,14.1902]	5.0073	[5.0000,5.0223]
ϑ_w	13.0092	11.00	12.0807	[10.6744,13.5401]	5.0212	[5.0000,5.0653]
ϕ_y	0.0957	0.12	0.0862	[0.0694,0.1028]	0.2194	[0.2184,0.2200]
ϕ_π	1.8014	2.00	1.8098	[1.7270,1.8945]	2.2845	[2.2345,2.3306]
κ_{ik}	5.1525	6.00	4.8711	[4.4724,5.2433]	6.2787	[6.0913,6.4748]

Notes: The table presents the parameter values used to generate the simulated data and the prior means in columns two and three, respectively. Columns four and five present the posterior means (column four) and 95% high posterior density intervals (column five) for the estimation procedure in which the initial states are drawn from the mixture density network. The analogous results for the procedure in which the initial states are set equal to their steady-state values are presented in columns six and seven. All computations are done on a six-core, twelve-thread Intel(R) Core(TM) i7-10750H processor. The sample that uses the mixture density network takes 33135.2 seconds (9.2 hours). The sample that uses the steady-state as the initial state takes 35019.82 seconds (9.73 hours).

First focusing on the method that uses the mixture density network, one sees that all of the 95% high-density intervals contain the true value. Additionally, the posterior mean tends to be close to the true value. The same cannot be said when analyzing the results where the initial state is set equal to the steady-state value. In fact, none of the 95% high-density intervals contain the true value. Many of the posterior means are near one of the prior truncation bounds. Additionally, the high posterior density intervals for

these parameters are narrowly concentrated near the bound. For example, the lower bounds of the 95% high posterior density intervals for the elasticity of substitution between goods, ϑ_p, and the elasticity of substitution between labor services, ϑ_w, are both equal to the prior lower bound of 5.0. The upper bounds of the high posterior density intervals are 5.02 and 5.07, while the true values are 13.57 and 13.01, respectively. Overall, the results presented in Table 5 show that the mixture density network approximation of the initial state distribution plays a crucial role in posterior inference.

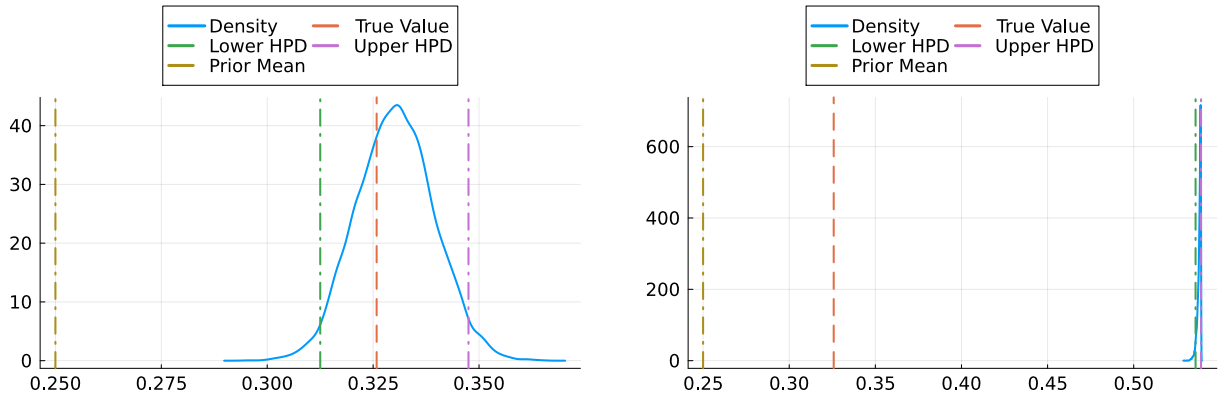
6.2.2 Role of the Initial State Density

To understand the role the mixture density network approximation plays in the performance of the algorithm, it is useful to analyze the posterior distribution of β_{draw}. Recall that the discount factor, β, is computed as

$$\beta = \frac{1}{\frac{\beta_{draw}}{100} + 1}$$

With a fixed steady-state inflation rate, there is a one-to-one mapping between the nominal interest rate and the discount factor given by the saver household's Euler equation for bonds.

Figure 6 shows the posterior densities for β_{draw} from the procedure that assumes the initial states are drawn from a mixture density network (Figure 6a) and the procedure where the initial states are equal to their steady-state values (Figure 6b). In the highlighted example, the steady-state nominal interest rate is 0.82%. The true initial nominal interest rate, however, is 1.34%. Figure 7 plots the posterior density of $\widetilde{r}_0$ as well as the value of $\widetilde{r}_0$ used to generate the data. Given that the initial nominal interest rate is above the steady-state interest rate, the procedure that assumes the economy always starts from the steady-state generates a posterior that concentrates around values of β_{draw} greater than the true value of β_{draw}, leading to a lower value of β.



(a) Mixture Density Network

(b) Steady-State

Figure 6: Posterior distributions of β_{draw}.

Notes: The figures show the kernel density plots of the posterior distributions for β_{draw}. Figure 6a (left) uses a mixture density network to approximate the initial state distribution. Figure 6b (right) assumes the initial states are equal to their steady-state values. The vertical gold line in each figure is the prior mean. The vertical orange line is the true value. The vertical green line is the lower bound of the 95% high posterior density interval. The vertical purple line is the upper bound of the 95% high posterior density interval.

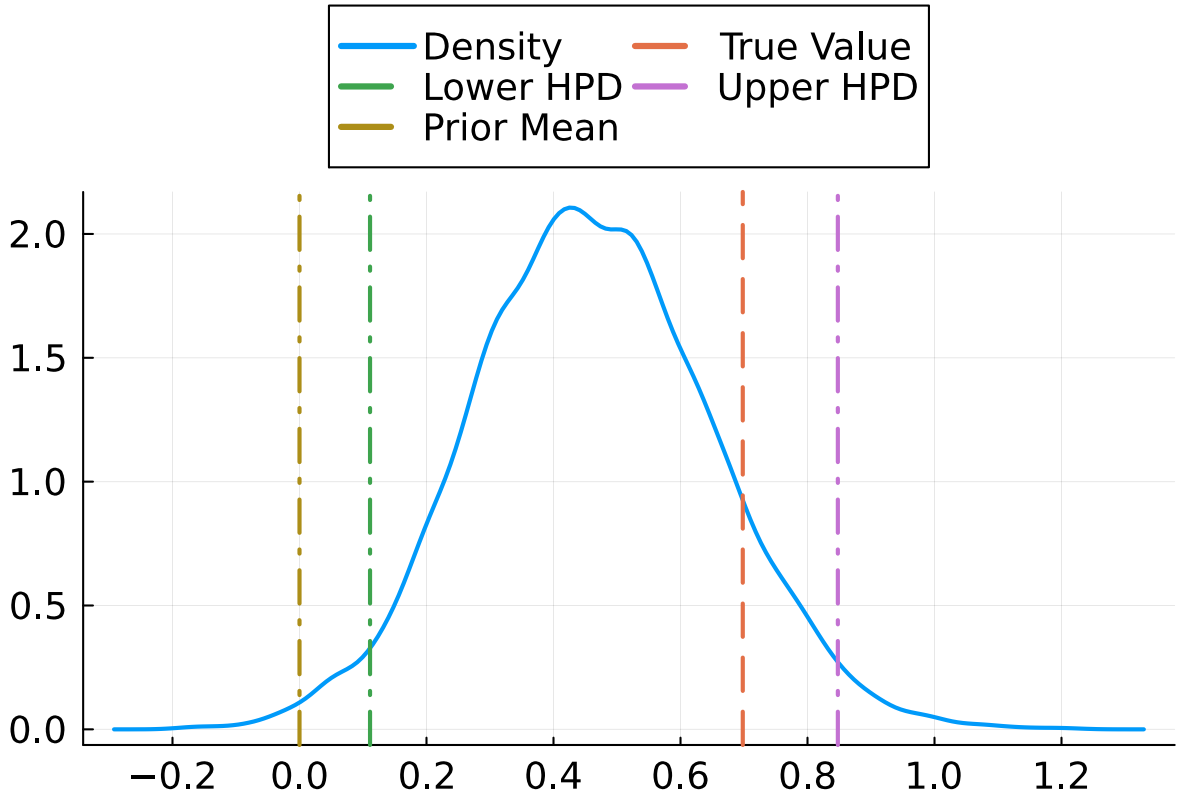


Figure 7: Posterior distribution of $\widetilde{r}_0$

7 Conclusion

To address some of the difficulties of estimating non-linear models using Bayesian methods, I build on recent work by Kase, Melosi, and Rottner (2022) and Childers et al. (2022) and develop a methodology for estimating non-linear, globally solved models with aggregate shocks. A key innovation in my methodology is the use of mixture density networks to approximate the initial state distribution. I show how this approach greatly improves posterior inference compared to a method that assumes the initial states are equal to their steady-state values. As a secondary contribution, I show how to incorporate insights from the deep reinforcement learning literature to improve neural network training stability and testing accuracy. The methodology can be used to estimate larger-scale models, including New Keynesian models with idiosyncratic risk. In future work, I plan on extending the methodology to solve and estimate models with mixed discrete-continous choices.

References

Adjemian, Stéphane, Houtan Bastani, Michel Juillard, Fréderic Karamé, Ferhat Mihoubi, Willi Mutschler, Johannes Pfeifer, Marco Ratto, Normann Rion, and Sébastien Villemot. 2022. *Dynare: Reference Manual Version 5.* Dynare Working Papers 72. CEPREMAP, January. https://ideas.repec.org/p/cpm/dynare/072.html.

Auclert, Adrien, Matthew Rognlie, and Ludwig Straub. 2018. *The Intertemporal Keynesian Cross.* Working Paper, Working Paper Series 25020. National Bureau of Economic Research, September. doi:10.3386/w25020. http://www.nber.org/papers/w25020.

Azinovic, Marlon, Luca Gaegauf, and Simon Scheidegger. 2022. "Deep Equilibrium Nets." *International Economic Review* 63, no. 4 (November): 1471–1525. doi:10.1111/iere.12575. https://ideas.repec.org/a/wly/iecrev/v63y2022i4p1471-1525.html.

Bilbiie, Florin O. 2008. "Limited asset markets participation, monetary policy and (inverted) aggregate demand logic." *Journal of Economic Theory* 140 (1): 162–196. ISSN: 0022-0531. doi:https://doi.org/10.1016/j.jet.2007.07.008. https://www.sciencedirect.com/science/article/pii/S0022053107001159.

Bishop, Christopher M. 1994. "Mixture density networks." https://api.semanticscholar.org/CorpusID:118227751.

Childers, David, Jesús Fernández-Villaverde, Jesse Perla, Christopher Rackauckas, and Peifan Wu. 2022. *Differentiable State-Space Models and Hamiltonian Monte Carlo Estimation.* Working Paper, Working Paper Series 30573. National Bureau of Economic Research, October. doi:10.3386/w30573. http://www.nber.org/papers/w30573.

Delong, Łukasz, Mathias Lindholm, and Mario V. Wüthrich. 2021. "Gamma Mixture Density Networks and their application to modelling insurance claim amounts." *Insurance: Mathematics and Economics* 101:240–261. ISSN: 0167-6687. doi:https://doi.org/10.1016/j.insmatheco.2021.08.003. https://www.sciencedirect.com/science/article/pii/S0167668721001232.

Fernández-Villaverde, Jesús, and Pablo A. Guerrón-Quintana. 2021. "Estimating DSGE Models: Recent Advances and Future Challenges." *Annual Review of Economics* 13 (1): 229–252. doi:10.1146/annurev-economics-081020-044812. eprint: https://doi.org/10.1146/annurev-economics-081020-044812. https://doi.org/10.1146/annurev-economics-081020-044812.

Fernández-Villaverde, Jesús, Galo Nũno, George Sorg-Langhans, and Maximilian Vogler. 2020. "Solving High-Dimensional Dynamic Programming Problems using Deep Learning." https://api.semanticscholar.org/CorpusID:229330871.

Herbst, Edward P., and Frank Schorfheide. 2015. *Bayesian estimation of DSGE models* [in eng]. The Econometric and Tinbergen Institutes Lectures. Princeton, NJ: Princeton University Press. ISBN: 1-4008-7373-8.

Hoffman, Matthew D., and Andrew Gelman. 2014. "The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo." *Journal of Machine Learning Research* 15 (47): 1593–1623. http://jmlr.org/papers/v15/hoffman14a.html.

Innes, Michael, Elliot Saba, Keno Fischer, Dhairya Gandhi, Marco Concetto Rudilosso, Neethu Mariya Joy, Tejan Karmali, Avik Pal, and Viral Shah. 2018. "Fashionable Modelling with Flux." *CoRR* abs/1811.01457. arXiv: 1811.01457. https://arxiv.org/abs/1811.01457.

Judd, Kenneth L. 1998. *Numerical Methods in Economics* [in eng]. 1st ed. Vol. 1. MIT Press Books. The MIT Press. ISBN: 9780262100717.

Kase, Hanno, Leonardo Melosi, and Matthias Rottner. 2022. *Estimating Nonlinear Heterogeneous Agents Models with Neural Networks.* CEPR Discussion Papers 17391. C.E.P.R. Discussion Papers, June. https://ideas.repec.org/p/cpr/ceprdp/17391.html.

Lillicrap, Timothy P., Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. 2016. "Continuous control with deep reinforcement learning." In *ICLR,* edited by Yoshua Bengio and Yann LeCun. http://dblp.uni-trier.de/db/conf/iclr/iclr2016.html#LillicrapHPHETS15.

Lin, Long-Ji. 1992. "Self-Improving Reactive Agents Based on Reinforcement Learning, Planning and Teaching." *Mach. Learn.* (USA) 8, no. 3–4 (May): 293–321. ISSN: 0885-6125. doi:10.1007/BF00992699. https://doi.org/10.1007/BF00992699.

Maliar, Lilia, Serguei Maliar, and Pablo Winant. 2021. "Deep learning for solving dynamic economic models." *Journal of Monetary Economics* 122 (C): 76–101. doi:10.1016/j.jmoneco.2021.07. https://ideas.repec.org/a/eee/moneco/v122y2021icp76-101.html.

Rotemberg, Julio J. 1982. "Sticky Prices in the United States." *Journal of Political Economy* 90, no. 6 (December): 1187–1211. doi:10.1086/261117. https://ideas.repec.org/a/ucp/jpolec/v90y1982i6p1187-1211.html.

Smets, Frank, and Rafael Wouters. 2007. "Shocks and Frictions in US Business Cycles: A Bayesian DSGE Approach." *American Economic Review* 97, no. 3 (June): 586–606. doi:10.1257/aer.97.3.586. https://www.aeaweb.org/articles?id=10.1257/aer.97.3.586.

Villemot, Sébastien. 2011. *Solving rational expectations models at first order: what Dynare does.* Dynare Working Papers 2. CEPREMAP, April. https://ideas.repec.org/p/cpm/dynare/002.html.

A Appendix

A.1 Two-Agent New Keynesian Model Appendix

In this appendix, I present the details of the model I use in Section 6.2. I first define the problems of the two household types. I then present the problem of a representative labor union. Next, I describe the problem of a representative firm. Finally, I describe the problems of the monetary authority and government. I conclude the section with the full list of equilibrium conditions.

A.1.1 Households

The model features two types of households, as in Bilbiie (2008). The first type is the "saver" type. There is a continuum of identical saver households uniformly distributed over the interval $[0, \omega_s)$. The second type of household is the "hand-to-mouth" household type. There is a continuum of identical hand-to-mouth households uniformly distributed over the interval $[\omega_s, 1]$. Since there is no heterogeneity within type, all households of a given type make the same choices. Therefore, there exists a representative saver and a representative hand-to-mouth household, and I do not need to keep track of the decisions of all households.

The (representative) saver has preferences defined over composite consumption, c^s, and composite labor, n. Composite consumption is a CES aggregate of differentiated intermediate goods. Composite labor is a CES aggregate of differentiated labor services. The utility function for consumption is iso-elastic, with the inverse elasticity of intertemporal substitution denoted by η. The labor dis-utility function is of the power form with the scale given by ψ and the power given by $(1 + \varphi)$. The scale of the dis-utility from labor is subject to a shock, z^n, which follows an $AR\,(1)$ process. The persistence of the labor dis-utility shock is ρ_n, and the conditional standard deviation is given by σ_n. The innovations to the labor dis-utility shock are drawn from a normal distribution with mean zero.

$$z^n = \rho_n z^n_{-1} + \sigma_n \varepsilon_n \qquad \varepsilon_n \sim \mathcal{N}\,(0, 1) \tag{41}$$

Labor supply is determined by labor unions, discussed further in Section A.1.2, and is the same for all households. Each household receives a wage w for each unit of labor it supplies, n. The saver household is allowed to trade in one-period nominal bonds. The real quantity of bonds is denoted by b. The net nominal return on bonds is denoted by r_{-1}. The second asset that saver households trade in is physical capital, k. The net real return on capital is denoted by r^k. Capital is subject to an investment adjustment cost. Denote the gross level of investment as i^k. Investment and investment adjustment costs use the same

CES aggregator as consumption. The adjustment cost depends on the ratio of current period investment to prior period investment. The investment adjustment cost function is denoted by $S(\cdot)$. In the numerical application, I use the following functional form:

$$S\left(\frac{i^k}{i^k_{-1}}\right) \triangleq \frac{\kappa_{i^k}}{2}\left(\frac{i^k}{i^k_{-1}} - 1\right)^2 \tag{42}$$

Furthermore, I assume that investment is irreversible ($i^k \geq 0$). Absent this assumption, the saver household can increase both the capital stock and its total resources by choosing a negative level of investment. In addition to labor income and asset income, the saver receives transfers denoted by T. The saver household discounts the future at a rate β. Let $V_s(\cdot)$ denote the value function of the representative saver. The problem of the saver household is

$$V_s\left(b_{-1}, k_{-1}, \left(i^k\right)_{-1}, w_{-1}, r_{-1}, z^a, z^m, z^n\right) = \max_{c^s, i^k}\left\{\frac{\left(c^s\right)^{1-\eta}}{1-\eta} - e^{\mu_n + z^n}\psi\frac{n^{1+\varphi}}{1+\varphi} + \right.$$
$$\left. \beta E\left[V_s\left(b, k, i^k, w, r, z'_a, z'_m, z'_n\right)|b, k, i^k, w, r, z_a, z_m, z_n\right]\right\} \tag{43}$$
$$\frac{1+r_{-1}}{1+\pi}b_{-1} - c^s - i^k + wn + r^k k_{-1} + T - b = 0 \tag{44}$$
$$\left(1-\delta\right)k_{-1} + \left(1 - S\left(\frac{i^k}{i^k_{-1}}\right)\right)i^k - k = 0 \tag{45}$$

where μ_n is a normalizing constant chosen so that the unconditional expectation of $e^{\mu_n + z^n}$ is equal to unity. Denote the multiplier on the saver's budget constraint as λ^s, the multiplier on the capital transition equation as μ^s, and the ratio of μ^s to λ^s as q. The optimal policies of the saver household satisfy the following first-order conditions:

$$\left(c^s\right)^{-\eta} - \lambda^s = 0 \tag{46}$$
$$-\lambda^s + \beta E_X\left[\left(\lambda^s\right)'\frac{1+r}{1+\pi'}\right] = 0 \tag{47}$$
$$-q + \beta E_X\left[\frac{\left(\lambda^s\right)'}{\lambda^s}\left[\left(r^k\right)' + \left(1-\delta\right)q'\right]\right] = 0 \tag{48}$$
$$-1 + q\left(1 - S\left(\frac{i^k}{i^k_{-1}}\right) - dS\left(\frac{i^k}{i^k_{-1}}\right)\frac{i^k}{i^k_{-1}}\right) + \beta E_X\left[\frac{\left(\lambda^s\right)'}{\lambda^s}q' dS\left(\frac{\left(i^k\right)'}{i^k}\right)\left(\frac{\left(i^k\right)'}{i^k}\right)^2\right] = 0 \tag{49}$$
$$\frac{1+r_{-1}}{1+\pi}b_{-1} - c^s - i^k + wn + r^k k_{-1} + T - b = 0 \tag{50}$$
$$\left(1-\delta\right)k_{-1} + \left(1 - S\left(\frac{i^k}{i^k_{-1}}\right)\right)i^k - k = 0 \tag{51}$$

The (representative) hand-to-mouth household uses the same utility function as the saver household to value (composite) consumption, c^h, and (composite) labor, n. Unlike the saver household, the hand-to-mouth household is excluded from participation in asset markets. The hand-to-mouth household receives labor income and transfers. Since the hand-to-mouth household is excluded from asset markets, the household's

problem is static. Let V_h denote the value function of the representative hand-to-mouth household. The problem of the hand-to-mouth household is

$$V_h\left(b_{-1}, k_{-1}, \left(i^k\right)_{-1}, w_{-1}, r_{-1}, z^a, z^m, z^n\right) = \max_{c^h}\left\{\frac{\left(c^h\right)^{1-\eta}}{1-\eta} - e^{\mu_n+z^n}\psi\frac{n^{1+\varphi}}{1+\varphi}\right\} \tag{52}$$

$$wn + T - c^h = 0 \tag{53}$$

Denote the Lagrange multiplier on the hand-to-mouth household's budget constraint as λ^h. The optimal policies satisfy the following first-order conditions:

$$\left(c^h\right)^{-\eta} - \lambda^h = 0 \tag{54}$$

$$wn + T - c^{=}0 \tag{55}$$

A.1.2 Labor Union

Following the literature on sticky wages, I assume that each household provides a continuum of differentiated labor services indexed by j. The differentiated labor services are imperfect substitutes with the elasticity of substitution given by ϑ_w. The differentiated labor services are aggregated into a composite labor service using a CES aggregator and sold to the representative firm.

The wage for each type of labor service is determined by a labor union. The labor unions set wages to maximize the population-weighted sum of utilities of the households subject to Rotemberg (1982) utility adjustment cost as in Auclert, Rognlie, and Straub (2018). The utility cost is paid whenever wage growth for labor services of type j, $\left(\frac{w_j}{(w_j)_{-1}}\right)$, deviates from target wage inflation, $1+\pi_w^*$. Let $W(\cdot)$ denote the value function of a labor union. The problem of labor union j is

$$\begin{aligned} W\left(\left(w_j\right)_{-1}, k_{-1}, \left(i^k\right)_{-1}, w_{-1}, r_{-1}, z^a, z^m, z^n\right) = \max_{\widehat{w}} & \sum_{i\in\{h,s\}}\omega_i\left[\frac{\left(c^i\right)^{1-\eta}}{1-\eta} - e^{\mu_n+z^n}\psi\frac{n^{1+\varphi}}{1+\varphi}\right] - \\ & \frac{\vartheta_w}{2\kappa_w}\left(\log\left(\left(\frac{1}{1+\pi_w^*}\right)\frac{\widehat{w}}{(w_j)_{-1}}\right)\right)^2 + \\ & \beta E_{(k,i^k,w,r,z_a,z_m,z_n)}\left[W\left(\widehat{w}, k, i^k, w, r, \left(z^a\right)', \left(z^m\right)', \left(z^n\right)'\right)\right] \end{aligned} \tag{56}$$

$$n_j = \left(\frac{\widehat{w}}{w}\right)^{-\vartheta_w} n \tag{57}$$

Optimizing with respect to the wage and imposing symmetry results in the following wage Phillips curve:

$$\begin{aligned} \log\left(\frac{1+\pi^w}{1+\pi_w^*}\right) = & \kappa_w\left[\psi e^{(\mu_n+z^n)}n^{1+\varphi} - \frac{\vartheta_w-1}{\vartheta_w}\mathcal{U}\left(c^h, c^s\right)wn\right] + \\ & \beta E_X\left[\log\left(\frac{1+\left(\pi^w\right)'}{1+\pi_w^*}\right)\middle| b, k, i^k, w, r, z_a, z_m, z_n\right] \end{aligned} \tag{58}$$

where $\mathcal{U}\left(c^h, c^s\right)$ is given by

$$\mathcal{U}\left(c^h, c^s\right) \triangleq \sum_{i \in \{h,s\}} \omega_i \left(c^i\right)^{-\eta} \tag{59}$$

A.1.3 Firms

There is a continuum of intermediate goods producers indexed by j. The intermediate goods are imperfect substitutes. The intermediate goods are aggregated into a composite good according to a CES aggregator with the elasticity of substitution given by ϑ_p. Each intermediate good producer operates a Cobb Douglas production function. The firms use labor, n_j^d, and capital, k_j^d, to produce output, y_j. The elasticity of output with respect to capital is given by α. The scale of the production function is given by A. Additionally, output is subject to a total factor productivity shock, z^a. The shock is common across all intermediate goods producers. The shock follows an $AR\left(1\right)$ process. The persistence of the shock is given by ρ_a, and the conditional standard deviation is σ_a. The innovation to the total factor productivity shock, ε_a, follows a normal distribution with mean zero.

$$z^a = \rho_a z_{-1}^a + \sigma_a \varepsilon_a \qquad\qquad \varepsilon_a \sim \mathcal{N}\left(0,1\right) \tag{60}$$

Firm j sets its price to maximize the present discounted value of dividends. I assume that firm j discounts the future at the constant rate β. I assume that dividends are taxed at a 100% rate and transferred to households. When setting its price, a firm faces a Rotemberg (1982) adjustment cost. The cost is non-zero whenever inflation for the firm, $\frac{p_j}{(p_j)_{-1}}$, deviates from target inflation, $(1+\pi^*)$. The problem of firm j can be divided into two sub-problems: a cost minimization problem and a price setting problem.

Cost Minimization

In the first stage, firm j chooses its combination of capital and labor so as to minimize the cost of producing $\bar{y}$ units of output:

$$\min_{\widehat{k}^d, \widehat{n}^d} r^k \widehat{k}^d + w\widehat{n}^d \tag{61}$$

$$\bar{y} - Ae^{(\mu_a + z^a)} \left(\widehat{k}^d\right)^{\alpha} \left(\widehat{n}^d\right)^{1-\alpha} = 0 \tag{62}$$

First-order conditions with respect to labor and capital are given by

$$w - \lambda_j^{mc} \left(1-\alpha\right) Ae^{(\mu_a + z^a)} \left(k_j^d\right)^{\alpha} \left(n_j^d\right)^{-\alpha} = 0 \tag{63}$$

$$r^k - \lambda_j^{mc} \alpha Ae^{(\mu_a + z^a)} \left(k_j^d\right)^{\alpha-1} \left(n_j^d\right)^{1-\alpha} = 0 \tag{64}$$

Note that λ_j^{mc} is the marginal cost. Therefore, I will use the notation mc throughout the remainder of these notes. I omit the subscript j since marginal costs are equal across firms.

Price Setting Problem

Given the marginal cost and demand schedule, the firm chooses its price so as to maximize the present discounted value of dividends. Let $J(\cdot)$ denote the value function of an intermediate good producer. The price setting problem of intermediate good producer j is

$$J\left(\left(p_j\right)_{-1}, k_{-1}, \left(i^k\right)_{-1}, w_{-1}, r_{-1}, z^a, z^m, z^n\right) = \max_{\widehat{p}} \frac{\widehat{p}}{p} y_j - mcy_j -$$
$$\frac{\vartheta_p}{2\kappa_p}\left[\log\left(\left(\frac{1}{1+\pi^*}\right)\frac{\widehat{p}}{\left(p_j\right)_{-1}}\right)\right]^2 y +$$
$$\beta E_{(k,i^k,w,r,z_a,z_m,z_n)}\left[J\left(\widehat{p}, k, i^k, w, r, \left(z^a\right)', \left(z^m\right)', \left(z^n\right)'\right)\right] \quad (65)$$
$$y_j = y\left(\frac{\widehat{p}}{p}\right)^{-\vartheta_p} \quad (66)$$

Optimizing with respect to $\widehat{p}$ and imposing symmetry generates the standard New Keynesian Phillips curve:

$$\log\left(\frac{1+\pi}{1+\pi^*}\right) = \kappa_p\left(mc - \frac{\vartheta_p - 1}{\vartheta_p}\right) + \beta E\left[\log\left(\frac{1+\pi'}{1+\pi^*}\right)\frac{y'}{y}\middle| b, k, i^k, w, r, z_a, z_m, z_n\right] \quad (67)$$

A.1.4 Monetary Policy and Government

The central bank sets the nominal interest rate subject to the zero lower bound. The central bank responds to deviations of output from its steady-state level, y^*, and deviations of inflation from target inflation, π^*. The strengths of the responses are captured by the parameters ϕ_y and ϕ_π. The policy rate is persistent with the degree of persistence captured by ρ_m. Finally, the policy rate is subject to a shock, z^m. The conditional standard deviation of the shock is denoted by σ_m. The monetary policy shock innovation, denoted by ε_m, follows a mean zero normal distribution.

$$z^m = \sigma_m \varepsilon_m \qquad \varepsilon_m \sim \mathcal{N}(0,1) \quad (68)$$

The nominal interest is given by

$$r = \max\left\{(1+r_{-1})^{\rho_m}\left((1+r^*)\left(\frac{y}{y^*}\right)^{\phi_y}\left(\frac{1+\pi}{1+\pi^*}\right)^{\phi_\pi}\right)^{1-\rho_m} e^{\mu_m + z^m}, 1\right\} - 1 \quad (69)$$

where μ_m is a normalizing constant chosen so that the unconditional expectation of $e^{z_m+\mu_m}$ is equal to unity.

The government runs a balanced budget period by period. Government revenues comprise proceeds from bond issuances and taxes on dividends. The government provides a fixed supply of bonds, b^s. The government uses its revenues to pay off the outstanding bonds and make transfers to the two types of households. The lump-sum transfers are the same for all households. The government budget constraint is given by

$$D + b^s = \frac{1+r_{-1}}{1+\pi} b^s + T \quad (70)$$

A.1.5 Equilibrium Conditions

The state variables in the model are

$$\left\{b_{-1}, k_{-1}, i^k_{-1}, w_{-1}, r_{-1}, z^a, z^m, z^n\right\}$$

In the equilibrium conditions listed below, the state is denoted by X. The non-predetermined variables are

$$\left\{\left\{c^i\right\}, \left\{\lambda^i\right\}, q, n, y, k^d, n^d, mc, r^k, \pi, \pi^w, T, D\right\}$$

The equilibrium conditions for the endogenous variables are

$$[\text{FOC } c^s] \qquad (c^s)^{-\eta} - \lambda^s = 0 \tag{71}$$

$$[\text{FOC } b] \qquad -\lambda^s + \beta E_X\left[(\lambda^s)' \frac{1+r}{1+\pi'}\right] = 0 \tag{72}$$

$$[\text{FOC } k] \qquad -q + \beta E_X\left[\frac{(\lambda^s)'}{\lambda^s}\left[\left(r^k\right)' + (1-\delta)\, q'\right]\right] = 0 \tag{73}$$

$$\left[\text{FOC } i^k\right] \qquad -1 + q\left(1 - S\left(\frac{i^k}{i^k_{-1}}\right) - dS\left(\frac{i^k}{i^k_{-1}}\right)\frac{i^k}{i^k_{-1}}\right) + \beta E_X\left[\frac{(\lambda^s)'}{\lambda^s} q' dS\left(\frac{\left(i^k\right)'}{i^k}\right)\left(\frac{\left(i^k\right)'}{i^k}\right)^2\right] = 0 \tag{74}$$

$$[\text{FOC } \lambda^s] \qquad \frac{1+r_{-1}}{1+\pi} b_{-1} - c^s - i^k + wn + r^k k_{-1} + T - b = 0 \tag{75}$$

$$[\text{FOC } \mu^s] \qquad (1-\delta)\, k_{-1} + \left(1 - S\left(\frac{i^k}{i^k_{-1}}\right)\right) i^k - k = 0 \tag{76}$$

$$\left[\text{FOC } c^h\right] \qquad \left(c^h\right)^{-\eta} - \lambda^h = 0 \tag{77}$$

$$\left[\text{FOC } \lambda^h\right] \qquad wn + T - c^h = 0 \tag{78}$$

$$[\text{Wage PC}] \qquad \kappa_w\left[\psi e^{(\mu_n + z^n)} n^{1+\varphi} - \frac{\vartheta_w - 1}{\vartheta_w} \mathcal{U}\left(c^h, c^s\right) wn\right] + \beta E_X\left[\log\left(\frac{1+(\pi^w)'}{1+\pi^*_w}\right)\right] - \log\left(\frac{1+\pi^w}{1+\pi^*_w}\right) = 0 \tag{79}$$

[Wage-Price Inflation]
$$(1+\pi)\frac{w}{w_{-1}} - (1+\pi^w) = 0 \tag{80}$$

[Labor Demand]
$$w - mc\,(1-\alpha)\,Ae^{(\mu_a+z^a)}\left(k^d\right)^{\alpha}\left(n^d\right)^{-\alpha} = 0 \tag{81}$$

[Capital Demand]
$$r^k - mc\alpha Ae^{(\mu_a+z^a)}\left(k^d\right)^{\alpha-1}\left(n^d\right)^{1-\alpha} = 0 \tag{82}$$

[Production]
$$y - Ae^{(\mu_a+z^a)}\left(k^d\right)^{\alpha}\left(n^d\right)^{1-\alpha} \tag{83}$$

[Price PC]
$$\kappa_p\left(mc - \frac{\vartheta_p - 1}{\vartheta_p}\right) + \beta E_X\left[\log\left(\frac{1+\pi'}{1+\pi^*}\right)\frac{y'}{y}\right] - \log\left(\frac{1+\pi}{1+\pi^*}\right) = 0 \tag{84}$$

[Dividend]
$$y\left(1 - \frac{\vartheta_p}{2\kappa_p}\left[\log\left(\frac{1+\pi}{1+\pi^*}\right)\right]^2\right) - wn^d - r^k k^d - D = 0 \tag{85}$$

[Gov. Budget Constraint]
$$D + b^s - \frac{1+r_{-1}}{1+\pi}b^s_{-1} - T = 0 \tag{86}$$

[Taylor Rule]
$$\max\left\{1, (1+r_{-1})^{\rho_r}\left[(1+r^*)\left(\frac{1+\pi}{1+\pi^*}\right)^{\phi_\pi}\left(\frac{y}{y^*}\right)^{\phi_y}\right]^{1-\rho_r} e^{\mu_m+z^m}\right\} - (1+r) = 0 \tag{87}$$

[Labor Market Clearing]
$$n^d - n = 0 \tag{88}$$

[Bond Market Clearing]
$$b^s - b = 0 \tag{89}$$

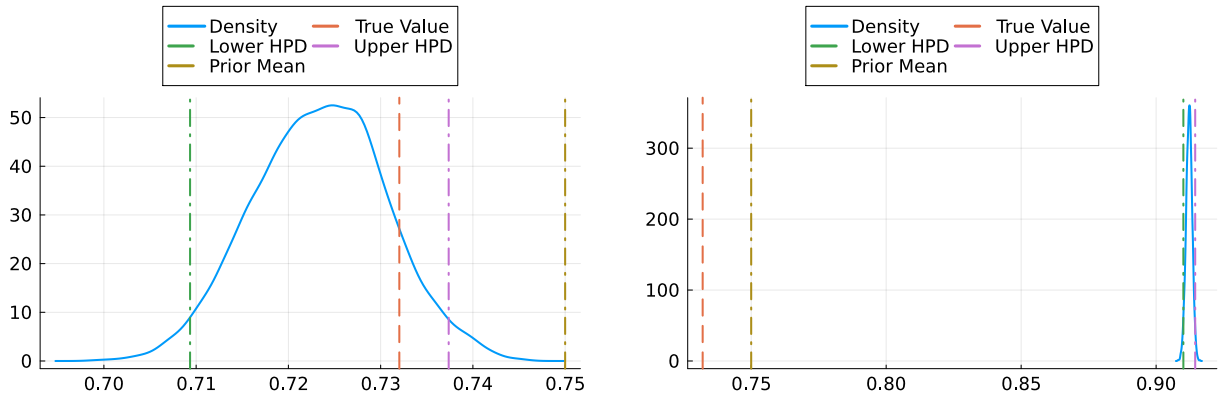
[Capital Market Clearing]
$$k^d - k_{-1} = 0 \tag{90}$$

[Goods Market Clearing]
$$\omega_h c^h + \omega_s\left(c^s + i^k\right) - y\left(1 - \frac{\vartheta_p}{2\kappa_p}\left[\log\left(\frac{1+\pi}{1+\pi^*}\right)\right]^2\right) = 0 \tag{91}$$

A.2 Additional Figures

This appendix presents the posterior distributions of the parameters estimated in Section 6.2. Figure 8 through Figure 23 show the posterior densities for the different parameters. In each figure, the subfigure on the left corresponds to the posterior for the methodology where the initial state density is approximated

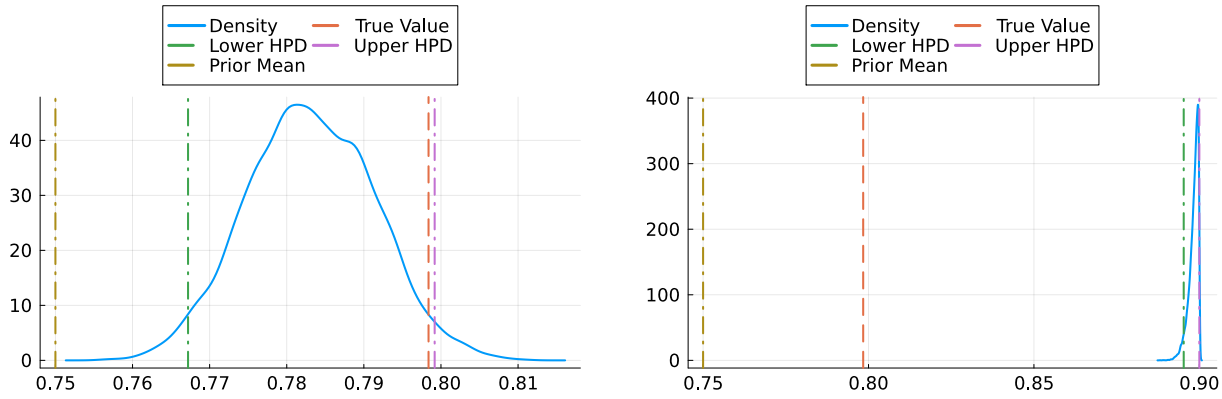
with a mixture density network. The subfigure on the right corresponds to the posterior for the methodology where the initial states are set to their steady-state values. Figure 24 plots the posterior distributions of the untransformed initial states. Note that these posteriors exist only for the method that uses the mixture density network. The vertical gold line in each figure is the prior mean. The vertical orange line is the true value. The vertical green line is the lower bound of the 95% high posterior density interval. The vertical purple line is the upper bound of the 95% high posterior density interval.



(a) Mixture Density Network

(b) Steady-State

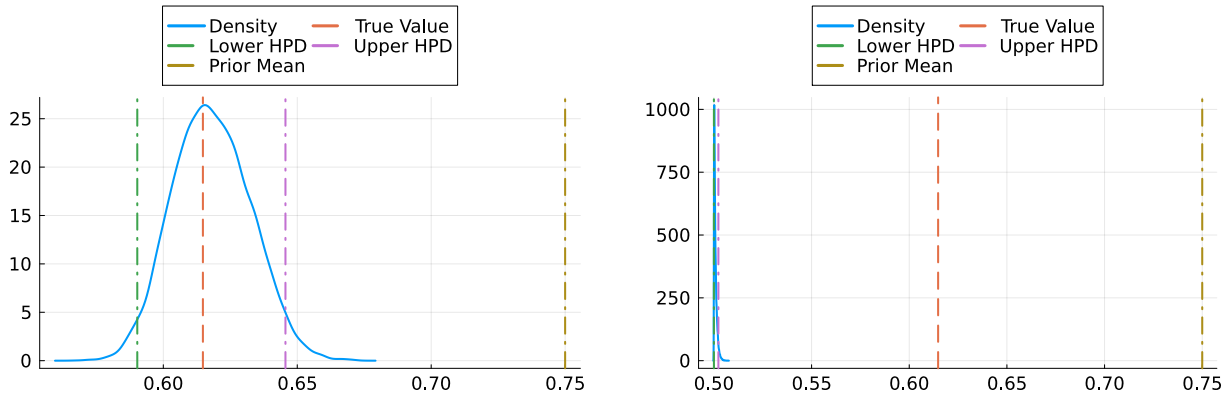
Figure 8: Posterior distributions of ρ_a.



(a) Mixture Density Network

(b) Steady-State

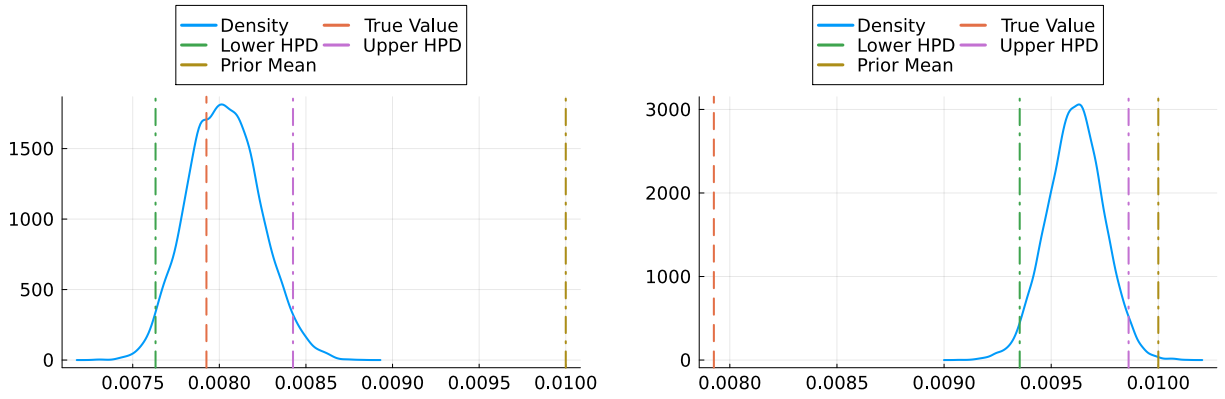
Figure 9: Posterior distributions of ρ_m.



(a) Mixture Density Network

(b) Steady-State

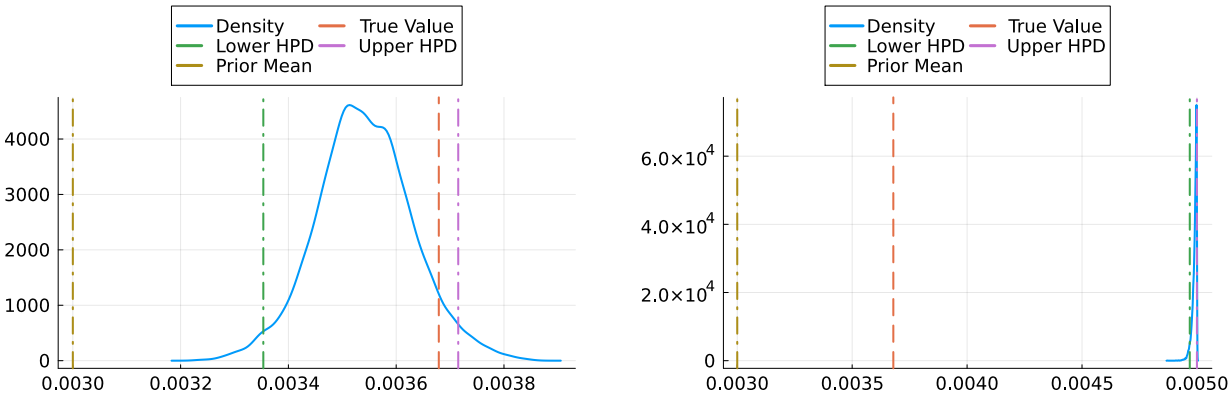
Figure 10: Posterior distributions of ρ_n.



(a) Mixture Density Network

(b) Steady-State

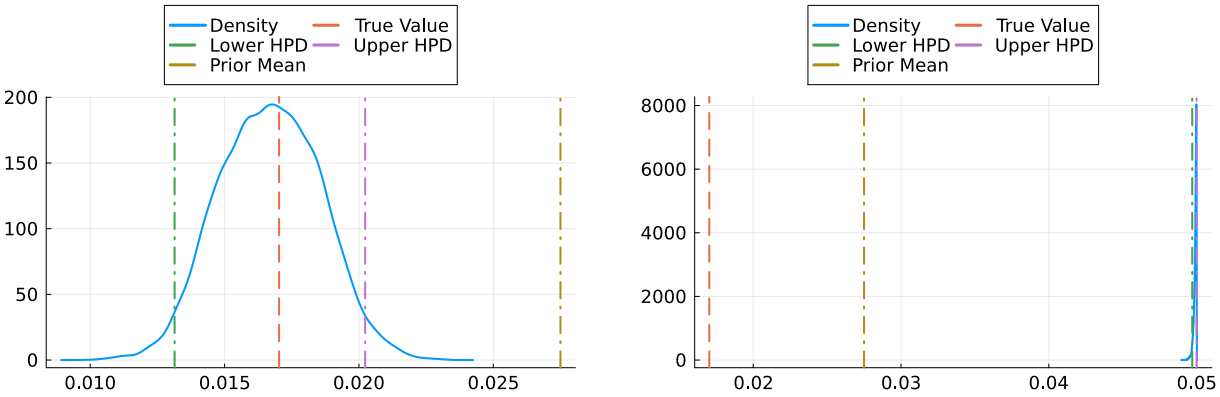
Figure 11: Posterior distributions of σ_a.



(a) Mixture Density Network

(b) Steady-State

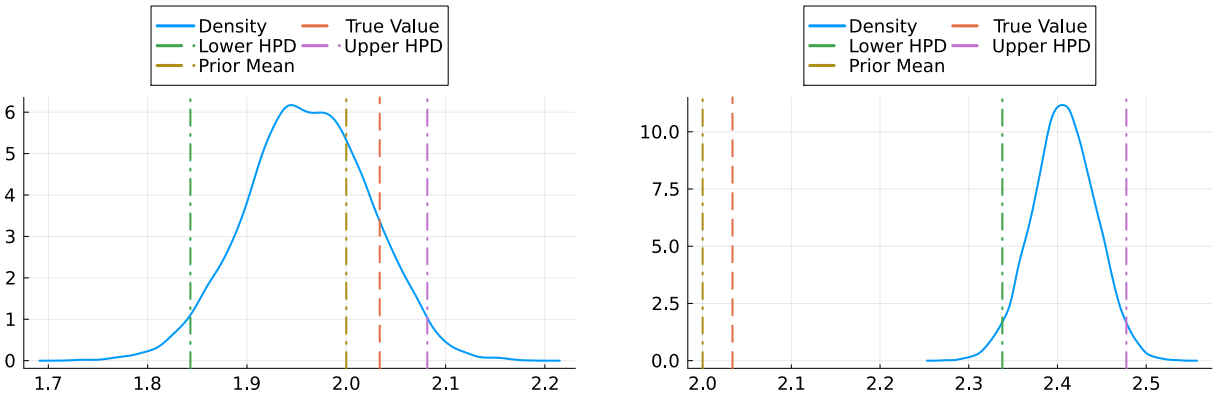
Figure 12: Posterior distributions of σ_m.



(a) Mixture Density Network

(b) Steady-State

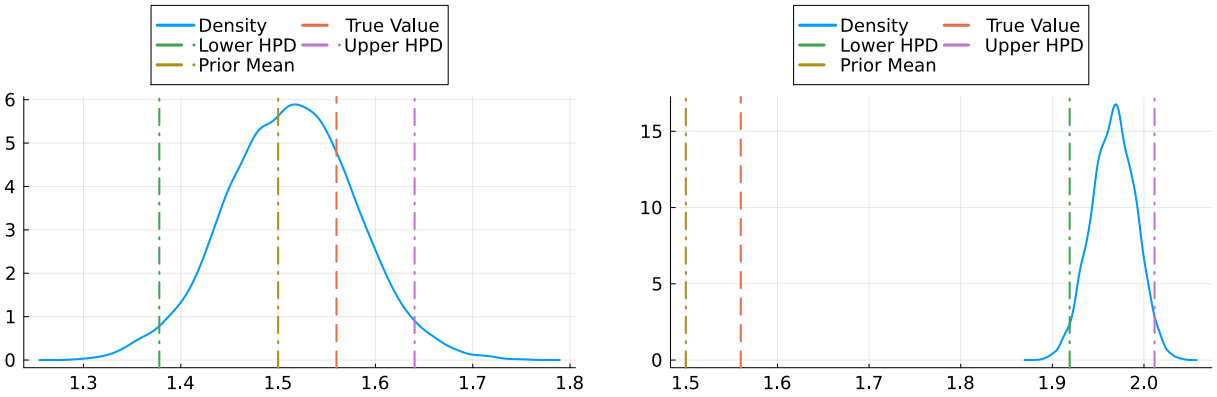
Figure 13: Posterior distributions of σ_n.



(a) Mixture Density Network

(b) Steady-State

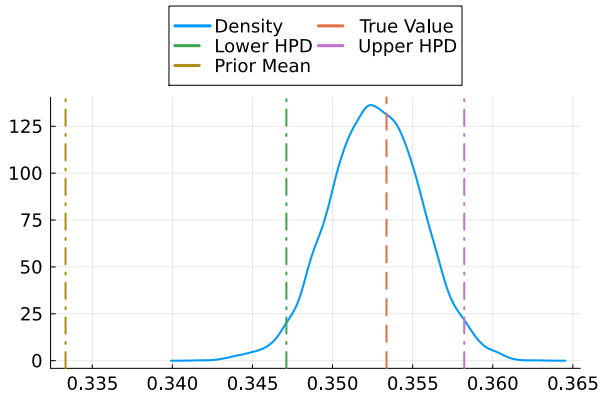
Figure 14: Posterior distributions of η.



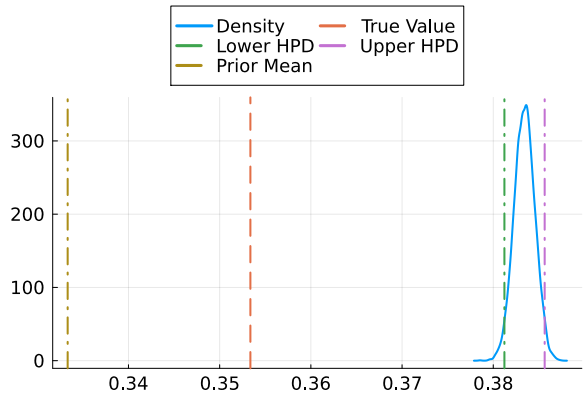
(a) Mixture Density Network

(b) Steady-State

Figure 15: Posterior distributions of φ.

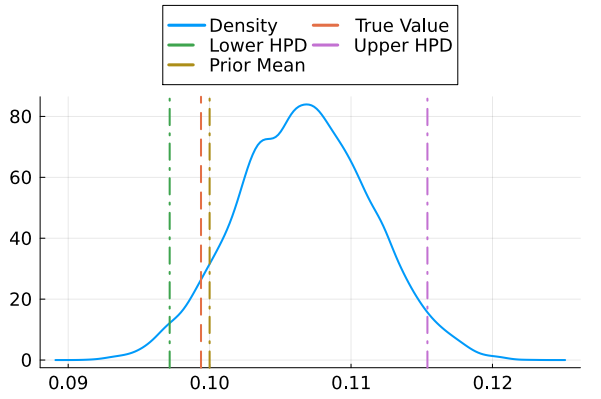


(a) Mixture Density Network

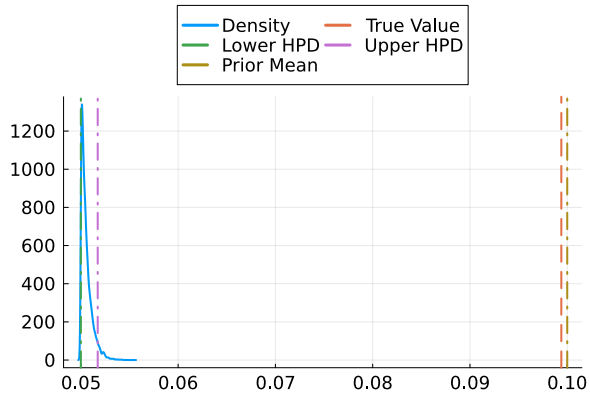


(b) Steady-State

Figure 16: Posterior distributions of α.

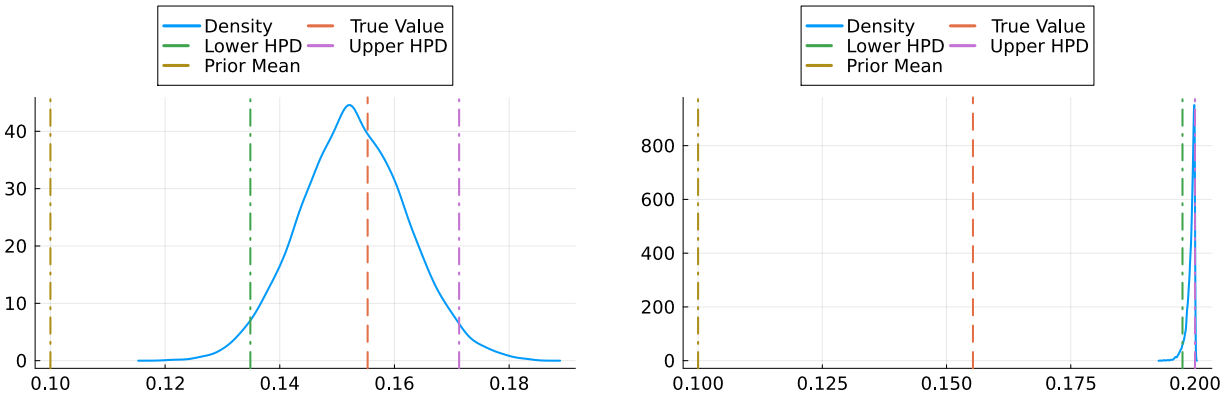


(a) Mixture Density Network



(b) Steady-State

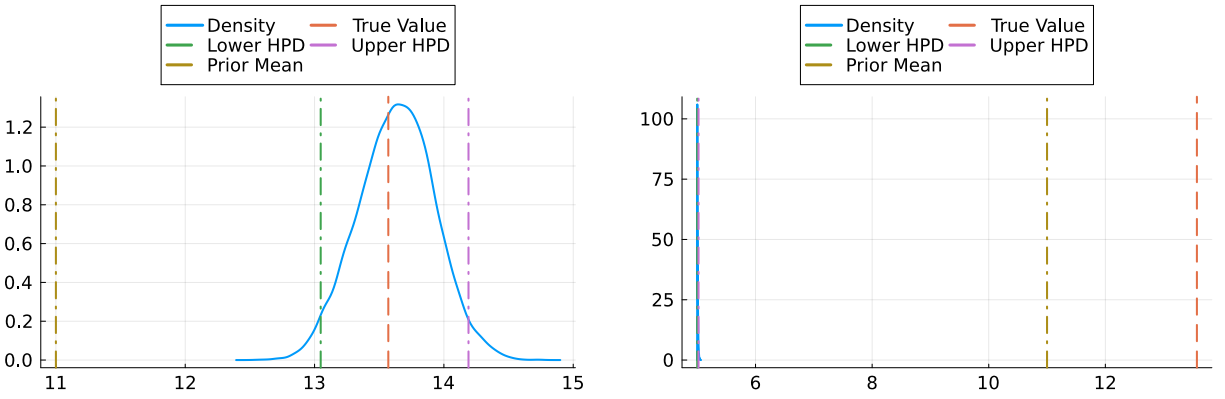
Figure 17: Posterior distributions of κ_p.



(a) Mixture Density Network

(b) Steady-State

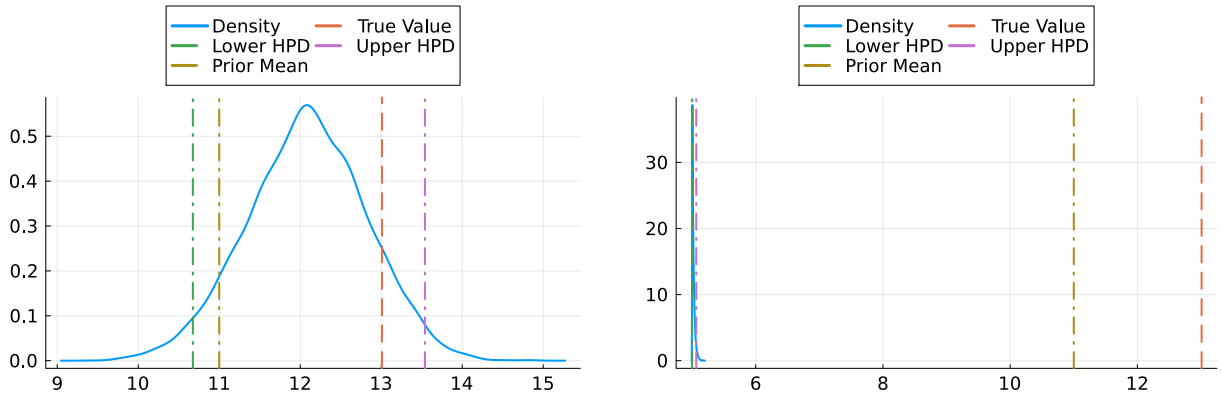
Figure 18: Posterior distributions of κ_w.



(a) Mixture Density Network

(b) Steady-State

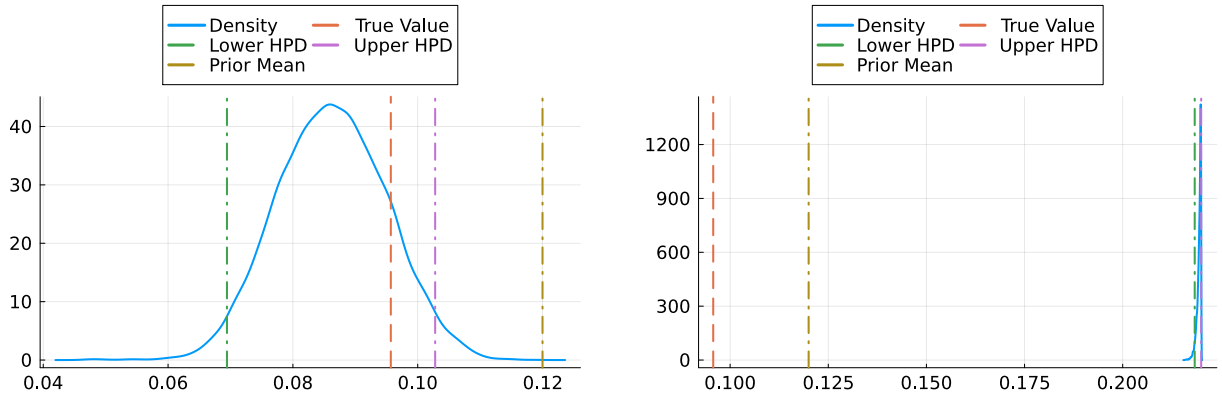
Figure 19: Posterior distributions of ϑ_p.



(a) Mixture Density Network

(b) Steady-State

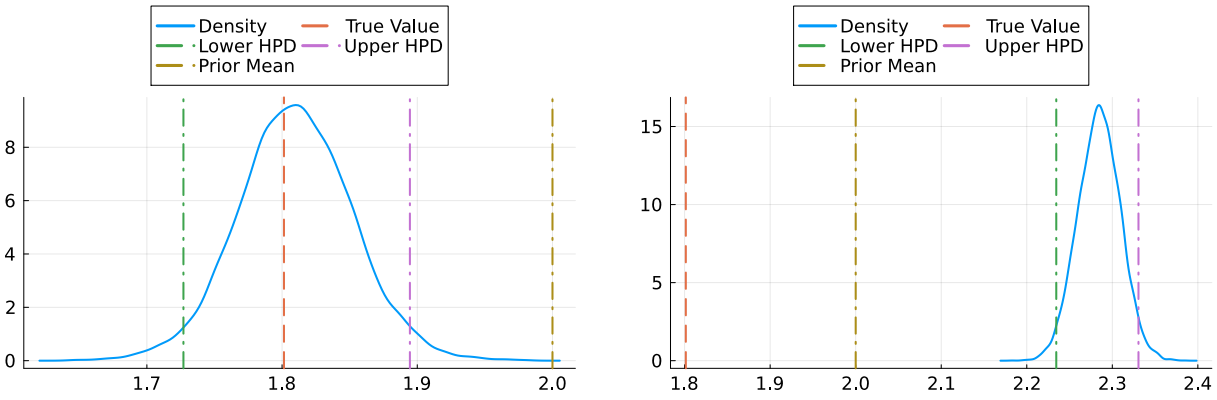
Figure 20: Posterior distributions of ϑ_w.



(a) Mixture Density Network

(b) Steady-State

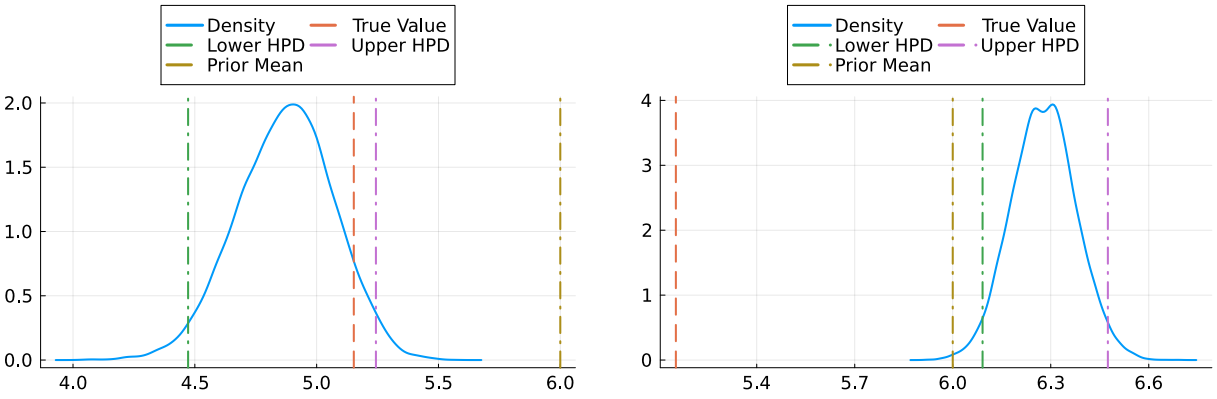
Figure 21: Posterior distributions of ϕ_y.



(a) Mixture Density Network

(b) Steady-State

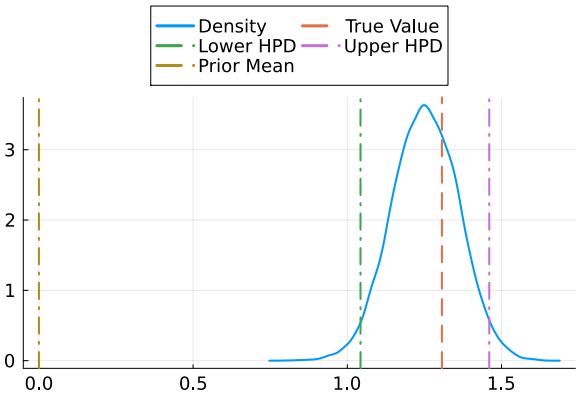
Figure 22: Posterior distributions of ϕ_π.



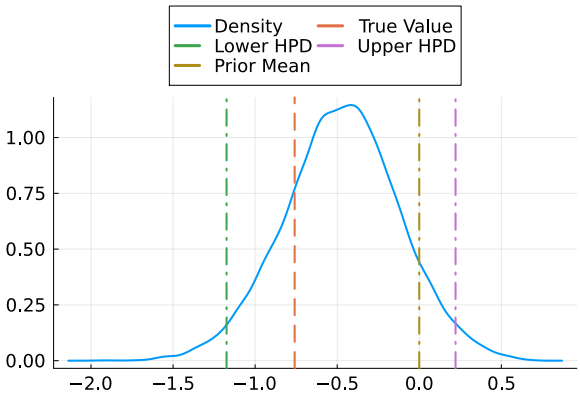
(a) Mixture Density Network

(b) Steady-State

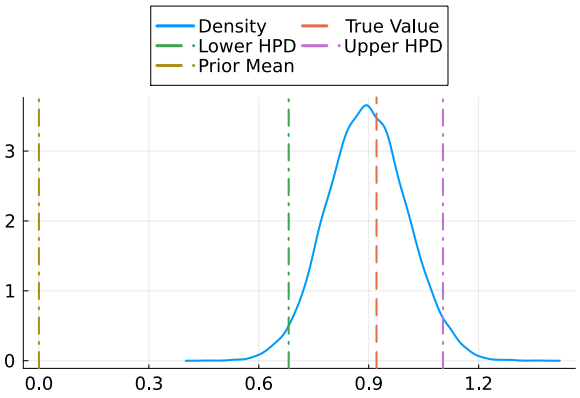
Figure 23: Posterior distributions of κ_{i^k}.



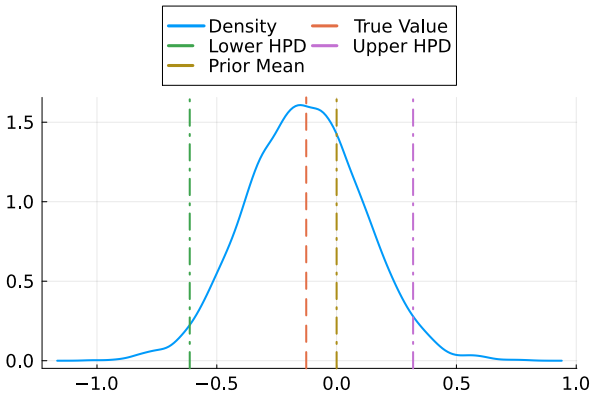
(a) Posterior distribution of $\widetilde{z}_0^a$



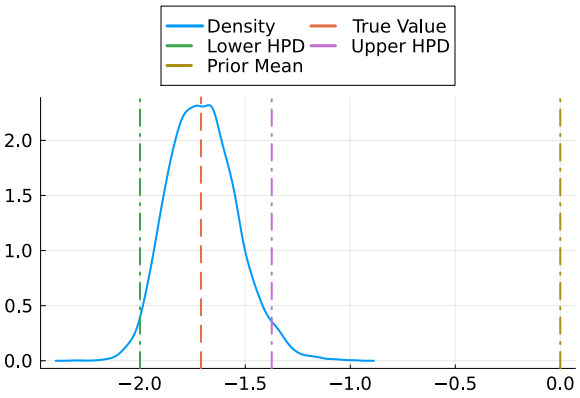
(b) Posterior distribution of $\widetilde{z}_0^m$



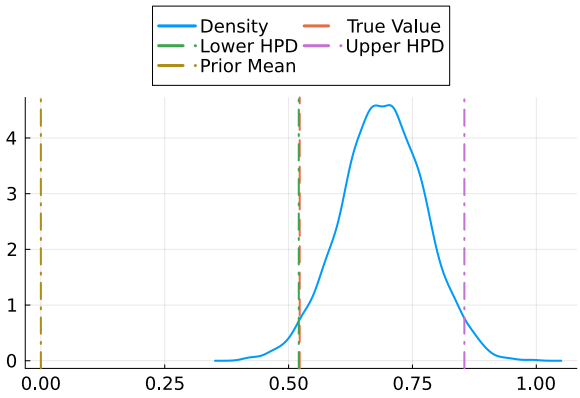
(c) Posterior distribution of $\widetilde{z}_0^n$



(d) Posterior distribution of $\widetilde{k}_0$



(e) Posterior distribution of $\widetilde{i}_0^k$



(f) Posterior distribution of $\widetilde{w}_0$

Figure 24: Posterior distributions of the initial states.