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

Christopher Naubert
Bank of Canada

October 22, 2024

[a]The views expressed here are those of the author and not necessarily those of the Bank of Canada.

Motivation

- Globally solved, non-linear models
  - Inequality constraints: zero lower bound, irreversible investment
  - State-dependent effects: increasing policy rate when house prices are high vs. low
- Challenges of Bayesian estimation of non-linear models
  - Need to repeatedly evaluate posterior $\rightarrow$ need policies solved for many parameter configurations
  - Non-linear state space model $\rightarrow$ cannot use Kalman filter to compute likelihood
  - Non-differentiable target density $\rightarrow$ cannot use gradient information to guide sampling (Hamiltonian Monte Carlo)
- **This paper:**
  - Avoid non-linear filter
  - Avoid repeated solving of policies
  - Create differentiable target density

Overview of Results

- Avoid non-linear filter: joint likelihood (Childers et al. 2022)
- Avoid repeated solving of policies: extended neural network (Kase, Melosi, and Rottner 2022)
- Create differentiable target density: approximate initial state density with mixture density network (MDN) (Bishop 1994)
- Application:
  - Medium scale two-agent New-Keynesian model, ZLB, irreversible investment, three aggregate shocks & 17 estimated parameters
- **Estimation Results on Simulated Data**:
  - Initial states drawn from MDN: $\frac{17}{17}$ parameters in 95% HDIs
  - Initial states equal to steady state: $\frac{0}{17}$ parameters in 95% HDIs contain
- Incorporate deep RL training strategies to improve training stability

Related Literature

Framework

Economic Models

- Collection of Markov decision problems
- States: x
- Economic parameters: u
- Optimal policy: Γ_i^*
- Agent i's problem

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

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

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

- **This paper:** first order conditions are necessary and sufficient

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

Joint Likelihood (Childers et al. 2022)

- u: economic parameters
- z^T: history of observations
- x^T: history of states
- ϵ^T: history of innovations

$$\underbrace{p\left(u|z^T\right)}_{\textbf{posterior}} = \int p\left(u, x^T|z^T\right) dx^T = \int\int \underbrace{p\left(u, \epsilon^T, x_0|z^T\right)}_{\textbf{joint likelihood}} d\epsilon^T dx_0$$

- Sample from $p\left(u, \epsilon^T, x_0|z^T\right)$ using MCMC and discard $\left\{\epsilon^T, x_0\right\}$

Joint Likelihood (Childers et al. 2022)

- Evaluating $p\left(u, \epsilon^T, x_0 | z^T\right)$

$$\ln p\left(u, \epsilon^T, x_0 | z^T\right) = \underbrace{\ln p(u)}_{\textbf{prior}} + \sum_{t=1}^{T} \underbrace{\ln p\left(z_t | \epsilon^t, x_0, u\right)}_{\textbf{measurement equations+states LOM}} + \sum_{t=1}^{T} \underbrace{\ln p\left(\epsilon_t | u\right)}_{\textbf{innovations}} + \underbrace{\ln p\left(x_0 | u\right)}_{\textbf{initial states}} + C$$

- Avoid non-linear filter ✓

Measurement Equations

- State at time t: x_t
  - Obtained from initial states (x_0) and history of innovations (ϵ^T)
- Observation i at time t: $z_{i,t}$
- Economic parameters: u
- True optimal policy for control i: $\Gamma_i^*(\cdot,\cdot)$
- Normally distributed measurement error

$$z_{i,t} = \Gamma_i^*(x_t, u) + \sigma_{i,me}\varepsilon_{i,me,t} \qquad \varepsilon_{i,me,t} \sim \mathcal{N}(0,1)$$

$$\Longrightarrow$$

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

Policy Function Approximation

- Residual function: $\mathfrak{R}[\cdot]$
- Equilibrium

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

- In practice

$$\Gamma_i^*(\cdot,\cdot) \approx \Gamma_i(\cdot,\cdot;\gamma_i)$$

- Projection approach (Kase, Melosi, and Rottner 2022)

$$\{\gamma_i\} = \underset{\{\hat{\gamma}_i\}}{\operatorname{argmin}} \int_{\mathrm{U}} \int_{\mathrm{X}(u)} ||\mathfrak{R}\left[\left\{\Gamma_i(x,u;\hat{\gamma}_i)\right\}\right]|| d\mu(x)\, d\nu(u) \tag{4}$$

- Only one set of approximation function parameters
- Avoid repeated solving of policies ✓
- $\Gamma_i(x,u;\gamma_i)$ chosen to be differentiable wrt. u

Policy Training

- Lillicrap et al. (2016) (Deep Deterministic Policy Gradient)
  - Asset Euler equation

$$\mathcal{U}'(c) - \beta E\left[\mathcal{U}'(c')(1 + r')\,|x, u\right] = 0 \tag{5}$$

  - Separate weights for current and future choices

$$c = \Gamma_c\left(x, u; \gamma^{\mathbb{C}}\right) \tag{6}$$
$$c' = \Gamma_c\left(x', u; \gamma^{\mathbb{F}}\right) \tag{7}$$

  - Slowly update weights for future choices

$$\gamma_j^{\mathbb{C}} = \texttt{GradientDescent}\left(\nabla_{\gamma^{\mathbb{C}}}\mathfrak{R}, \gamma_{j-1}^{\mathbb{C}}\right) \tag{8}$$
$$\boldsymbol{\gamma_j^{\mathbb{F}}} = \tau^{\gamma}\boldsymbol{\gamma_j^{\mathbb{C}}} + (1 - \tau^{\gamma})\,\boldsymbol{\gamma_{j-1}^{\mathbb{F}}} \tag{9}$$

- Use trained policies, parameters, initial states and innovations to compute $p\left(z_t|\epsilon^t, x_0, u\right)$

Eq. Errors

Initial State Distribution

- Stationary distribution

$$\ln p\left(x_0|u\right) = \ln p\left(x_0^{endo}|x_0^{exo}, u\right) + \ln p\left(x_0^{exo}|u\right) \tag{10}$$

- Approximate density of initial **endogenous** states with mixture density network (Bishop 1994)

$$\{\zeta_{\mu_i}, \zeta_{\Sigma_i}, \zeta_{\pi_i}\}_{i=1}^{N_{mix}} \leftarrow \Lambda\left(x_0^{exo}, u; \lambda\right) \tag{11}$$

- Approximate density

$$p\left(x_0^{endo}|x_0^{exo}, u\right) = \sum_{i=1}^{N_{mix}} \zeta_{\pi_i} \mathcal{N}\left(x_0^{endo}|\zeta_{\mu_i}, \zeta_{\Sigma_i}\right) \tag{12}$$

- $\Lambda\left(x_0^{exo}, u; \lambda\right)$ is differentiable wrt. u
- Create differentiable target density ✓

Multiple Components

MDN Training

- Expectations maximization (Delong, Lindholm, and Wüthrich 2021)
- Posterior probabilities

$$\wp_k^{j-1}\left(x_0^{endo}\right) = \frac{\zeta_{\pi_k}^{j-1}\mathcal{N}\left(x_0^{endo}|\zeta_{\mu_k}^{j-1}, \zeta_{\Sigma_k}^{j-1}\right)}{\sum_{i=1}^{N_{mix}} \zeta_{\pi_i}^{j-1}\mathcal{N}\left(x_0^{endo}|\zeta_{\mu_i}^{j-1}, \zeta_{\Sigma_i}^{j-1}\right)} \tag{13}$$

- Objective function

$$Q(\lambda) = \sum_{n=1}^{N_{batch}} \sum_{i=1}^{N_{mix}} \wp_i^{j-1}\left(x_{n,0}^{endo}\right)\left[\log\left(\zeta_{\pi_i}\right) + \log\left(\mathcal{N}\left(x_{n,0}^{endo}|\zeta_{\mu_i}, \zeta_{\Sigma_i}\right)\right)\right] \tag{14}$$

- Network updates

$$\lambda_j = \texttt{GradientDescent}\left(-\nabla_\lambda Q, \lambda_{j-1}\right) \tag{15}$$

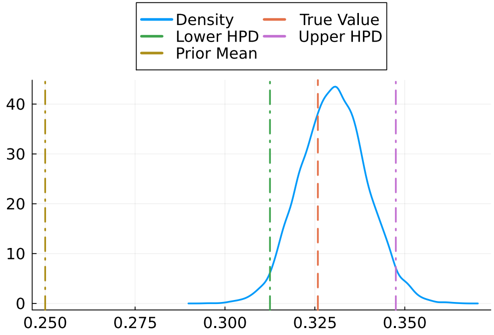
Quantitative Results

Exercise Setup

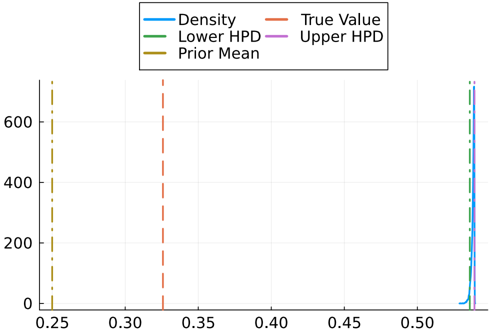
- Model
  - Two-agent New Keynesian model
  - Irreversible capital investment subject to adjustment cost
  - Zero lower bound on nominal rates
  - Labor dis-utility, monetary policy and total factor productivity shocks
  - Four endogenous states, three exogenous states
  - 17 estimated parameters
- Estimation
  - Simulated data
  - No-U-Turn Sampler (Hoffman, Gelman, et al. 2014)
  - Measurement error: 5% $\times$ (stdev. observable)
  - Number of samples: 10,000 with 10,000 warm-up ($<$10 hours)
  - MDN: initial states drawn from the mixture density network
  - SS: initial states equal to steady state values

Estimation Results

- MDN vs. steady state
- Posterior distributions of β_{draw} $(\beta = \frac{1}{\frac{\beta_{draw}}{100}+1})$



(a) Mixture Density Network



(b) Steady-State

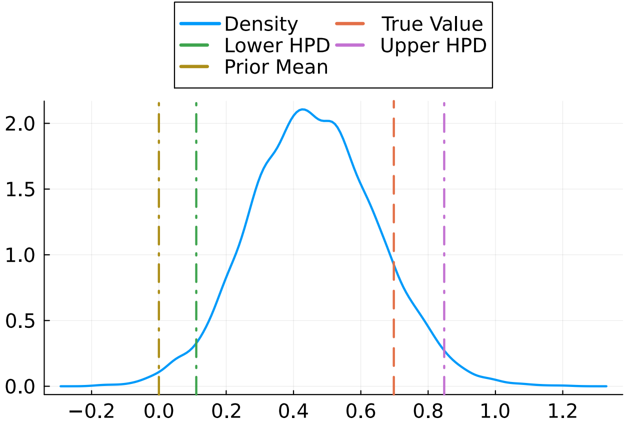
Estimation Results

- 95% HDI coverage: ✓ (yes) or ✗ (no)?

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_n	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_{i^k}	5.1525	6.00	4.8711	[4.4724,5.2433]	✓	6.2787	[6.0913,6.4748]	✗

Estimation Results

- Initial r vs. steady state r: 1.34% vs 0.82%
- Posterior distribution of $\widetilde{r}_0$



Conclusion

- Use mixture density network to approximate distribution of initial states
  - Differentiable $\rightarrow$ compatible with HMC/NUTS
  - Recover "true" parameters even when initial states are far from steady state
- Incorporate neural network training strategies from deep RL to improve training
  - Separate weights for current and future choices
  - Slowly update weights for future choices

Thank you!
Questions?

Appendix

Related Literature

- Estimation of DSGE models
  - Kase, Melosi, and Rottner (2022)
    - Global, non-linear solutions
    - Compute likelihood on sample from parameter space using particle filter. Fit neural network to likelihood values.
  - Childers et al. (2022)
    - Local first and second-order perturbation solutions
    - Use joint likelihood to avoid filter
- Neural Networks:
  - Fernández-Villaverde et al. (2020)
  - L. Maliar, S. Maliar, and Winant (2021);
  - Azinovic, Gaegauf, and Scheidegger (2022)
  - Kase, Melosi, and Rottner (2022)
  - Han, Yang, and E (2022)

Motivation

Equilibrium Condition Errors

- log 10 absolute errors
- Simulation length: 1,000 periods
- Batches of pseudo-states: 128

	Mean	Median	Max	Mean Max	Median Max
HtM BC	-2.3583	-2.2632	-1.1358	-1.5232	-1.5302
Saver BC	-2.1987	-2.1012	-1.0313	-1.3844	-1.3776
EE Bond	-3.0818	-2.9910	-1.6704	-2.2029	-2.2131
EE Capital	-3.3474	-3.2610	-1.9240	-2.3724	-2.3824
EE Invst.	-2.5242	-2.4363	-1.2345	-1.6337	-1.6391
Resource Constraint	-2.3633	-2.2668	-1.1459	-1.5457	-1.5388
Price PC	-3.3549	-3.2820	-1.9049	-2.4987	-2.4824
Wage PC	-2.6817	-2.6046	-1.3767	-1.8673	-1.8497

Policy Training

Multiple Components

$$\ln p\left(u, \epsilon^T, x_0 | z^T\right) = \ln p(u) + \sum_{t=1}^{T} \ln p(\epsilon_t | u) + \ln\left(\sum_{i=1}^{N_{mix}} \exp\left(\ln(\zeta_{\pi_i}) + \sum_{t=1}^{T} \ln p\left(z_t | \zeta_{\mu_i}, \zeta_{\Sigma_i}, \epsilon^t, x_0, u\right)\right)\right) + \ln p\left(x_0^{endo} | x_0^{exo}, u\right) + \ln p\left(x_0^{exo} | u\right) + C \quad (16)$$

MDN

NK Model Priors

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_n	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]

Estimation Results

Estimation Results

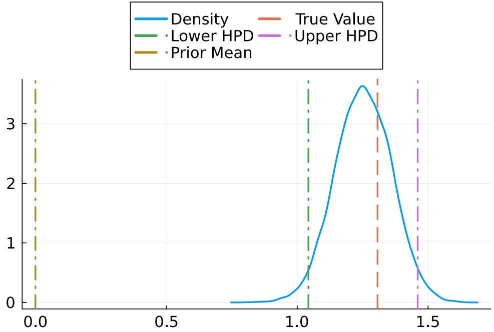
- Bulk-ESS as percentage of total samples

Parameter	Bulk-ESS %
ρ_a	2.0345
ρ_m	1.0967
ρ_n	2.1649
σ_a	1.4391
σ_m	0.6460
σ_n	2.2119
β_{draw}	9.6773
η	1.6478
φ	4.2173

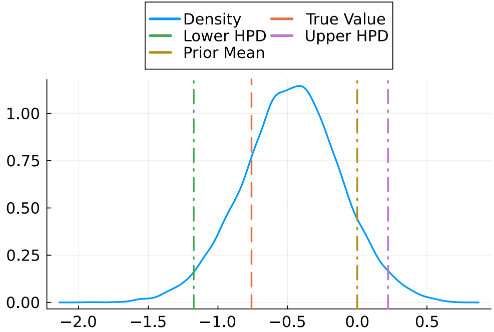
Parameter	Bulk-ESS %
α	4.0907
κ_p	1.7432
κ_w	4.3625
ϑ_p	4.0286
ϑ_w	2.3221
ϕ_y	3.5782
ϕ_π	3.1840
κ_{i^k}	1.8047

HPD

Estimation Results: Initial States



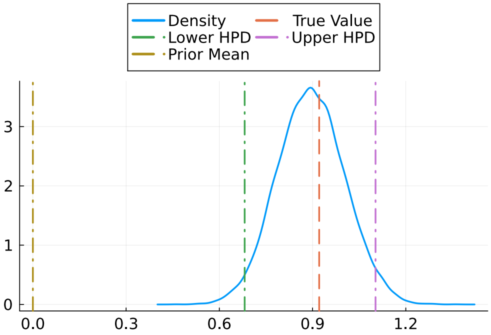
(a) Posterior distribution of $\widetilde{z}_0^a$



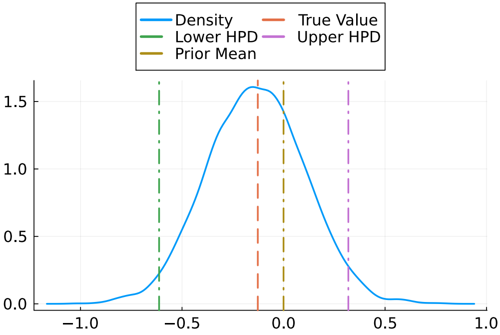
(b) Posterior distribution of $\widetilde{z}_0^m$

$\widetilde{r}_0$

Estimation Results: Initial States



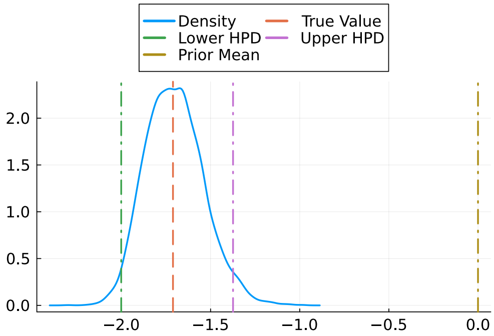
(a) Posterior distribution of $\widetilde{z}_0^n$



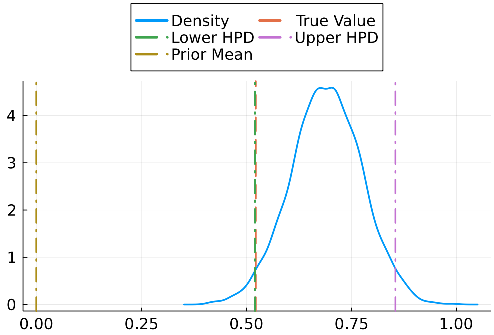
(b) Posterior distribution of $\widetilde{k}_0$

$\widetilde{r}_0$

Estimation Results: Initial States



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



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

Figure 4: Posterior distributions of the initial states

$\widetilde{r}_0$

References

- Azinovic, Marlon, Luca Gaegauf, and Simon Scheidegger (Nov. 2022). "Deep Equilibrium Nets". In: *International Economic Review* 63.4, pp. 1471–1525. DOI: 10.1111/iere.12575. URL: https://ideas.repec.org/a/wly/iecrev/v63y2022i4p1471-1525.html.
- Bishop, Christopher M. (1994). "Mixture density networks". In: URL: https://api.semanticscholar.org/CorpusID:118227751.
- Childers, David et al. (Oct. 2022). *Differentiable State-Space Models and Hamiltonian Monte Carlo Estimation*. Working Paper 30573. National Bureau of Economic Research. DOI: 10.3386/w30573. URL: 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". In: *Insurance: Mathematics and Economics* 101, pp. 240–261. ISSN: 0167-6687. DOI: https://doi.org/10.1016/j.insmatheco.2021.08.003. URL: https://www.sciencedirect.com/science/article/pii/S0167668721001232.

Fernández-Villaverde, Jesús et al. (2020). "Solving High-Dimensional Dynamic Programming Problems using Deep Learning". In: URL: https://api.semanticscholar.org/CorpusID:229330871.

Han, Jiequn, Yucheng Yang, and Weinan E (2022). *DeepHAM: A Global Solution Method for Heterogeneous Agent Models with Aggregate Shocks*. arXiv: 2112.14377 [econ.GN]. URL: https://arxiv.org/abs/2112.14377.

- Hoffman, Matthew D, Andrew Gelman, et al. (2014). "The No-U-Turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo.". In: *J. Mach. Learn. Res.* 15.1, pp. 1593–1623.
- Kase, Hanno, Leonardo Melosi, and Matthias Rottner (June 2022). *Estimating Nonlinear Heterogeneous Agents Models with Neural Networks*. CEPR Discussion Papers 17391. C.E.P.R. Discussion Papers. URL: https://ideas.repec.org/p/cpr/ceprdp/17391.html.
- Lillicrap, Timothy P. et al. (2016). "Continuous control with deep reinforcement learning.". In: *ICLR*. Ed. by Yoshua Bengio and Yann LeCun. URL: http://dblp.uni-trier.de/db/conf/iclr/iclr2016.html#LillicrapHPHETS15.
- Maliar, Lilia, Serguei Maliar, and Pablo Winant (2021). "Deep learning for solving dynamic economic models". In: *Journal of Monetary Economics* 122.C, pp. 76–101. DOI: 10.1016/j.jmoneco.2021.07. URL: https://ideas.repec.org/a/eee/moneco/v122y2021icp76-101.html.