Operator Learning in Macroeconomics

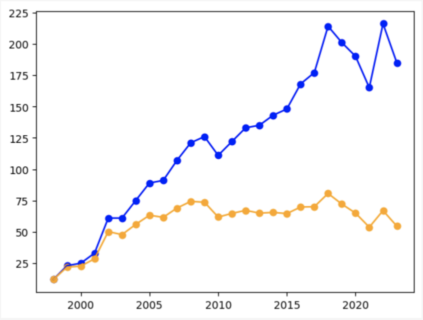
Yaolang Zhong

University of Warwick

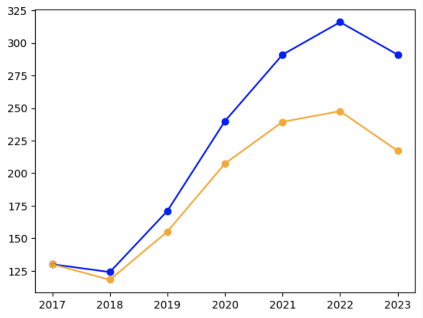
Motivation

Dynamic models incorporating both heterogeneity and aggregate uncertainty have become one of the key areas of focus in macroeconomics

Figure 1: Year vs. Incremental Citation Count per Year (Blue Line)
Assumed 5% Annual Growth Rate Detrend (Orange Line)



Krusell and Smith (1998): 1998-2023



The HANK model (Kaplan, Moll and Violante, 2018): 2017-2023

Motivation

- Some reasons for popularity
  - Empirical Justification: The implication of heterogeneity for aggregate behaviors (Blundell, Pistaferri and Preston, 2008; Krueger and Perri, 2006)
  - Realistic Representation: Richer dynamics and more complex interactions (Cagetti and De Nardi, 2008)
- However, the computation of heterogeneous agent models is still challenging, reflecting the inherent complexities of the models:
  - High Dimensionality: Numerous variables to capture diversity
  - Non-Linear Dynamics: For example, saving decisions of hand-to-mouth households
  - Policy Analysis: Reliant on simulations

This Paper

- Focuses on the numerical solution of a specific type of heterogeneous agent model with aggregate shocks:
  - Discrete time, infinite horizon, and a continuum of agents
  - Key feature of the model: The agents' state variables include not only their individual state vectors but also the cross-sectional distribution of all agents' individual states, an infinite-dimensional object
  - Intuition: Certain variables (e.g., prices) and their dynamics depend on the aggregated distribution
- Proposes a novel numerical method that is generally applicable and computationally efficient for globally solving these models

This Paper (cont.)

- Considers a general case: policy function $k_i' = \mathbf{g}(k_i, \mathbf{\Gamma})$
  - Where k_i is the individual's capital holding, and $\mathbf{\Gamma}$ is the distribution function of k
- Novel contributions in three aspects:
  - Formulation of the Problem: Reformulate the agents' policy function (more precisely, functional) as a "policy operator" (mapping between function spaces)
    - $\mathbf{g}(k_i, \mathbf{\Gamma}) = \mathbf{G}(\mathbf{\Gamma})(k_i)$
  - Numerical Approximation: Parameterize the policy operator using the neural operator, an advanced neural network architecture from machine learning literature
    - $\mathbf{G}_\theta(\mathbf{\Gamma})(k_i)$
  - Implementation Algorithm: Design an optimization scheme to facilitate convergence (not covered in this talk)
    - $\theta^* = \arg\min |\mathbf{G}_\theta(\mathbf{\Gamma})(k_i) - \mathbf{G}(\mathbf{\Gamma})(k_i)|$, where $\mathbf{\Gamma} \in \mathcal{T}, k_i \in [k_{\text{min}}, k_{\text{max}}]$

Preview

- Two current frameworks in the literature:
  - The Krusell-Smith Algorithm (KS): Computationally efficient but less generalizable (Krusell and Smith, 1998; Maliar, Maliar and Valli, 2010).
  - Deep Learning with Feed-Forward Neural Network (NN): Generally applicable but slower (Maliar, Maliar and Winant, 2021; Han, Yang et al., 2021)
- The operator framework addresses their issues through both operator formulation and operator parameterization
- Experiments on a Bewley-Huggett-Aiyagari model with aggregate uncertainty:
  - KS framework: Approximately 5 minutes, error (relative Euler residual) at the level of 0.1%
  - NN framework: Error remains high even after 30 minutes.
  - Operator framework: Approximately 10 minutes, error level between 0.1% and 1%

Model Setup

- The first benchmark model in the computational suite project for the comparison of the properties of numerical algorithms (Den Haan, Judd and Juillard, 2011)
- Lowercase letters for individual variables, uppercase letters for aggregate variables, and bold letters for operations
- A continuum of infinitely lived and ex-ante identical agents. Agent i in period t:
  - Receives a fixed time endowment $\bar{\ell}$
  - Earns the after-tax wage $(1-\tau_t)\bar{\ell}W_t$ if employed $(\epsilon_t^i = 1)$
  - Earns the unemployment benefit μW_t if unemployed $(\epsilon_t^i = 0)$
  - W_t is the per unit of time wage rate, τ_t is the tax rate, and μ is a model parameter denoting the fraction of the wage for subsidy

Model Setup (cont.)

- Market is incomplete: non-zero capital holding $k_t^i \geqslant 0$
- The net rate of return for capital: $R_t - \delta$, where R_t is the market-determined interest rate and δ is the fixed depreciation rate
- Agents' maximization problem:

$$\mathbb{E} \sum_{t=0}^{\infty} \beta^t \frac{(c_t^i)^{1-\gamma} - 1}{1-\gamma}$$

subject to:

$$c_t^i + k_{t+1}^i = \underbrace{R_t k_t^i}_{\text{capital gain}} + \left[\underbrace{(1-\tau_t)\,\bar{\ell}\epsilon_t^i}_{\text{labor income}} + \underbrace{\mu\left(1-\epsilon_t^i\right)}_{\text{unemployment subsidy}} \right] W_t + (1-\delta) k_t^i$$

Model Setup (cont.)

- **Firms**: Use a Cobb-Douglas production function $Y_t = Z_t K_t^{\alpha}(\bar{\ell} L_t)^{1-\alpha}$
- K_t is the per capita capital, L_t is the employment rate, and $\alpha \in [0, 1]$ is the capital share. Z_t is a binary aggregate productivity shock: $Z_t \in \{Z_b, Z_g\}$
- **Government**: Maintains a balanced budget by redistributing all tax revenue
- The system of prices is determined by firms' first-order optimality and government's budget constraint:

$$R_t = \alpha Z_t \left(\frac{K_t}{\bar{\ell} L_t}\right)^{\alpha-1}, \quad W_t = (1-\alpha) Z_t \left(\frac{K_t}{\bar{\ell} L_t}\right)^{\alpha}, \quad \tau_t = \frac{\mu(1-L_t)}{\bar{\ell} L_t} \tag{1}$$

- Agents' decision-making processes are influenced by the current levels of aggregate variables (K_t, L_t, Z_t) as well as their dynamics

Model Setup (cont.)

- **Shocks**: Z_t is first-order Markovian. ϵ_t^i is first-order Markovian conditional on the transition of Z_t and conforms to the law of large numbers
- $(\epsilon_t^i, Z_t) \sim \mathbf{\Pi}$: The element $\pi_{\epsilon\epsilon' ZZ'}$ denotes $\mathsf{P}[(\epsilon_t^i, Z_t) \rightarrow (\epsilon_{t+1}^i, Z_{t+1})]$
- To write down the recursive form:
  - Agents' individual state vector (k_t^i, ϵ_t^i)
  - $\mathbf{\Pi}$ is calibrated such that the employment rate L_t is a function of Z_t: $L_t \in \{L_b, L_g\}$. Agents do not need to know the distribution of ϵ_t^i for levels and motions of L_t
  - $K_t = \int k_t^i \mathbf{f}(k_t^i) \mathrm{d}k$, implying a requirement for the knowledge of $\mathbf{f}(k_t^i)$
  - The motion of K_t is more subtle: consider $k_{t+1}^i = \mathbf{g}(k_t^i, \epsilon_t^i)$. Then, $K_{t+1} = \int \mathbf{g}(k_t^i, \epsilon_t^i) \mathbf{f}(k_t^i, \epsilon_t^i) \mathrm{d}k \mathrm{d}\epsilon$

Incomplete Information Assumption

- A rational expectation equilibrium requires that agents observe $(k_t^i, \epsilon_t^i, Z_t, \mathbf{f}(k_t^i, \epsilon_t^i))$
- Incomplete information assumption for simplicity: agents observe only $\mathbf{f}(k_t^i)$, or equivalently, $\mathbf{f}(k_t^i, \epsilon_t^i) = \mathbf{f}(k_t^i)\mathbf{f}(\epsilon_t^i)$:
  - To be consistent with the implementation of the KS framework (Maliar, Maliar and Valli, 2010), replacing $\mathbf{f}(k_t^i, \epsilon_t^i)$ with K_t
  - ϵ_t^i is binary: $\mathbf{f}(k_t^i, \epsilon_t^i) = \begin{cases} \mathbf{f}(k_t^i, 0) & \text{if } \epsilon_t^i = 0 \\ \mathbf{f}(k_t^i, 1) & \text{if } \epsilon_t^i = 1 \end{cases}$, resulting in two continuous one-dimensional functions of k_t^i. This assumption assists the discussion to focus on $\mathbf{f}(k_t^i)$
  - Note that the proposed framework can effortlessly generalize to the case of continuous shocks where $\mathbf{f}(k_t^i, \epsilon_t^i)$ is then a continuous two-dimensional function

The Recursive Form

- Denote $\mathbf{\Gamma}$ as the representation of the distribution of agents over capital k
- Denote the law of motion of $\mathbf{\Gamma}$ by $\mathbf{H} : \mathbf{\Gamma}' = \mathbf{H}(\mathbf{\Gamma}, Z, Z')$
- The agents' problem can therefore be expressed recursively as

$$\mathbf{V}(k_i, \epsilon_i; Z, \mathbf{\Gamma}) = \max_{k_i'} \left\{ \mathbf{U}(c_i) + \beta \mathbb{E} \left[\mathbf{V}(k_i', \epsilon_i'; Z', \mathbf{\Gamma}') \mid \epsilon_i, Z \right] \right\} \tag{2}$$

subject to

$$\begin{aligned} c_i + k_i' &= R k_i + \left[(1-\tau)\bar{\ell}\epsilon_i + \mu(1-\epsilon_i) \right] W + (1-\delta) k_i, & (3) \\ \epsilon_i', Z' &\sim \mathbf{\Pi}(\epsilon_i, Z), & (4) \\ \mathbf{\Gamma}' &= \mathbf{H}(\mathbf{\Gamma}, Z, Z'), & (5) \\ k_i' &\geqslant 0 & (6) \end{aligned}$$

- Denote the solution to (2) subject to (3), (4), (5), and (6) as $\mathbf{V}^*(\cdot)$ and the corresponding policy function as $\mathbf{g}^*(\cdot)$

Literature: The KS Framework

- We are interested in the recursive policy function: $k_i' = \mathbf{g}^*(k_i, \epsilon_i, Z, \mathbf{\Gamma})$
- (Krusell and Smith, 1998):

$$\mathbf{g}_{KS}(k_i, \epsilon_i, Z, \mathbf{m})$$

  where $\mathbf{m} = (m_1, m_2, \ldots, m_L)$ is a vector of moments
- In implementation:
  - Manage a cross-section of N simulated agents $(k_1, k_2, \ldots, k_N)$
  - $\mathbf{m} \equiv K = \frac{1}{N}\sum_{i=1}^{N} k_i$
- Pros: Tractable, intuitive, and fast
- Cons: Incomplete information of the distribution

Literature: The NN Framework

- Deep Learning with feed-forward neural network (Maliar, Maliar and Winant, 2021):

$$\mathbf{g}_{NN}(k_i, \epsilon_i, Z, (k_1, k_2, ..., k_N))$$

- $\mathbf{\Gamma}$ is represented by a "plug-in" vector $(k_1, k_2, ..., k_N)$
- Feed-forward neural network to overcome the curse-of-dimensionality (Goodfellow et al., 2016)

This Paper: The Operator Framework

- Reformulate the policy function as the policy operator:

$$\mathbf{g}(k_i, \epsilon_i, Z, \mathbf{\Gamma}) := \mathbf{G}(\mathbf{\Gamma})(k_i, \epsilon_i, Z) = \mathbf{G}(\mathbf{\Gamma})(k_i \mid \epsilon_i, Z) \tag{7}$$

- Two-step decomposition of processing $(k_i, \epsilon_i, Z, \mathbf{\Gamma})$:
    - Input $\mathbf{\Gamma}$ to an operator $\mathbf{G}$ for a conditional policy function $\mathbf{G}(\mathbf{\Gamma})$
    - Input (k_i, ϵ_i, Z) to $\mathbf{G}(\mathbf{\Gamma})$ for $k_i' = \mathbf{G}(\mathbf{\Gamma})(k_i \mid \epsilon_i, Z)$
- Represent $\mathbf{\Gamma}$ by the cumulative distribution function (CDF)
- Parameterize the operator $\mathbf{G}$ by the neural operator $\mathbf{G}_\theta$
- The superiority of this framework is driven by three properties:
    - Sharing-Aggregation, Permutation-Invariance, Discretization-Invariance

Sharing-Aggregation

- Consider $\mathbf{g}_{NN}(k_i, \epsilon_i, Z, (k_1, k_2, \ldots, k_N))$
- In simulation, we process each agent's $(k_i, \epsilon_i, Z, (k_1, k_2, \ldots, k_N))$ to determine policy $\mathbf{g}_{NN}$ for k_i'
- The computational cost is $\mathcal{O}(N^2)$
- However, this approach does not utilize the information that agents share the same aggregation

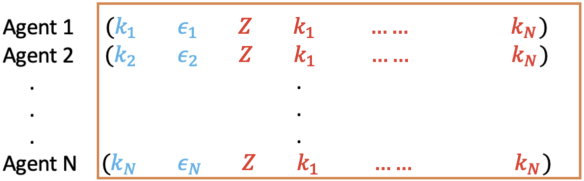


Figure 2: Illustration of the Computational Complexity

Sharing-Aggregation (cont.)

- In the neural operator formulation $\mathbf{G}(\mathbf{\Gamma})(k_i \mid \epsilon_i, Z)$, we only need to process the distribution function part once
- The computational cost is therefore $\mathcal{O}(N)$

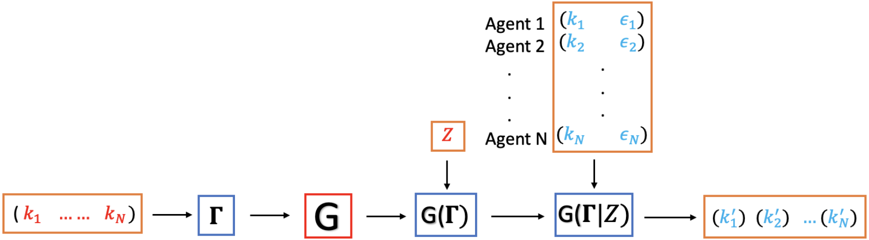


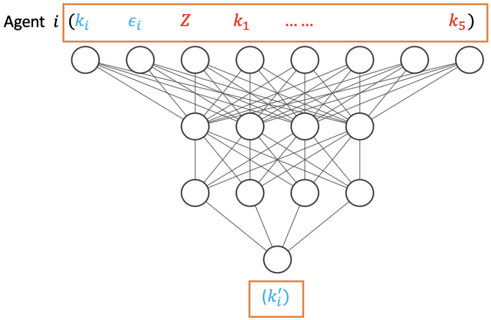
Figure 3: Neural Operator Process Illustration

Permutation-Invariance

- Consider $k_i' = \mathbf{g}_{NN}(k_i, \epsilon_i, Z, (k_1, k_2, \ldots, k_N))$
- k_i' should be invariant to the ordering of $(k_1, k_2, \ldots, k_N)$. For example, $(k_1 = a, k_2 = b, \ldots, k_N)$ and $(k_1 = b, k_2 = a, \ldots, k_N)$ should yield the same k_i' for a fixed (k_i, ϵ_i, Z).
- Simulated data and training time required to learn this pattern are extensive
- In the operator framework, $(k_1, k_2, \ldots, k_N)$ is used to construct an empirical CDF $\hat{\mathbf{\Gamma}}$ with sorted values, resulting in invariance to ordering

Discretization-Invariance

- Revisiting $\mathbf{g}_{NN}(k_i, \epsilon_i, Z, (k_1, k_2, \ldots, k_N))$.
- There's a trade-off: a larger N provides a better approximation of the continuous distribution $\mathbf{\Gamma}$ but increases the complexity of $\mathbf{g}_{NN}$
- Additional issue: the approach may not be applicable to varying N



Discretization-Invariance (cont.)

- In the operator framework, the operator $\mathbf{G}$ is parameterized by the neural operator $\mathbf{G}_\theta$, specifically, the Fourier neural operator as per (Li et al., 2020).
- The size of the neural operator $\mathbf{G}_\theta$ is invariant, regardless of the discretization of input and output functions.
- $\mathbf{G}_\theta$ essentially consists of a sequence of convolutions parameterized in the Fourier domain.

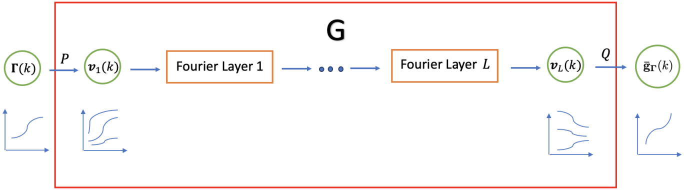


Figure 5: Fourier Neural Operator Architecture

Discretization-Invariance (cont.)

- The discretization in the spatial domain does not impact the parameterization in the Fourier domain.

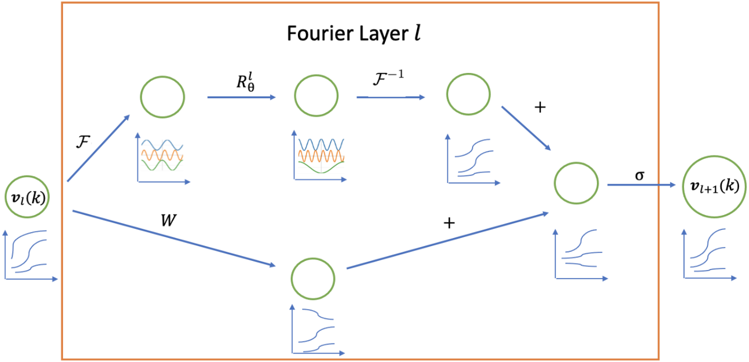


Figure 6: Illustration of the Fourier Neural Operator Layer

Discretization-Invariance: an example

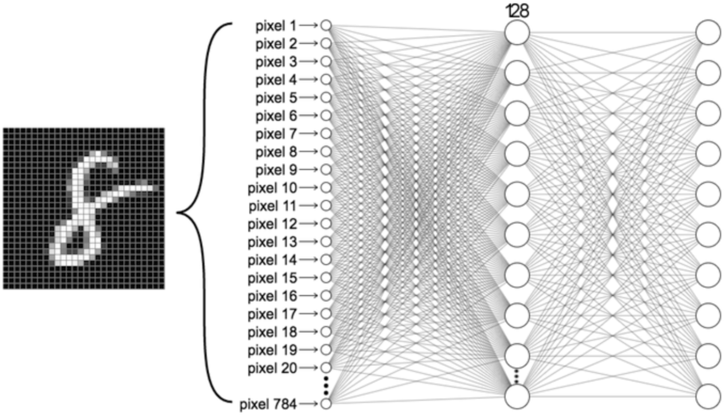


Figure 7: Handwriting Recognition as an Example

The Neural Operator Framework: Summary

Table 1: Comparison of Three Numerical Frameworks for the Desirable Properties

Property	Framework		
	KS[1]	NN[2]	Operator[3]
Full Information of Distribution	$\times$	✓	✓
Discretization-Invariance	✓	$\times$	✓
Permutation-Invariance	✓	$\times$	✓
Sharing-Aggregation	✓	$\times$	✓

[1] Krusell-Smith
[2] Deep Learning with feed-forward neural network
[3] Deep Learning with neural operator (This Paper)

Implementation: The Objective Function

- The unique solution that solves the Bellman equation must satisfy the derived Euler equation in the absence of a borrowing constraint:

$$\frac{\mathrm{d}u}{\mathrm{d}c}(c) = \beta\mathbb{E}[(1-\delta+R')\frac{\mathrm{d}u}{\mathrm{d}c}(c')] \tag{8}$$

- For a given state $(k, \epsilon, Z, \mathbf{\Gamma})$ and a neural operator parameterized policy $k' = \mathbf{g}_\theta(k, \epsilon, Z, \mathbf{\Gamma})$, define the unit-free Lagrange multiplier:

$$h \equiv 1 - \frac{\beta\mathbb{E}[(1-\delta+R')\frac{\mathrm{d}u}{\mathrm{d}c}(\text{wealth}' - k'')]}{\frac{\mathrm{d}u}{\mathrm{d}c}(\text{wealth} - k')} \tag{9}$$

where wealth $= \mathbf{M}(k, \epsilon, Z, \mathbf{\Gamma})$ is agents' total budget

Implementation: The Objective Function (cont.)

- Agents' optimality can be expressed in terms of the Kuhn-Tucker conditions:

$$h \geqslant 0, \quad k' \geqslant 0, \quad hk' = 0 \tag{10}$$

- Apply the Fischer-Burmeister (FB) transformation to make the Kuhn-Tucker conditions differentiable:

$$\Psi^{FB}(k', h) = k' + h - \sqrt{k'^2 + h^2} = 0 \tag{11}$$

with $a = k'$ and $b = h$.

- The objective function for a particular state $\omega = (k, \epsilon, Z, \mathbf{\Gamma})$ is:

$$\xi(\omega, \theta) \equiv \|\Psi^{FB}(k', h)\|^2 \tag{12}$$

Implementation: The Objective Function (cont.)

- In iteration ℓ, suppose there is a set of collected states $\{\omega : \omega \in \Omega^\ell\}$, then the objective function is:

$$\Xi^\ell(\theta) = \frac{1}{|\Omega^\ell|} \sum_{\omega \in \Omega^\ell} \xi(\omega, \theta) \tag{13}$$

- Parameters in the neural operator are updated using the gradient descent method:

$$\theta^{\ell+1} \leftarrow \theta^\ell - \lambda^\ell \nabla_\theta \Xi^\ell(\theta^\ell) \tag{14}$$

Results

- Benchmark case of the neural operator
- In around 10 minutes, the optimization loss reaches a level of 10^{-4} to 10^{-5} (corresponding to 0.1% to 1% relative Euler error)

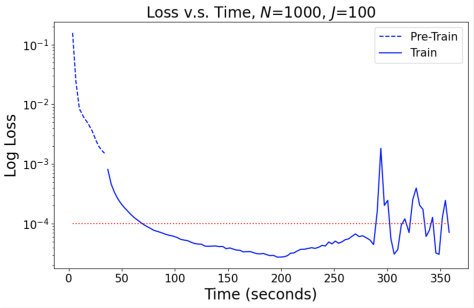


Figure 8: Training Losses vs. Time (seconds)

Results: Operator vs. NN

- The optimization in the NN framework remains at a high loss level at around 30 minutes

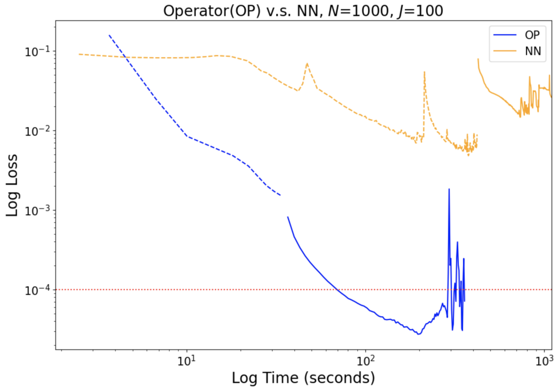


Figure 9: Comparison of the Operator Framework and the NN Framework

Results: Operator vs. KS

- To visualize how the operator framework solution compares to that of KS
- Select a random period with the simulated distribution $\mathbf{\Gamma}$ as input
- Compare the conditional function $\mathbf{g}_\theta(k, \epsilon, Z|\mathbf{\Gamma})$ to the policy $g_{KS}(k, \epsilon, Z, K)$.
- The converged relative Euler loss of g_{KS} is around 0.1%

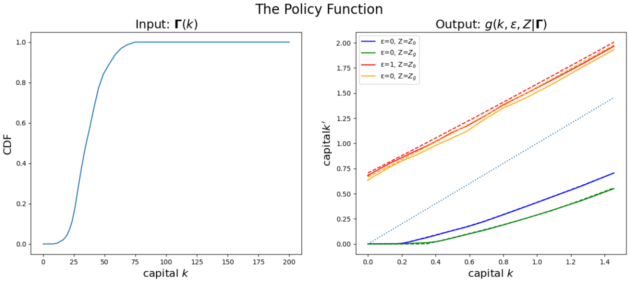


Figure 10: An Instance of Conditional Policy Function

Results: Operator vs. KS

- To visualize the similarity of the operator framework solution to that of KS
- Simulate the economy using both $\mathbf{g}_\theta$ and $\mathbf{g}_{KS}$ with the same initialization

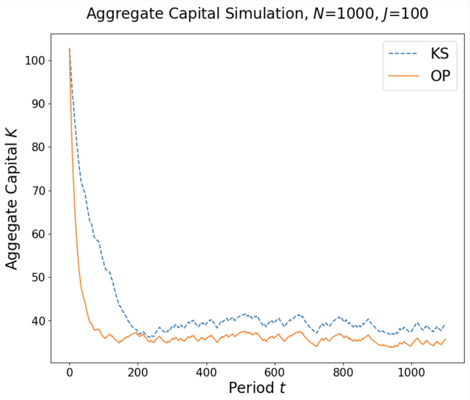


Figure 11: The Simulated Aggregate Capitals

Conclusion

- This paper presents a novel approach to solving heterogeneous agents models with aggregate shocks in a discrete time, infinite horizon, and continuum agent setting. The approach incorporates the cross-sectional distribution of all individual states as part of the agents' state variable and leverages neural operator learning
- Computational advancements are attributed to the sharing-aggregation and parameterization-invariance property of operator formulation, as well as the discretization-invariance property in the proposed parameterization
- An optimization scheme tailored for this problem is formulated to facilitate the convergence of training
- Experiments on a Bewley-Huggett-Aiyagari model with aggregate uncertainty demonstrate computational efficiency compared to contemporary frameworks

Blundell, Richard, Luigi Pistaferri, and Ian Preston, "Consumption inequality and partial insurance," *American Economic Review*, 2008, *98* (5), 1887–1921.

Cagetti, Marco and Mariacristina De Nardi, "Wealth inequality: Data and models," *Macroeconomic dynamics*, 2008, *12* (S2), 285–313.

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville, *Deep learning*, MIT press, 2016.

Haan, Wouter J Den, Kenneth L Judd, and Michel Juillard, "Computational suite of models with heterogeneous agents II: Multi-country real business cycle models," *Journal of Economic Dynamics and Control*, 2011, *35* (2), 175–177.

Han, Jiequn, Yucheng Yang et al., "Deepham: A global solution method for heterogeneous agent models with aggregate shocks," *arXiv preprint arXiv:2112.14377*, 2021.

Kaplan, Greg, Benjamin Moll, and Giovanni L Violante, "Monetary policy according to HANK," *American Economic Review*, 2018, *108* (3), 697–743.

Krueger, Dirk and Fabrizio Perri, "Does income inequality lead to consumption inequality? Evidence and theory," *The Review of Economic Studies*, 2006, *73* (1), 163–193.

Krusell, Per and Anthony A Smith Jr, "Income and wealth heterogeneity in the macroeconomy," *Journal of political Economy*, 1998, *106* (5), 867–896.

Li, Zongyi, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar, "Fourier neural operator for parametric partial differential equations," *arXiv preprint arXiv:2010.08895*, 2020.

Maliar, Lilia, Serguei Maliar, and Fernando Valli, "Solving the incomplete markets model with aggregate uncertainty using the Krusell–Smith algorithm," *Journal of Economic Dynamics and Control*, 2010, *34* (1), 42–49.

_ , _ , and Pablo Winant, "Deep learning for solving dynamic economic models.," *Journal of Monetary Economics*, 2021, *122*, 76–101.