Deep Equilibrium NETs

Marlon Azinovic, Luca Gaegauf, Simon Scheidegger

2025 年 4 月 1 日

汇报人：张培元

Motivation

Humans are all different – heterogeneous

- Heterogeneity a crucial ingredient in contemporary models:
  - to study e.g. cross-sectional consumption response to aggregate shocks.
  - to model, e.g., social security, taxation.
- Example OLG models:
  - How many age groups?
  - borrowing constraints?
  - aggregate shocks?
  - idiosyncratic shocks?
  - liquid / illiquid assets?
- Models: heterogeneous & high-dimensional

Traditional Methods

- Dynamic Stochastic Models

$$\mathbb{E}[E(x_t, x_{t+1}, p(x_t), p(x_{t+1}))|x_t, p(x_t)] = 0$$

$$x_{t+1} \sim P(\cdot|x_t, p(x_t))$$

  - x: point in state space; describes your system.
  - p: time-invariant policy function.
  - State-space potentially **irregularly-shaped** and **high-dimensional**.
- Traditional Numerical Methods:
  - Value-based Iteration
  - Policy-based Iteration
- Problems
  - N^d points in ordinary discretization schemes.
  - Curse of dimensionality
  - Usually: solve many non-linear systems of equations by invoking a solver

State Variables (Dimensions)	Points	Time to Solution
1	10	10 sec
2	100	$\sim$ 1.6 min
3	1,000	$\sim$ 16 mins
4	10,000	$\sim$ 2.7 hours
5	100,000	$\sim$ 1.1 days
6	1,000,000	$\sim$ 1.6 weeks
...	...	...
20	1e20	$\sim$ 3 trillion years
Optimizer		
Dimension reduction	Deal with Points	High-performance computing
Exploit symmetries, e.g., via the active subspace method see ECTA	e.g., via (Smolyak/adpative) sparse grids EGM	Reduces time to solution, but not the problem size MPI/OpenMP

表: What is high-dimensional?

Why use AI Method?

- Artificial intelligence (AI) has remarkable applications (recognition of images and speech, facilitation of computer vision, operation of self-driving cars)
- Most modern macroeconomic dynamics are based on a set of nonlinear dynamic systems and combined with various constraints to match the problems studied.
- AI is good at dealing with optimization problems.
- AI has accumulated rich experience and conducted extensive research in solving similar dynamic system frameworks in other fields (PINN).
- There is the possibility of isomorphism between them.

Literature Review

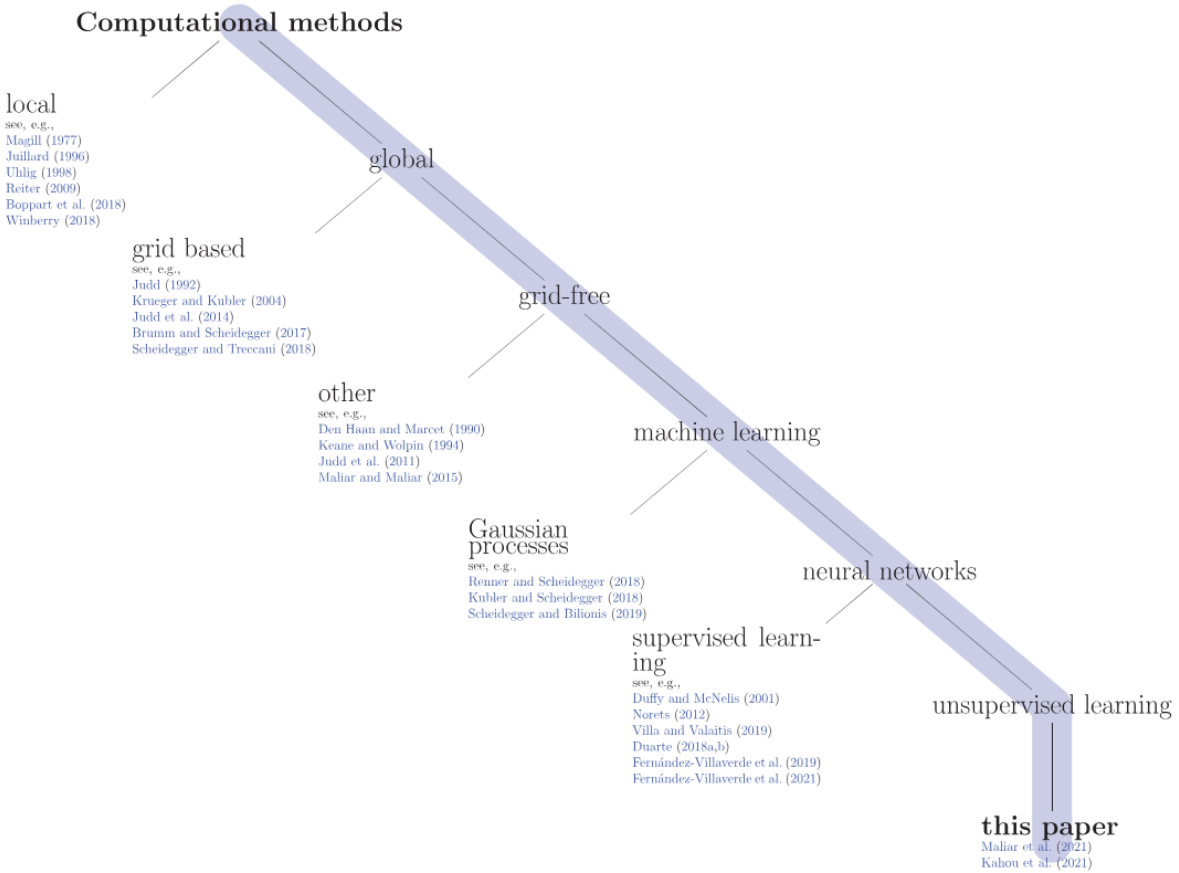


图: Literature Review

Unsupervised Learning

- Solving discrete-time models (Methodology)
  - Utilizing deep learning combined with a large amount of simulated data
  - Reference:
    - HAM - Maliar et al. (JME, 2021) (More general, infinty horizon, life-cycle)
    - OLG - Azinovic et al. (IER, 2022) (Specially on OLG, results matching)
    - Discrete choices - Maliar and Maliar (2022, JEDC)
- Solving continuous-time models (Mathematics)
  - HAM-DP to MFG-sys, and solving the BFSDEs.
    - BFSDEs = HJB + KF(KP)
    - Mean Field Games (2006, Jean-Michel Lasry and Pierre-Louis Lions; Huang-Malhame-Caines)
    - Income and Wealth Distribution in Macroeconomics: A Continuous-Time Approach (2022, RES, Yves Achdou, Jiequn Han, Jean-Michel Lasry, Pierre-Louis-Lions, Benjamin Moll)
  - Reference:
    - Jesus Fernandez-Villaverde(WP, 2020)
    - DeepBSDE - HAN et al. (PNAS, 2018) to DeepHAM (with Yang)
    - DeepFBSDE - Germain et al., 2022
    - FAME (JPE RR), GLME, DeepSAM.
- Performing estimation or calibration
  - Calibration, Estimation.

The Benchmark Model

Base settings

- Time is disceret: $t = 0, ..., \infty$
- Agents live for N periods ($N = 60$ years).
- One representative household per cohort.
- Every t, a representative household in born.
- No uncertainty about lifetime.
- There are exogenous aggregate shocks z that follow a Markov chain.

Household

- Household born in time t, will be given a shock as z^t and distinguished by it ($z^{t^{birth}}$).
- Household age $s = t - t^{birth} + 1 \leq N$
- Each period, the agents alive receive a strictly positive labor endowment l^s which depends on the age of the agent alone.
- Household supplies its labour endowment inelastically for a market wage $w(z^t) = w_t$.
- Agents alive maximize their remaining time-separable discounted expected lifetime utility ($\beta < 1$):

$$\sum_{i=0}^{N-s} E_t[\beta^i u(c_{t+i}^{s+i})]$$

- Households can save a unit of consumption good to obtain a unit of capital good next period (denoted as $a^s(z^t) = a_t^s$).
- The savings will become capital in the next period:

$$a_t^s = k_{t+1}^{s+1}, \forall t, \forall s \in \{1, ..., N-1\}$$

- There are two ways for households to transfer consumption over time: illiquid capital and one-period bond.
- Illiquid capital
  - with price r_t.
  - adjustment is costly

$$\Delta_t^s = a_t^s - k_t^s r_t$$

$$\frac{\zeta}{2}(\Delta_t^s)^2$$

  - with borrowing constrain

$$a_t^s \geq \underline{a}$$

- One-period bond
  - with price p_t
  - The bond is in zero net supply.
  - A bond promises a payoff of 1 in the subsequent period.
  - b_t^s denote the bond holding of household s in period t and let d_t^s denote the amount of bonds purchased by household.

$$b_{t+1}^{s+1} = d_t^s$$

  - No adjustment fee, household can buy(> 0) or sell(< 0) bonds.
  - Selling the bond is subject to collateral constraints:

$$\kappa d_t^s + a_t^s \geq 0$$

  - The bond is liquid in the sense that there are no costs associated with adjusting the bond holding

$$min\{\kappa r_t, 1\}$$

- The budget constrain of houshold s in period t is:

$$c_t^s + p_t d_t^s + a_t^s + \frac{\zeta}{2}(\Delta_t^s)^2 = l^s w_t + b_t^s + r_t k_t^s$$

- The households are born and die without any assets—that is

$$k_t^1 = b_t^1 = 0, \text{ and } a_t^N = d_t^N = 0$$

Frim

- There is a single representative firm with Cobb-Douglas production.
- The total factor productivity η (TFP) and the depreciation δ depend on the exogenous shock z.
- Each period, after the shock has been realized, the firm buys capital and hires labor to maximize its profits, taking prices as given. The production function is given by

$$f(K_t, L_t, z^t) = \eta_t K_t^{\alpha} L_t^{1-\alpha} + K_t(1 - \delta_t)$$

- The price of capital (r_t) and wages (w_t), as well as the price of the bond (p_t), are determined by market clearing in competitive spot markets.

Equilibrium

A competitive equilibrium, given initial conitions $z^0, k^{s\,N}_{0\,s=1}, b^{s\,N}_{0\,s=1}$, is a collection of choices for households $\{(c^s_t, a^s_t, d^s_t)^N_{s=1}\}^\infty_{t=0}$ and for the representative firm $(K_t, L_t)^\infty_{t=0}$ as well as prices $(r_t, w_t, p_t)^\infty_{t=0}$, such that

1. Given $(r_t, w_t)^\infty_{t=0}$, the choices $\{(c^s_t, a^s_t, d^s_t)^N_{s=1}\}^\infty_{t=0}$ maximize lifetime utility under s.t.s
2. Given r_t, w_t, the firm maximizes profits.
3. All market clear: For all t

$$L_t = \sum_{s=1}^{N} l^s_t, \quad K_t = \sum_{s=1}^{N} k^s_t, \quad 0 = \sum_{s=1}^{N} b^s_t$$

4. By FOCs:

$$w_t = \frac{\partial f(K_t, L_t, z^t)}{\partial L_t} = (1-\alpha)\eta_t K^\alpha_t L^{-\alpha}_t$$

$$r_t = \alpha\eta_t K^{\alpha-1}_t L^{1-\alpha}_t + (1-\delta_t)$$

More Conditions

1. The KKT conditions for any given generation of ages s at node z^t are given by:
2. 1. capital limitation

$$(1+\zeta\Delta_t^s)u'(c_t^s) = \beta E_t[u'(c_{t+1}^{s+1})r_{t+1}(1+\zeta\Delta_{t+1}^{s+1})] + \lambda_t^s + \mu_t^s$$

$$\lambda_t^s(a_t^s - \underline{a}) = 0$$

$$a_t^s - \underline{a} \geq 0$$

$$\lambda_t^s \geq 0$$

3. 2. bond limitation

$$p_t u'(c_t^s) = \beta E_t[u'(c_{t+1}^{s+1})] + \kappa\mu_t^s$$

$$\mu_t^s(a_t^s + \kappa d_t^s) = 0$$

$$a_t^s + \kappa d_t^s \geq 0$$

$$\mu_t^s \geq 0$$

Approximation of Equilibria with DNN

- Functional Rational Expectations Equilibrium (FREE) consists of equilibrium functions

$$\theta = [\theta_a^T, \theta_\lambda^T, \theta_d^T, \theta_\mu^T, \theta_p]^T : Z \times R^{2N} \to R^{4(N-1)+1}$$

- where
  - $\theta_a^T : Z \times R^{2N} \to R^{(N-1)}$ denotes the capital investment functions
  - $\theta_\lambda^T : Z \times R^{2N} \to R^{(N-1)}$ denotes the KKT conds on capital
  - $\theta_d^T : Z \times R^{2N} \to R^{(N-1)}$ denotes the bond investment functions
  - $\theta_\mu^T : Z \times R^{2N} \to R^{(N-1)}$ denotes the KKT conds on bond
  - $\theta_p : Z \times R^{2N} \to R$ denotes the bond price function

- For all states $\boldsymbol{x} := [z, \boldsymbol{k}^T, \boldsymbol{b}^T] \in Z \times R^{2N}$
- where exogenous shock ($z \in Z$), capital holding dist ($\boldsymbol{k} = [k_1, .., k_N]^T$), bond holding dist ($\boldsymbol{b} = [b_1, .., b_N]^T$)
- with $k_1 = 0, b_1 = 0$ for all $i = 1, .., N-1$:
  - capital limitation

$$(1 + \zeta \Delta_t^s(\boldsymbol{x})_i) u'(c(\boldsymbol{x})_i) = \beta E_z[u'(c(\boldsymbol{x}_+)_{i+1}) r(\boldsymbol{x}_+)_{i+1} (1 + \zeta \Delta(\boldsymbol{x}_+)_{i+1})] + \theta_\lambda(\boldsymbol{x})_i + \theta_\mu(\boldsymbol{x})_i$$

$$\theta_\lambda(\boldsymbol{x})_i \;\; \theta_a(\boldsymbol{x})_i = 0$$

$$\theta_a(\boldsymbol{x})_i \geq 0$$

$$\theta_\lambda(\boldsymbol{x})_i \geq 0$$

  - bond limitation

$$\theta_p u'(c(\boldsymbol{x})_i) = \beta E_z[u'(c(\boldsymbol{x}_+)_{i+1})] + \kappa \theta_\mu(\boldsymbol{x})_i$$

$$\theta_\mu(\boldsymbol{x})_i (\theta_a(\boldsymbol{x})_i + \kappa \theta_d(\boldsymbol{x})_i) = 0$$

$$\theta_a(\boldsymbol{x})_i + \kappa \theta_d(\boldsymbol{x})_i \geq 0$$

$$\theta_\mu(\boldsymbol{x})_i \geq 0$$

$$\sum_{i=1}^{N-1} \theta_d(\boldsymbol{x})_i = 0$$

  - and $\boldsymbol{x}_+ = [z_+, 0, \theta_a(\boldsymbol{x})^T, 0, \theta_d(\boldsymbol{x})^T]^T$

- Market clear eqs

$$\Delta(\boldsymbol{x})_i := \theta_a(\boldsymbol{x})_i - r(\boldsymbol{x}) x_{1+i}$$

$$r(\boldsymbol{x}) = f_K(\sum_{i=1}^{N} x_{1+i}, \sum_{i=1}^{N} l^i(x_1), x_1)$$

$$w(\boldsymbol{x}) = f_L(\sum_{i=1}^{N} x_{1+i}, \sum_{i=1}^{N} l^i(x_1), x_1)$$

$$c(\boldsymbol{x})_i = l^i w(\boldsymbol{x}) + x_{1+i+N} - \theta_p(\boldsymbol{x})\theta_d(\boldsymbol{x})_i - \Delta(\boldsymbol{x})_i - \frac{\zeta}{2}(\Delta(\boldsymbol{x})_i)^2$$

Training Algorithm

- The goal of our solution framework is to approximate the equilibrium functions θ
- Generically, the four components are given by:
  1. A suitable class of function approximators
  2. A loss function measuring the quality of a given approximation at a given state
  3. An updating mechanism to improve the approximation
  4. A sampling method for choosing states for the updating and the evaluation of the approximation quality.

Function approximator

- Neural networks are universal function approximators, can resolve distinct local, highly nonlinear features, and can handle a large amount of high-dimensional input data.
- Using densely connected feedforward neural networks as function approximators
- Given hyperparameters $\{K, \{m_i\}_{i=1}^K, \{\sigma_i(\cdot)\}_{i=1}^K\}$ and trainable parameters ρ, a neural network $\mathcal{N}_\rho$ encodes the map

$$\mathbf{x} \to \mathcal{N}_\rho(\mathbf{x}) = \sigma_K(\mathbf{W}_K \dots \sigma_2(\mathbf{W}_2\sigma_1(\mathbf{W}_1\mathbf{x} + \mathbf{b}_1) + \mathbf{b}_2) \dots + \mathbf{b}_K)$$

  - where $\mathbf{W}_i \in \mathbb{R}^{m_{i+1} \times m_i}$ are matrices often referred to as weight matrices.
  - where $\mathbf{b}_i \in \mathbb{R}^{m_{i+1}}$ are vectors often referred to as bias vectors.
  - the ρ denotes the collection of all entries of the weight matrices and the bias vectors.
  - K is referred to as the number of layers of the neural network.
  - m_i as the number of nodes in layer i
  - σ_i are referred to as activation functions
- A FFN is hence given by a sequence of matrix-vector multiplications followed by the application of an activation function.

Loss function

- Loss function: a measure of the quality of NN system approximation at a given state of the economy.
- The neural network maps the state $\boldsymbol{x}$ into the approximated equilibrium functions

$$\mathcal{N}_\rho : \mathcal{Z} \times \mathbb{R}^{2N} \to \mathbb{R}^{4(N-1)+1} : \mathbf{x} \to \mathcal{N}_\rho(\mathbf{x})$$

- and

$$\mathcal{N}_\rho(\mathbf{x}) = \hat{\boldsymbol{\theta}}(\mathbf{x}) = [\hat{\boldsymbol{\theta}}_a(\mathbf{x})^T, \hat{\boldsymbol{\theta}}_\lambda(\mathbf{x})^T, \hat{\boldsymbol{\theta}}_d(\mathbf{x})^T, \hat{\boldsymbol{\theta}}_\mu(\mathbf{x})^T, \hat{\theta}_p(\mathbf{x})]^T$$

- The general goal is to find parameters ρ, such that the resulting approximation of the policy functions fulfills Equations(p18).
- Given neural network parameters ρ and a state $\boldsymbol{x}_i$ for the economy, need to define a measure of the quality of the approximation by evaluating the errors in the equilibrium conditions, which result from the policy functions encoded by the NN.
- Define the error in each of the equations separately before summing them to form a single measure encoding all equilibrium conditions.

Loss function - Firm optimization

- For any state $\boldsymbol{x}$ and neural network parameters ρ, the equilibrium prices for capital and labor depend on the state alone.
- Thus, we can use above functions to satisfy them exactly:

$$r(\boldsymbol{x}) = f_K(\sum_{i=1}^{N} x_{1+i}, \sum_{i=1}^{N} l^i(x_1), x_1)$$

$$w(\boldsymbol{x}) = f_L(\sum_{i=1}^{N} x_{1+i}, \sum_{i=1}^{N} l^i(x_1), x_1)$$

Loss function - Budget constraint of the household

$$c(\boldsymbol{x})_i = I^i w(\boldsymbol{x}) + x_{1+i+N} - \theta_p(\boldsymbol{x})\theta_d(\boldsymbol{x})_i - \Delta(\boldsymbol{x})_i - \frac{\zeta}{2}(\Delta(\boldsymbol{x})_i)^2$$

- the budget constraints of the households hold exactly by substituting $\hat{\boldsymbol{a}}, \hat{\boldsymbol{d}}, \hat{p}$:

$$\hat{c}(\mathbf{x})_i = I^i w(\mathbf{x}) + x_{1+i+N} - \hat{p}(\mathbf{x})\hat{d}(\mathbf{x})_i - \hat{\Delta}(\mathbf{x})_i - \frac{\zeta}{2}\left(\hat{\Delta}(\mathbf{x})_i\right)^2$$

- where

$$\hat{\Delta}(\mathbf{x})_i := \hat{a}(\mathbf{x})_i - r(\mathbf{x})x_{1+i}$$

Loss function - State transition

- The transition from one state to the next,can also be satisfied exactly by setting the next period′ s endogenous state consistent with the current period′ s policies:

$$\mathbf{x}_+ = [z_+, 0, \hat{\mathbf{a}}(\mathbf{x})^T, 0, \hat{\mathbf{d}}(\mathbf{x})^T]^T$$

Loss function - Household′ s optimization

- For Euler Equation (capital):

$$(1+\zeta\Delta_t^s(\boldsymbol{x})_i)u'(c(\boldsymbol{x})_i) = \beta E_z[u'(c(\boldsymbol{x}_+)_{i+1})r(\boldsymbol{x}_+)_{i+1}(1+\zeta\Delta(\boldsymbol{x}_+)_{i+1})] + \theta_\lambda(\boldsymbol{x})_i + \theta_\mu(\boldsymbol{x})_i$$

- Substituted by $\{\bar{\cdot}\}$:

$$(1+\zeta\hat{\Delta}_t^s(\boldsymbol{x})_i)u'(\hat{c}(\boldsymbol{x})_i) = \beta E_z[u'(\hat{c}(\boldsymbol{x}_+)_{i+1})r(\boldsymbol{x}_+)_{i+1}(1+\zeta\hat{\Delta}(\boldsymbol{x}_+)_{i+1})] + \hat{\lambda}(\boldsymbol{x})_i + \hat{\mu}(\boldsymbol{x})_i$$

$$u'(\hat{c}(\boldsymbol{x}_j)_i) = \frac{\beta E_{z_j}[u'(\hat{c}(\boldsymbol{x}_{j,+})_{i+1})r(\boldsymbol{x}_{j,+})_{i+1}(1+\zeta\Delta(\boldsymbol{x}_{j,+})_{i+1})] + \hat{\lambda}(\boldsymbol{x}_j)_i + \hat{\mu}(\boldsymbol{x}_j)_i}{1+\zeta\hat{\Delta}(\boldsymbol{x}_j)_i}$$

- Get the form with $c =$:

$$\hat{c}(\boldsymbol{x}_j)_i = u'^{-1}\Big[\frac{\beta E_{z_j}[u'(\hat{c}(\boldsymbol{x}_{j,+})_{i+1})r(\boldsymbol{x}_{j,+})_{i+1}(1+\zeta\Delta(\boldsymbol{x}_{j,+})_{i+1})] + \hat{\lambda}(\boldsymbol{x}_j)_i + \hat{\mu}(\boldsymbol{x}_j)_i}{1+\zeta\hat{\Delta}(\boldsymbol{x}_j)_i}\Big]$$

- Euler equation error of generation i in state $\boldsymbol{x}_j$:

$$e_{x_j}^{i,REE\ cap}(\rho) := \frac{u'^{-1}\Big[\frac{\beta E_{z_j}[u'(\hat{c}(\boldsymbol{x}_{j,+})_{i+1})r(\boldsymbol{x}_{j,+})_{i+1}(1+\zeta\Delta(\boldsymbol{x}_{j,+})_{i+1})]+\hat{\lambda}(\boldsymbol{x}_j)_i+\hat{\mu}(\boldsymbol{x}_j)_i}{1+\zeta\hat{\Delta}(\boldsymbol{x}_j)_i}\Big]}{\hat{c}(\boldsymbol{x}_j)_i} - 1$$

- Same for Euler Equation (bond):

$$\theta_p u'(c(\boldsymbol{x})_i) = \beta E_z[u'(c(\boldsymbol{x}_+)_{i+1})] + \kappa\theta_\mu(\boldsymbol{x})_i$$

- Euler equation error of generation i in state $\boldsymbol{x}_j$:

$$e^{i,\text{REE bond}}_{x_j}(\rho) := \frac{u'^{-1}\left[\frac{\beta E_{z_j}[u'(c(\boldsymbol{x}_{j,+})_{i+1})]+\kappa\hat{\mu}(\boldsymbol{x}_j)_i}{\hat{p}}\right]}{c(\boldsymbol{x}_j)_i} - 1$$

- KKT conditions

$$e^{i,KKT\ cap}_{x_j}(\rho) := \hat{\lambda}(\boldsymbol{x}_j)_i \hat{a}(\boldsymbol{x}_j)_i$$

$$e^{i,KKT\ bond}_{x_j}(\rho) := \hat{\mu}(\boldsymbol{x}_j)_i (\hat{a}(\boldsymbol{x}_j)_i + \kappa \hat{d}(\boldsymbol{x}_j)_i)$$

- Market clearing:

$$e^{mc}_{x_j}(\rho) = \sum_{i=1}^{N-1} \hat{d}(\boldsymbol{x}_j)_i$$

Total Loss function

- If the NN would encode the true equilibrium functions exactly, above loss functions well be zero.
- Define a loss function, that is, a measure of the total quality of the approximation (by NN).

$$Loss_{\mathcal{D}_{train}} := \frac{1}{|\mathcal{D}_{train}|} \sum_{x_j \in \mathcal{D}_{train}} \{ \frac{1}{N-1} \sum_{i=1}^{N-1} [(e_{x_j}^{i,REE\ cap}(\rho))^2$$

$$+(e_{x_j}^{i,REE\ bond}(\rho))^2 + (e_{x_j}^{i,KKT\ cap}(\rho))^2 + (e_{x_j}^{i,KKT\ bond}(\rho))^2] + (e_{x_j}^{mc}(\rho))^2 \}$$

- where $\mathcal{D}_{train}$ is a set of state.

Updating mechanism

- The main goal is to find the best approx
- Thus, minimize the Total Loss function as:

$$\rho_k^{new} = \rho_k^{old} - \alpha^{learn} \frac{\partial Loss_{\mathcal{D}_{train}}}{\partial \rho_k^{old}}$$

- with α^{learn} as learning rate.

Sampling

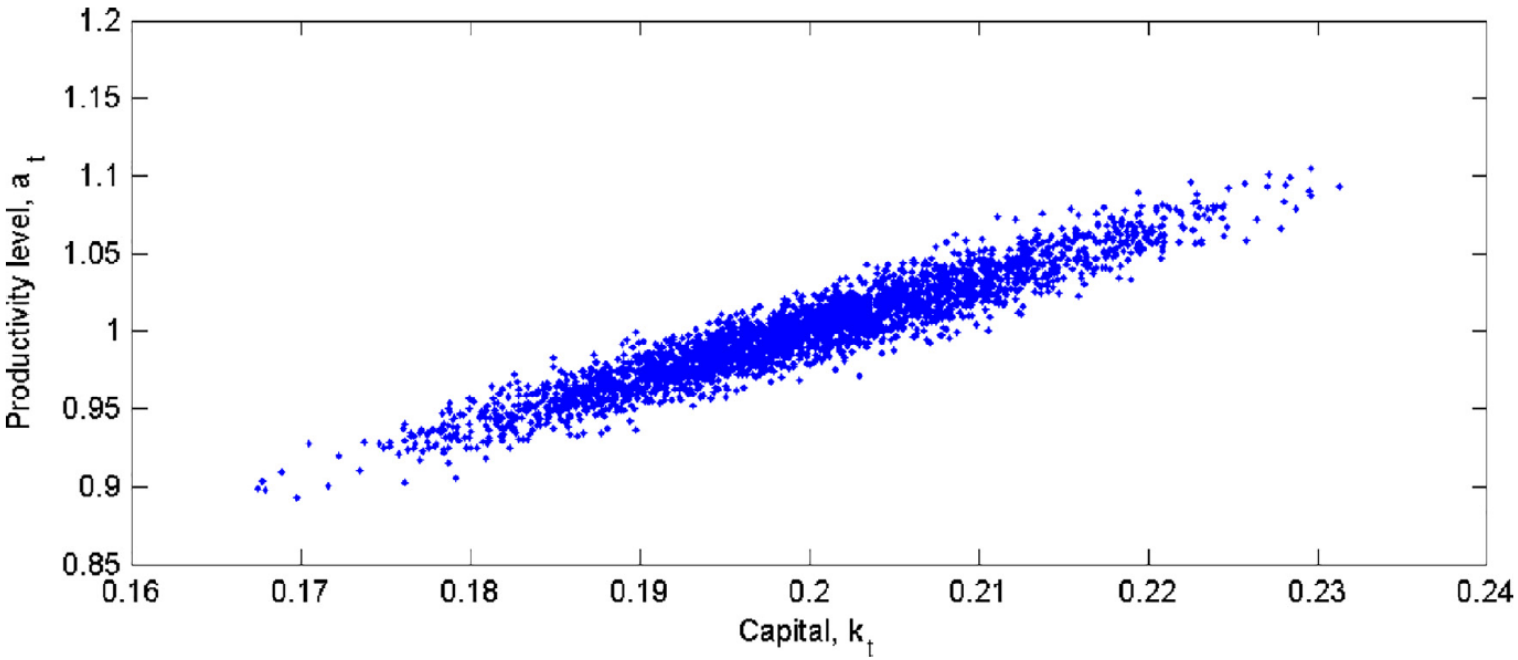


图: Useful Datas

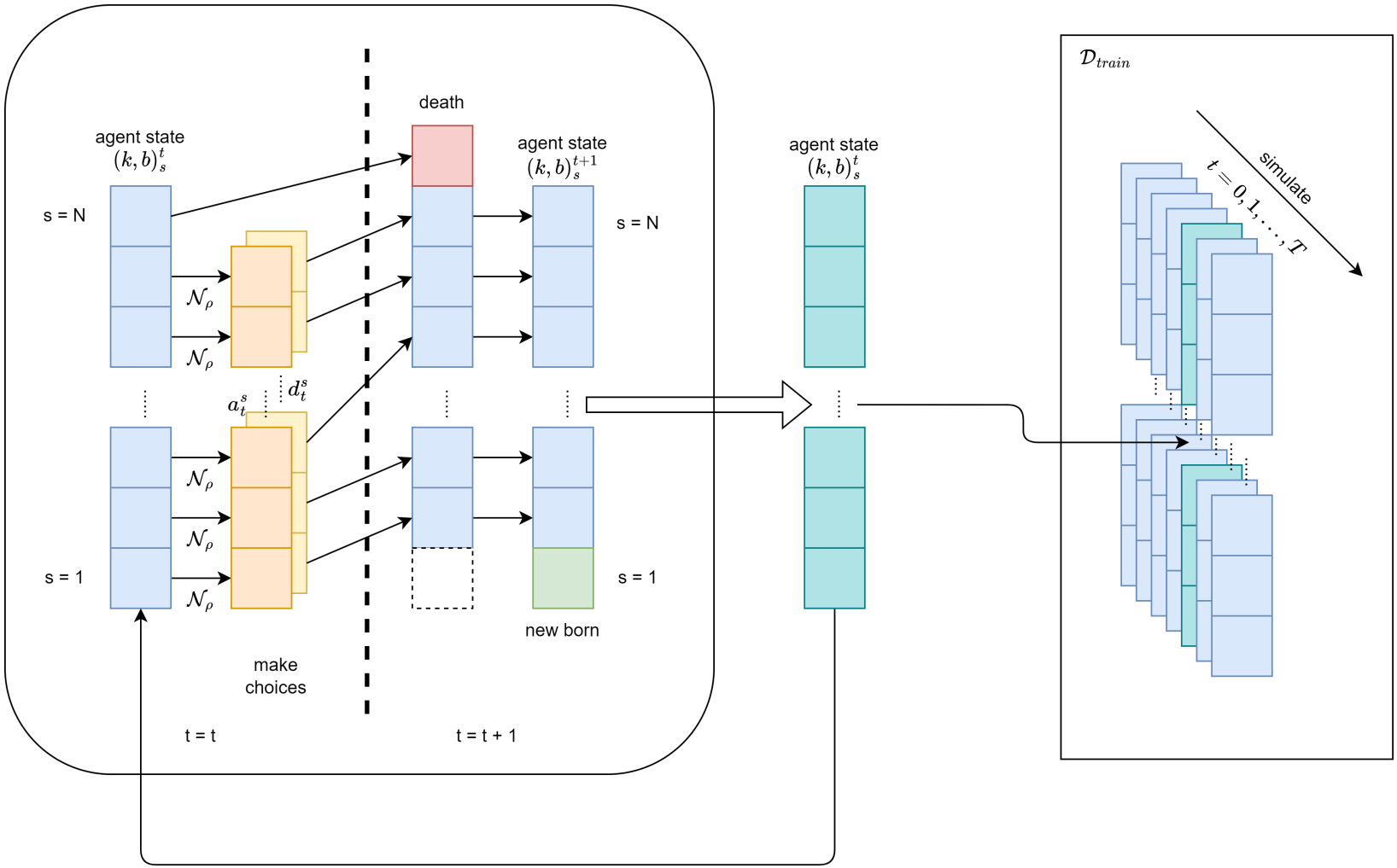


图: Data Simulation

Algorithm 1. Algorithm for training deep equilibrium nets

Data:
T (length of an episode), N^{epochs} (number of epochs on each episode), N^{iter} (maximum number of iterations),
τ^{mean}, $\tau^{\max}$ (desired threshold for mean and max error, respectively),
$\epsilon^{\text{mean}} = \infty$, $\epsilon^{\max} = \infty$ (starting value for current mean and max error, respectively),
$\boldsymbol{\rho}^0$ (initial parameters of the neural network), $\mathbf{x}_1^0$ (initial state to start simulations from), $i = 0$ (set counter),
α^{learn} (learning rate)
Result:
success (boolean if thresholds were reached)
$\boldsymbol{\rho}^{\text{final}}$ (final neural network parameters)
while $((i < N^{iter}) \wedge ((\epsilon^{mean} \geq \tau^{mean}) \vee (\epsilon^{max} \geq \tau^{max})))$ **do**
  $\mathcal{D}^i_{\text{train}} \leftarrow \{\mathbf{x}_1^i, \mathbf{x}_2^i, \ldots, \mathbf{x}_T^i\}$ (generate new training data)
  $\mathbf{x}_0^{i+1} \leftarrow \mathbf{x}_T^i$ (set new starting point)
  $\epsilon_{\max} \leftarrow \max\left\{\max_{\mathbf{x}\in\mathcal{D}^i_{\text{train}}} |e^{\cdot\cdot\cdot}_{\mathbf{x}}(\boldsymbol{\rho})|\right\}$, $\epsilon_{\text{mean}} \leftarrow \max\left\{\frac{1}{T}\sum_{\mathbf{x}\in\mathcal{D}^i_{\text{train}}} |e^{\cdot\cdot\cdot}_{\mathbf{x}}(\boldsymbol{\rho})|\right\}$ (update errors)
  for $j \in [1, ..., N^{epochs}]$ **do**
    (learn N^{epochs} on data)
    for $k \in [1, ..., length(\boldsymbol{\rho})]$ **do**

$$\rho_k^{i+1} = \rho_k^i - \alpha^{\text{learn}} \frac{\partial \ell_{\mathcal{D}^i_{\text{train}}}(\boldsymbol{\rho}^i)}{\partial \rho_k^i} \tag{50}$$

      (do a gradient descent step to update the network parameters)
    end
  end
  $i \leftarrow i + 1$ (update episode counter)
end
if $i = N^{iter}$ **then return** (success $\leftarrow False, \boldsymbol{\rho}^{\text{final}} \leftarrow \boldsymbol{\rho}^i$) ;
else return (success $\leftarrow True, \boldsymbol{\rho}^{\text{final}} \leftarrow \boldsymbol{\rho}^i$) ;

Applied on a simple Model

Household

- Household live for $A = 6$ periods, period t, household age s
- Household supply labor $l_t^0 = 1$ only in new born.

$$\max_{\{c_{t+i}^i, a_{t+i}^i\}_{i=0}^{A-1}} E_t \sum_{i=0}^{A-1} \log(c_{t+i}^i)$$

$$c_t^h + a_t^s = r_t k_t^s + w_t l_t^h$$

$$k_{t+1}^{s+1} = a_t^s$$

$$a_t^{A-1} \geq 0$$

- Household born with $k_0 = 0$

Firm

- With C-D form product function

$$f(K_t, L_t, z_t) = \eta_t K_t^{\alpha} L_t^{1-\alpha} + K_t(1-\delta_t)$$

- with

$$K_t = \sum_{s=0}^{A-1} k_t^s$$

$$L_t = \sum_{s=0}^{A-1} l_t^s = 1$$

- and

$$r_t = \alpha \eta_t K_t^{\alpha-1} L_t^{1-\alpha} + (1-\delta_t)$$

$$w_t = (1-\alpha)\eta_t K_t^{\alpha} L_t^{-\alpha}$$

- and shock with $\{1,2,3,4\}$i.i.d, accordingly $\delta_t \in \{0.5, 0.5, 0.9, 0.9\}$, $\eta_t \in \{0.95, 1.05, 0.95, 1.05\}$.

Equilibrium

- Basic equilibrium.

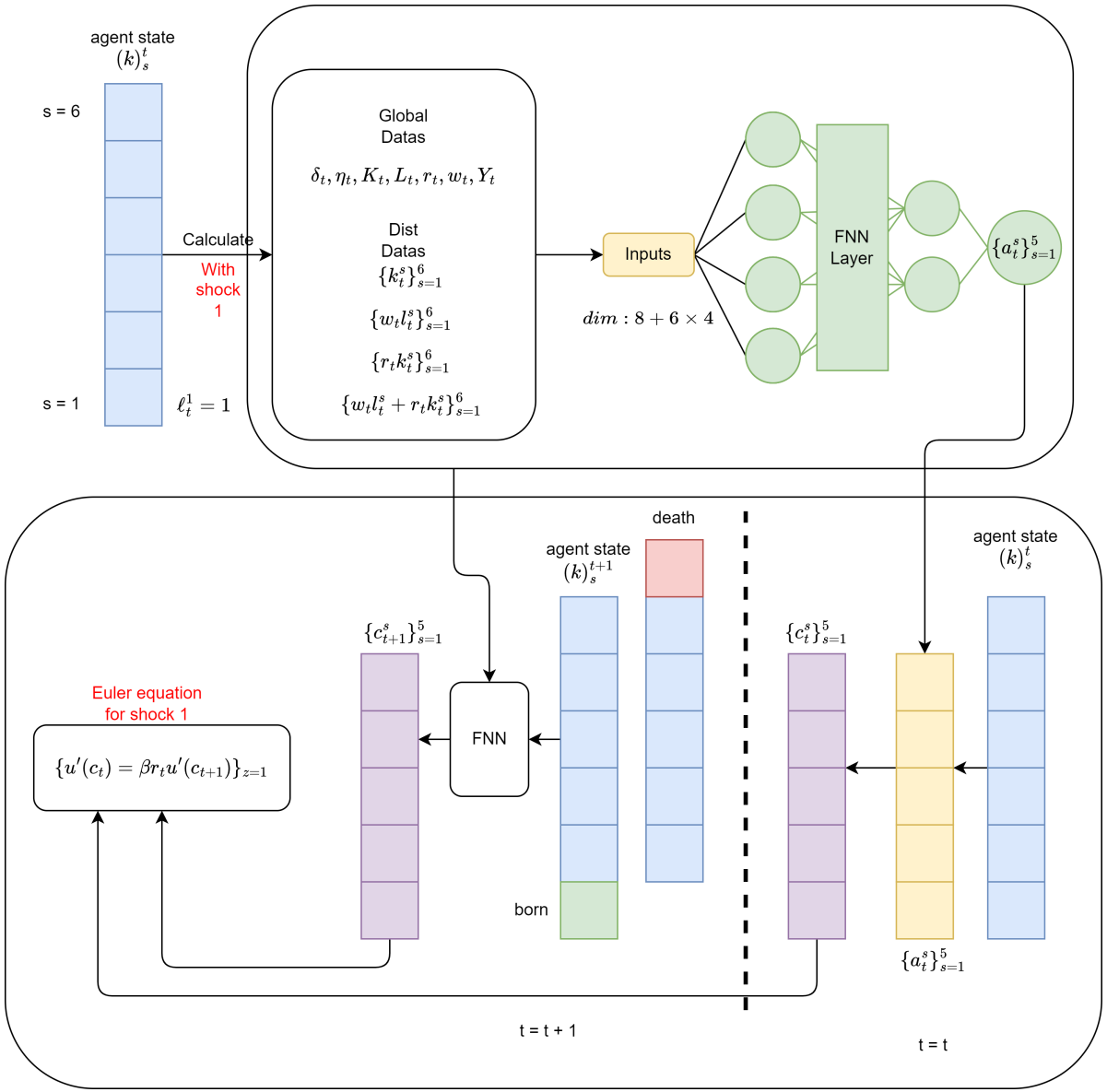


图: Solving Method - partial

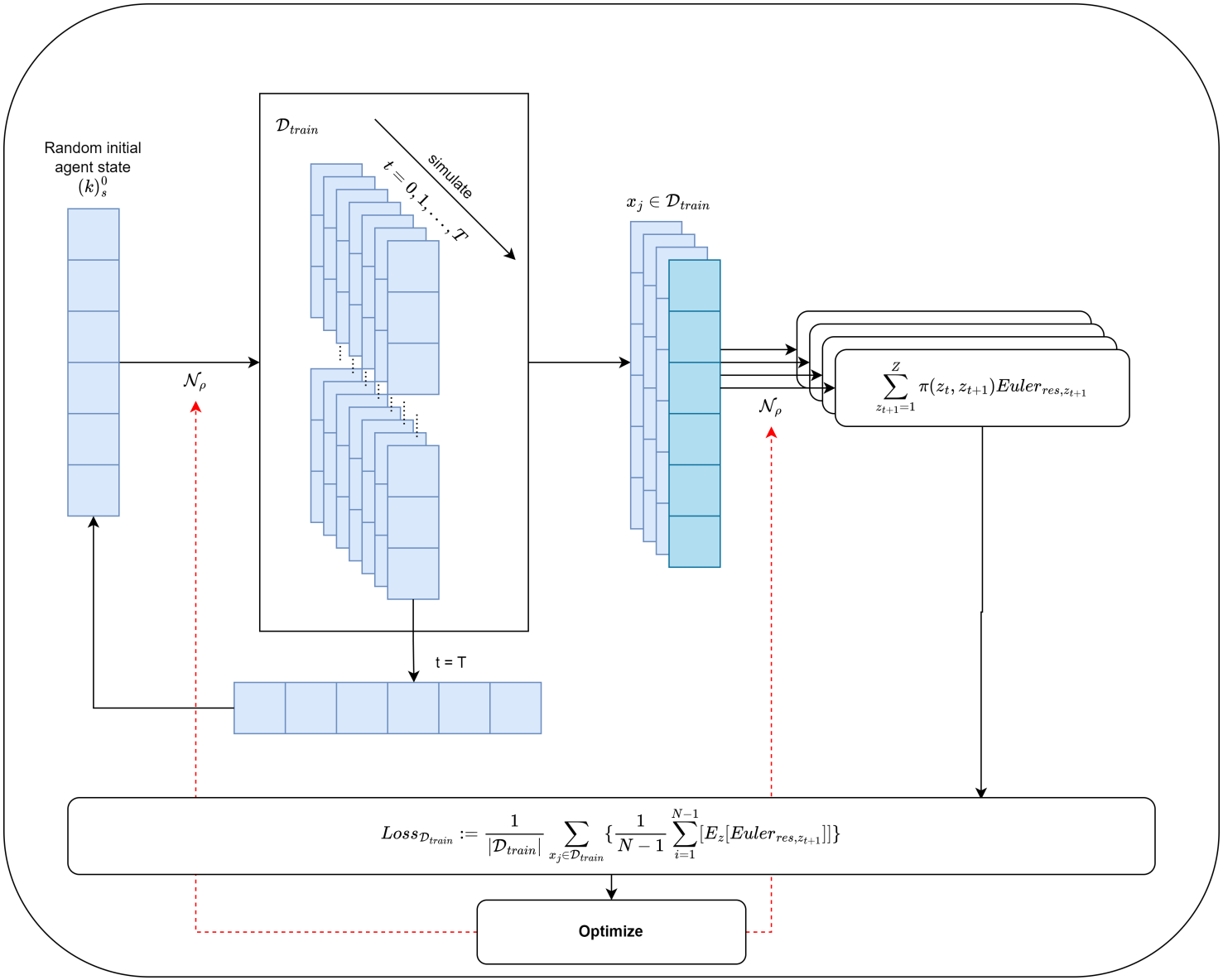
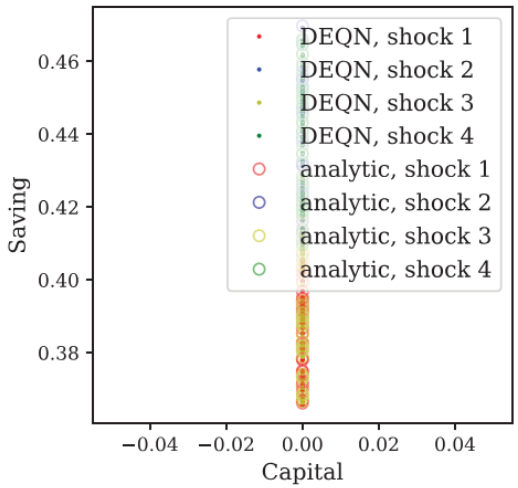
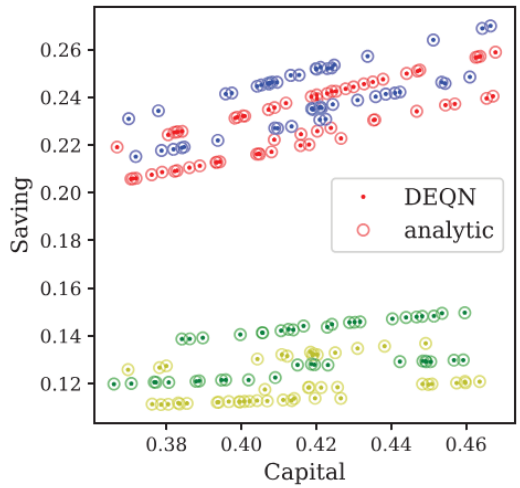
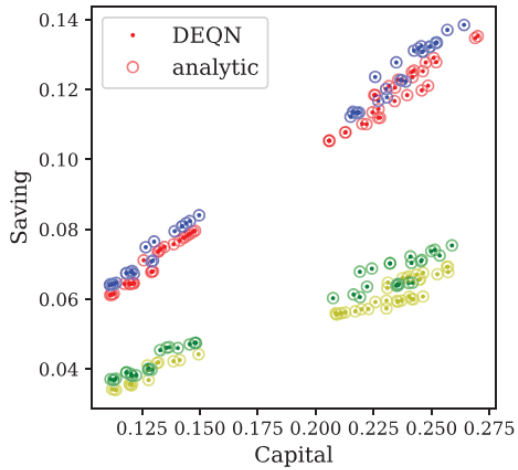
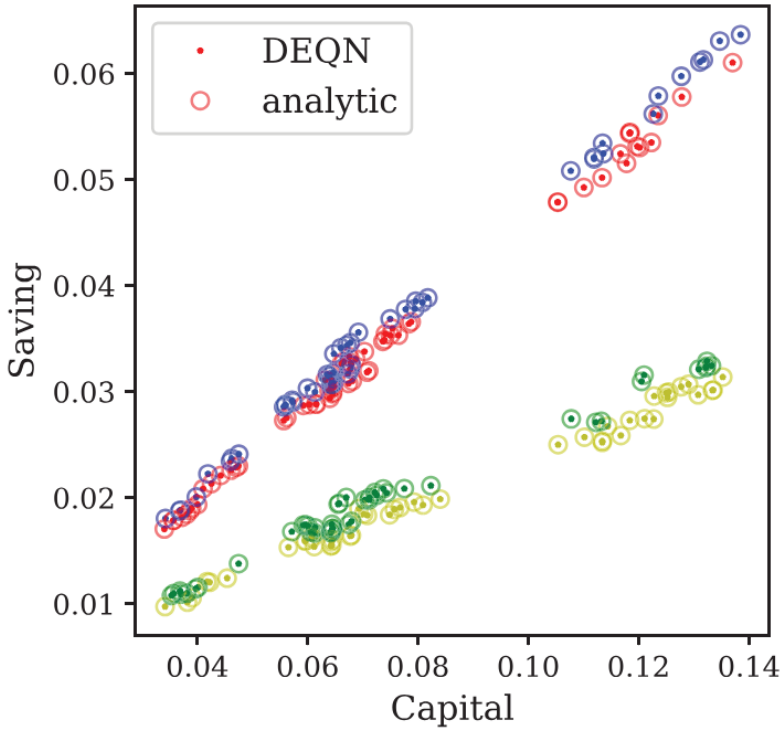


图: Solving Method - general









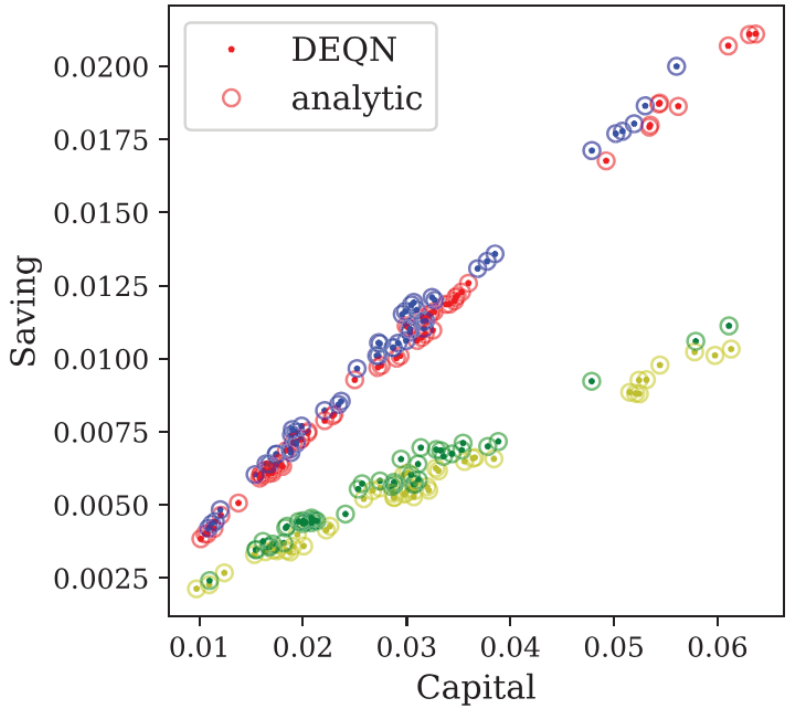


图: Result - DEQN vs. analytic

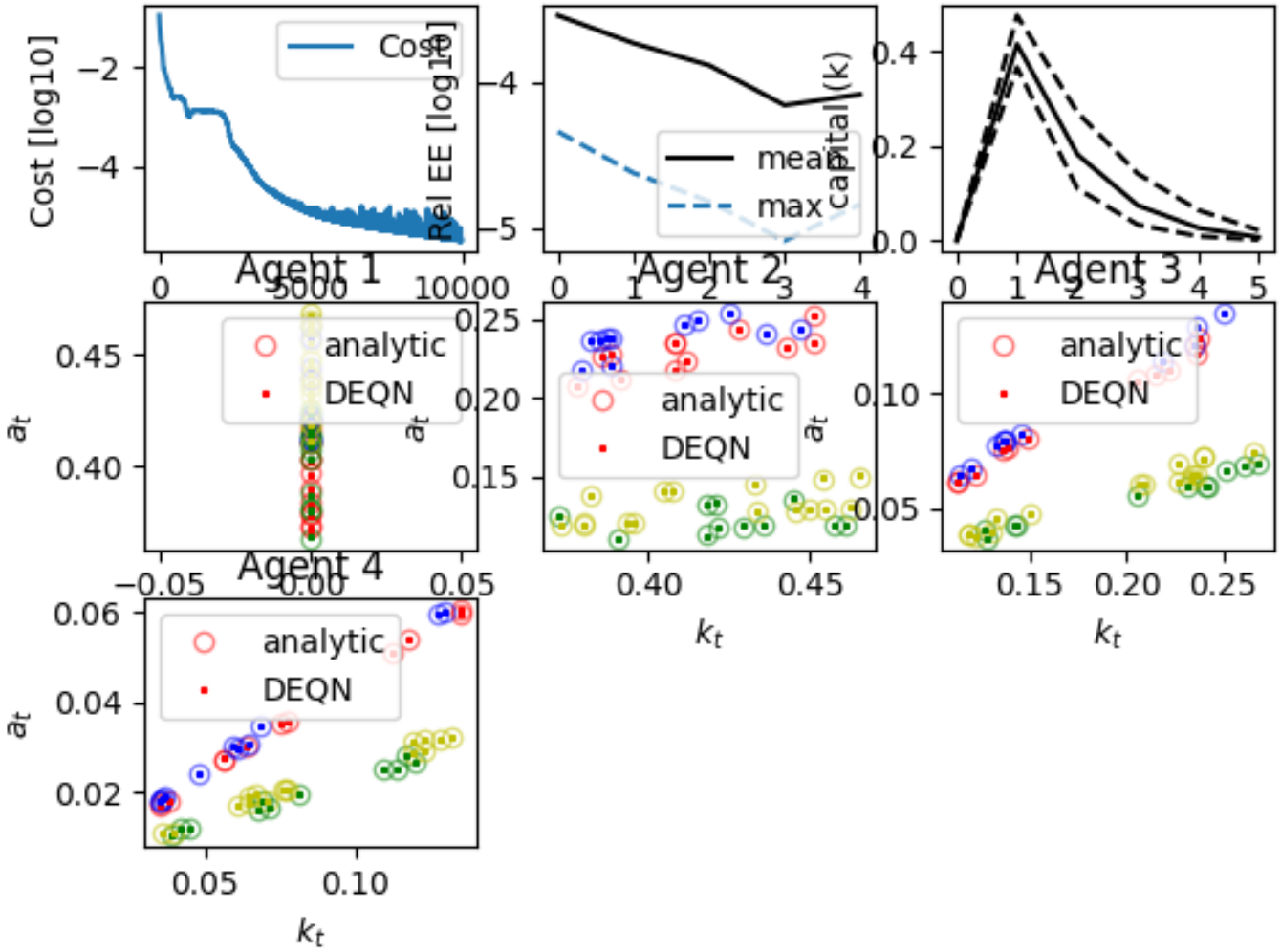


图: Result - my replication

Applied on Benchmark Model

Household

- Household live for $A = 56$ periods, period t, household age s, $\underline{a} = 0$
- Household supply labor roughly mimics the Guvenen et al. (2021).

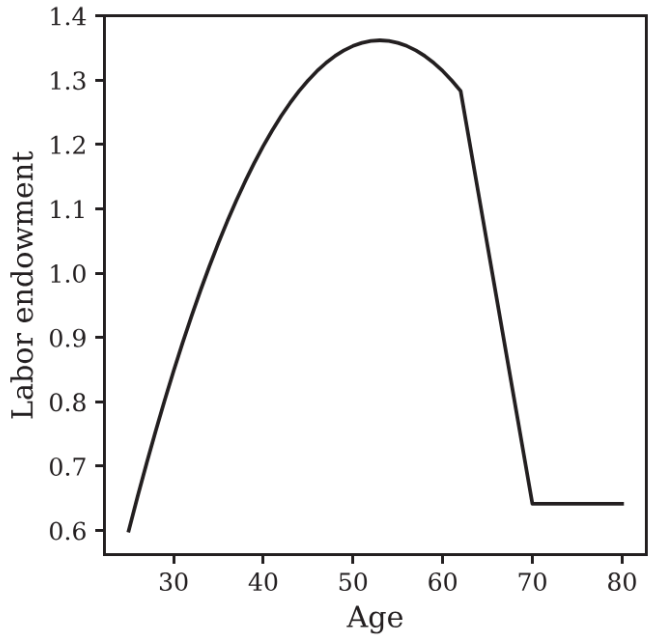


图: Labor endownment

Frim

- Divided in to 4 shocks.

$$shock = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

$$\delta = \begin{bmatrix} 0.08 & 0.08 \\ 0.11 & 0.11 \end{bmatrix}, \quad \delta = \begin{bmatrix} 0.978 & 1.022 \\ 0.978 & 1.022 \end{bmatrix}$$

- prob matrix:

$$\pi^{\delta} = \begin{bmatrix} 0.972 & 0.028 \\ 0.300 & 0.700 \end{bmatrix}, \quad \pi^{\delta} = \begin{bmatrix} 0.905 & 0.095 \\ 0.095 & 0.905 \end{bmatrix}$$

- Total trans prob matrix

$$\Pi = \begin{bmatrix} 0.972 * 0.905 & 0.972 * 0.095 & 0.028 * 0.905 & 0.028 * 0.095 \\ 0.972 * 0.095 & 0.972 * 0.905 & 0.028 * 0.095 & 0.028 * 0.905 \\ 0.300 * 0.905 & 0.300 * 0.095 & 0.700 * 0.905 & 0.700 * 0.095 \\ 0.300 * 0.095 & 0.300 * 0.905 & 0.700 * 0.095 & 0.700 * 0.905 \end{bmatrix}$$

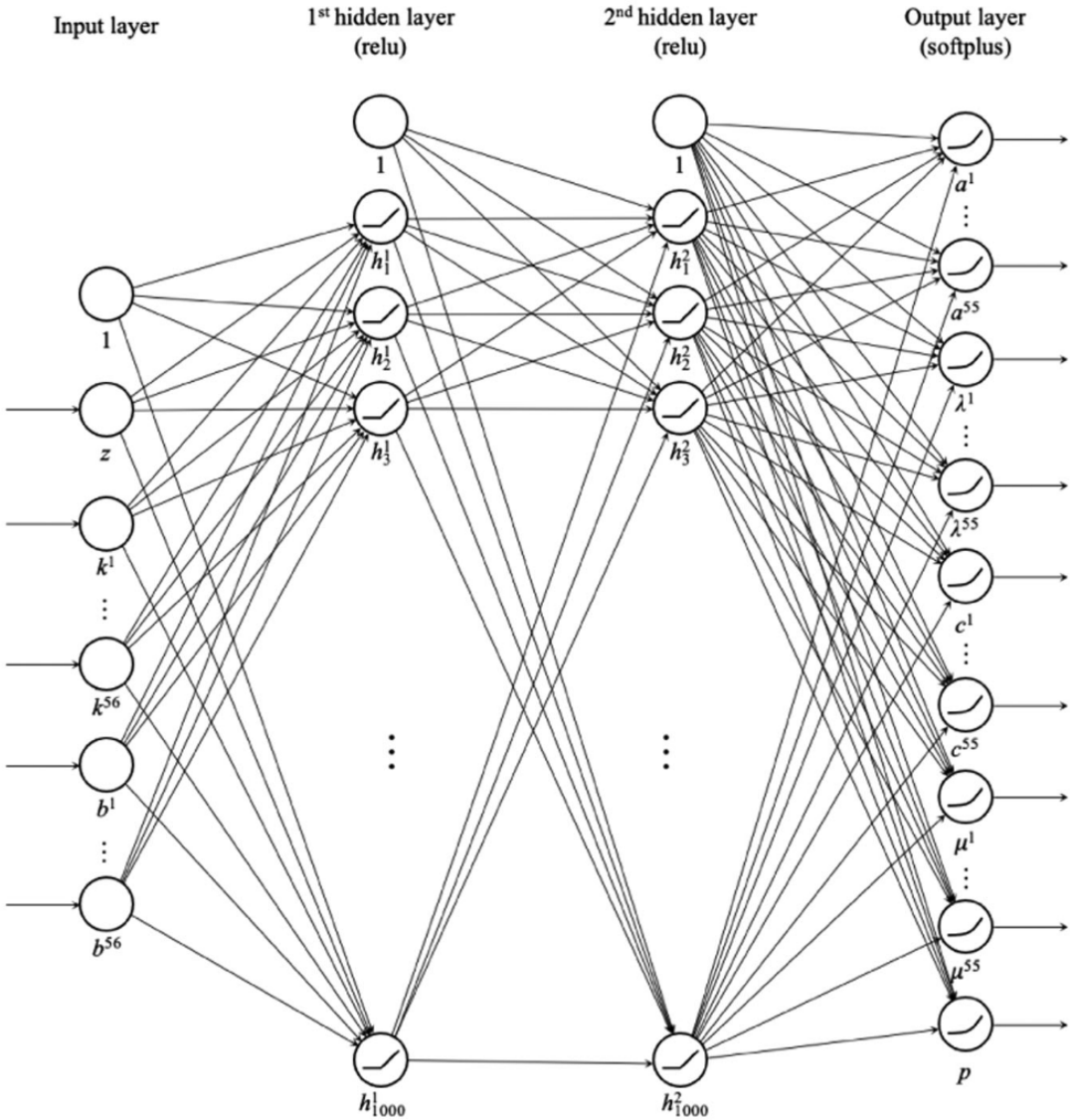
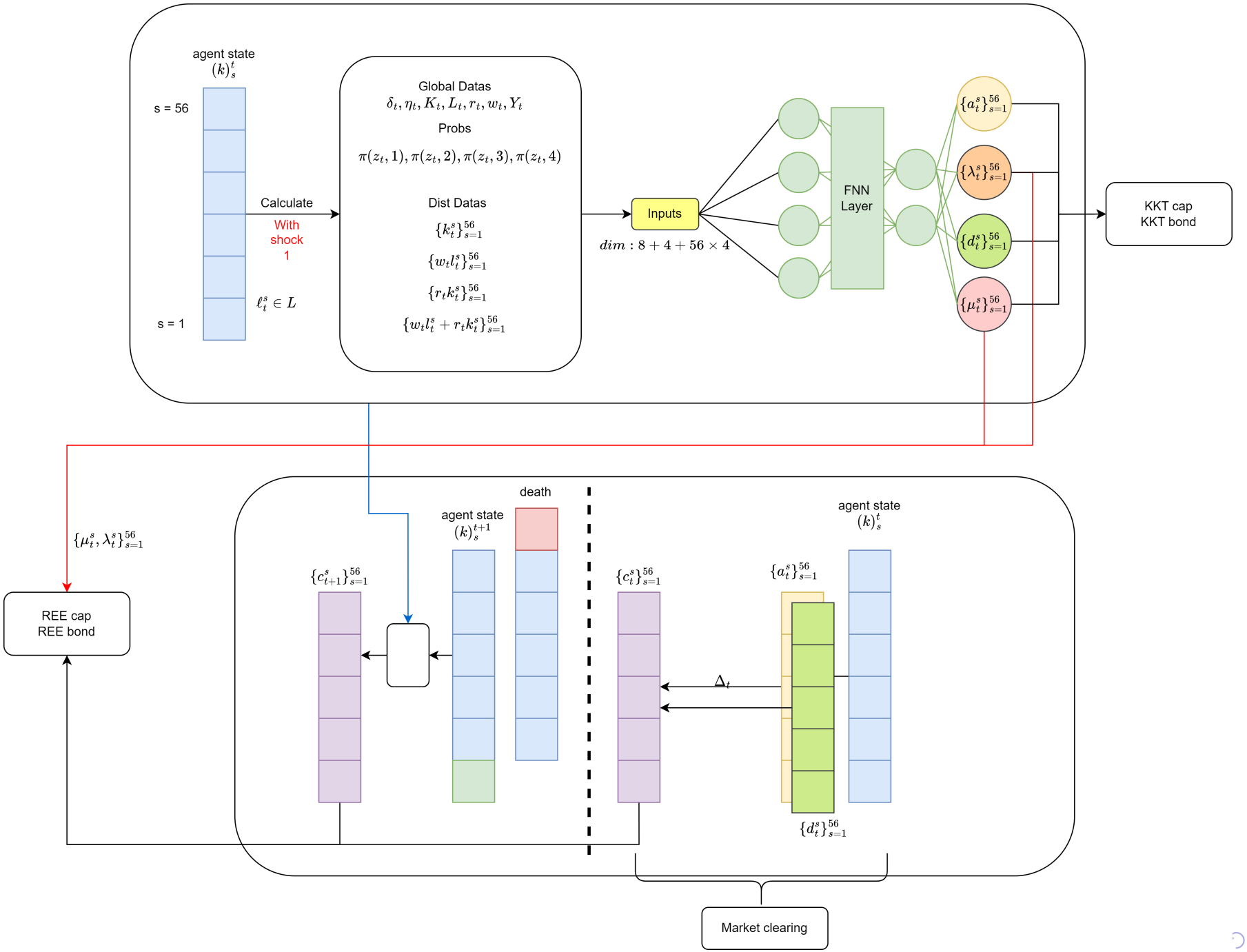


图: Nerual Network


agent state
$(k)_s^t$
s = 56
s = 1
$\ell_t^s \in L$
Calculate
With
shock
1
Global Datas
$\delta_t, \eta_t, K_t, L_t, r_t, w_t, Y_t$
Probs
$\pi(z_t,1), \pi(z_t,2), \pi(z_t,3), \pi(z_t,4)$
Dist Datas
$\{k_t^s\}_{s=1}^{56}$
$\{w_t l_t^s\}_{s=1}^{56}$
$\{r_t k_t^s\}_{s=1}^{56}$
$\{w_t l_t^s + r_t k_t^s\}_{s=1}^{56}$
Inputs
$dim: 8 + 4 + 56 \times 4$
FNN
Layer
$\{a_t^s\}_{s=1}^{56}$
$\{\lambda_t^s\}_{s=1}^{56}$
$\{d_t^s\}_{s=1}^{56}$
$\{\mu_t^s\}_{s=1}^{56}$
KKT cap
KKT bond
$\{\mu_t^s, \lambda_t^s\}_{s=1}^{56}$
REE cap
REE bond
death
agent state
$(k)_s^{t+1}$
$\{c_{t+1}^s\}_{s=1}^{56}$
$\{c_t^s\}_{s=1}^{56}$
Δ_t
agent state
$(k)_s^t$
$\{a_t^s\}_{s=1}^{56}$
$\{d_t^s\}_{s=1}^{56}$
Market clearing

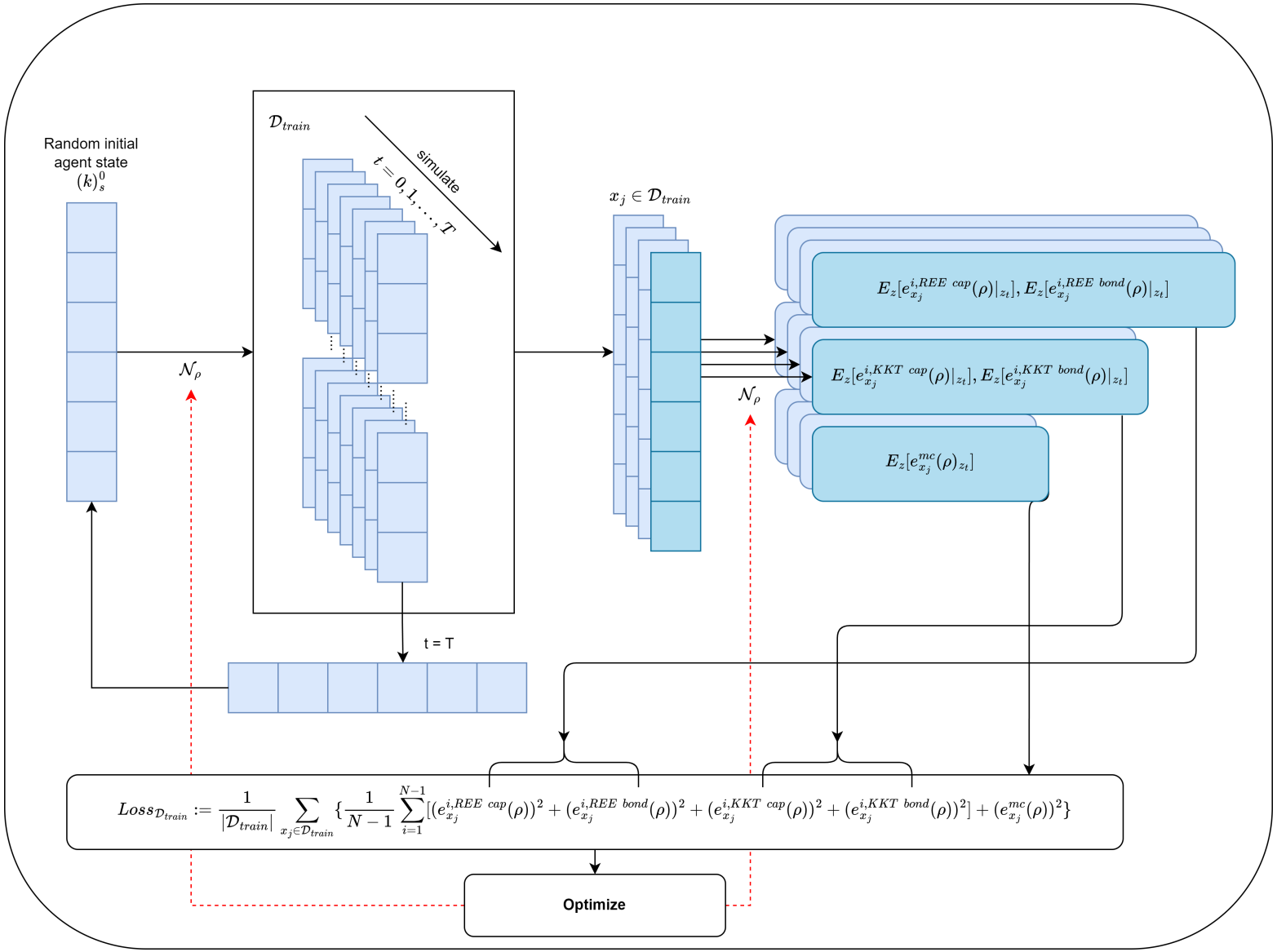

Random initial agent state $(k)_s^0$
$\mathcal{D}_{train}$
simulate
$t = 0, 1, \ldots, T$
$\mathcal{N}_\rho$
t = T
$x_j \in \mathcal{D}_{train}$
$\mathcal{N}_\rho$
$E_z[e_{x_j}^{i,REE\ cap}(\rho)|_{z_t}], E_z[e_{x_j}^{i,REE\ bond}(\rho)|_{z_t}]$
$E_z[e_{x_j}^{i,KKT\ cap}(\rho)|_{z_t}], E_z[e_{x_j}^{i,KKT\ bond}(\rho)|_{z_t}]$
$E_z[e_{x_j}^{mc}(\rho)_{z_t}]$
$Loss_{\mathcal{D}_{train}} := \frac{1}{|\mathcal{D}_{train}|} \sum_{x_j \in \mathcal{D}_{train}} \{\frac{1}{N-1} \sum_{i=1}^{N-1} [(e_{x_j}^{i,REE\ cap}(\rho))^2 + (e_{x_j}^{i,REE\ bond}(\rho))^2 + (e_{x_j}^{i,KKT\ cap}(\rho))^2 + (e_{x_j}^{i,KKT\ bond}(\rho))^2] + (e_{x_j}^{mc}(\rho))^2\}$
Optimize

TABLE 2
HYPERPARAMETERS CHOSEN TO TRAIN THE DEEP EQUILIBRIUM NET FOR THE BENCHMARK MODEL

Episodes	Learning Rate α^{learn}	Periods per Episode T	Epochs per Episode N^{epochs}	Mini-Batch Size m	Nodes Hidden Layers	Activations Hidden Layers
1–60,000	1×10^{-5}	10,000	1	64	1,000 1,000 1,000	relu relu relu
60,000–200,000	1×10^{-6}	10,000	1	1,000	1,000	relu

图: Training process

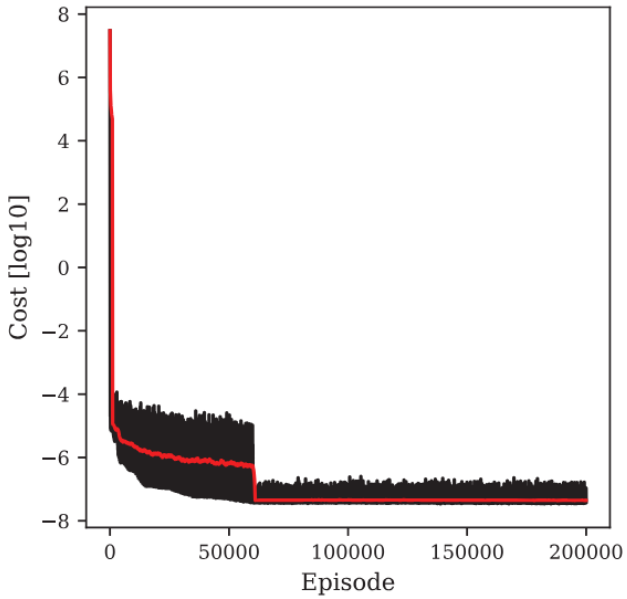
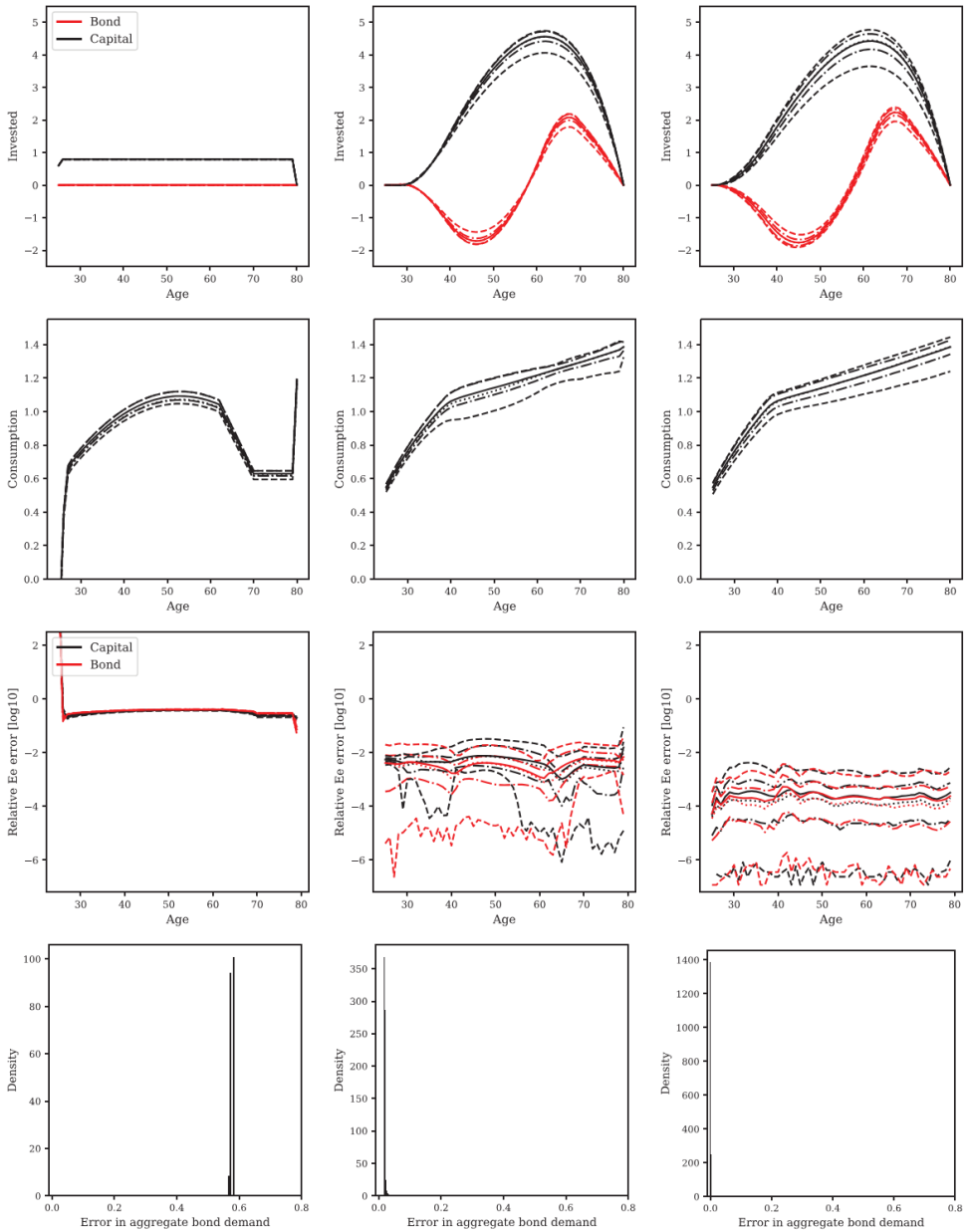


图: Loss function

TABLE 3
STATISTICS OF ERRORS IN THE EQUILIBRIUM CONDITIONS

	Mean	Max	0.1	10	50	90	99.9
Rel Ee capital [%]	0.024	0.528	0.000	0.002	0.013	0.060	0.324
Rel Ee bond [%]	0.021	0.608	0.000	0.002	0.012	0.051	0.255
KKT capital [$\times 10^{-2}$]	0.000	0.127	0.000	0.000	0.000	0.000	0.000
KKT bond [$\times 10^{-2}$]	0.005	0.104	0.000	0.000	0.000	0.000	0.021
Market clearing [$\times 10^{-2}$]	0.053	0.333	0.000	0.017	0.047	0.089	0.252
Market clearing (normalized) [$\times 10^{-2}$]	0.000	0.002	0.000	0.000	0.000	0.000	0.001

图: Loss function - each


Bond
Capital
Invested
Consumption
Relative Ee error [log10]
Density
Age
Error in aggregate bond demand

Compare with MMP(2021, JME)

Recap - Minimize Euler-residual in MMP

- Euler equation
  - Euler equations are a set of equations written in the form:

$$E_\epsilon[f_j(m, s, x, m', s', x')] = 0, j = 1, ..., J$$

  - With transition functions $s' = S(m, s, x, m'); m' = M(m, \epsilon); x \in X(m, s)$
  - If there is a decision rule $\psi(\cdot; \theta)$ which makes Euler residual $= 0$, then the model is considered solved
- Euler-residual
  - Select a decision rule $\psi(\cdot; \theta)$ and define a distribution of state variable (m, s), calaulate the expected squared residuals in the Euler equations:

$$\Xi(\theta) = E_{(m,s)}[\sum_{j=1}^{J} v_j(E_\epsilon[f_j(m, s, \psi(m, s; \theta), m', s', \psi(m', s'; \theta))])^2]$$

  - where v_j is a vector of weights on J optimality conditions
  - The goal is to construct a decision rule $\psi(\cdot; \theta)$ that solves $\min_{\theta \in \Theta} \Xi(\theta)$.

Recap - Minimize Euler-residual with AIO

- Minimize Euler-residual:
  randonly draw $(m, s, \epsilon_1, \epsilon_2)$ and use $E_{\epsilon_1}[f(\epsilon_1)]E_{\epsilon_2}[f(\epsilon_2)] = E_{(\epsilon_1,\epsilon_2)}[f(\epsilon_1)f(\epsilon_2)]$

$$\Xi(\theta) = E_\omega[\xi(\omega;\theta)] = E_{(m,s,\epsilon_1,\epsilon_2)}\{\sum_{j=1}^{J} v_j[f_j(m,s,x,m',s',x')|_{\epsilon=\epsilon_1}][f_j(m,s,x,m',s',x')|_{\epsilon=\epsilon_2}]\}$$

Rewrite - AR(1) to Markov matrix

- suppose with

$$\Omega = \begin{bmatrix} \epsilon_1 & \epsilon_2 \\ \epsilon_2 & \epsilon_1 \end{bmatrix}$$

$$\Pi = \begin{bmatrix} \pi_1 & \pi_2 \\ \pi_3 & \pi_4 \end{bmatrix}$$

- no AIO but totaly Exp: $E_\omega[\xi(\omega;\theta)|_{\epsilon_1}] = E[\Pi(1,1)f(\epsilon_1)^2 + \Pi(1,2)f(\epsilon_2)^2]$

$$\begin{aligned}\Xi(\theta) &= E_\omega[\xi(\omega;\theta)|_{\epsilon_1}] \\ &= E_{(m,s}\{\sum_{j=1}^{J} v_j[\Pi(1,1)f_j(m,s,x,m',s',x'){\epsilon_1}^2|_{\epsilon_1}][\Pi(1,2)f_j(m,s,x,m',s',x')^2_{\epsilon_1}|_{\epsilon_1}]\end{aligned}$$

$$\begin{aligned}\Xi(\theta) &= E_\omega[\xi(\omega;\theta)|_{\epsilon_2}] \\ &= E_{(m,s}\{\sum_{j=1}^{J} v_j[\Pi(2,1)f_j(m,s,x,m',s',x')^2_{\epsilon_2}|_{\epsilon_1}][\Pi(2,2)f_j(m,s,x,m',s',x')^2_{\epsilon_2}|_{\epsilon_2}]\}\end{aligned}$$

MMP nn method Vs. DEQN nn method

- Input
  - MMP: ℓ is agent num in economy, $2\ell + 3$ which $s_t^i = \{\{y_t^j, w_t^j\}_{i=1}^{\ell}, z_t, y_t^j, w_t^j\}$.
  - DEQN: N is max age, $4 * N + 8 + 4$ which $s_t^i = \{\{k_t^s, w_t l_t^s, r_t k_t^s, w_t l_t^s + r_t k_t^s\}_{s=1}^{N}, \delta_t, \eta_t, K_t, L_t, r_t, w_t, Y_t, \{\pi_{z_t,z_1}, \pi_{z_t,z_2}, \pi_{z_t,z_3}, \pi_{z_t,z_4}\}\}$.
- Output
  - MMP: policies for agent i
  - DEQN: policies for agents at age $s = 1, \ldots, N-1$
- Sampling
  - Almost Same.

MMP Vs DEQN

- the code which replicate the MMP and DEQN best
- Fixing the other part (sampling, Exp method or so), only change the NNs and Loss function.

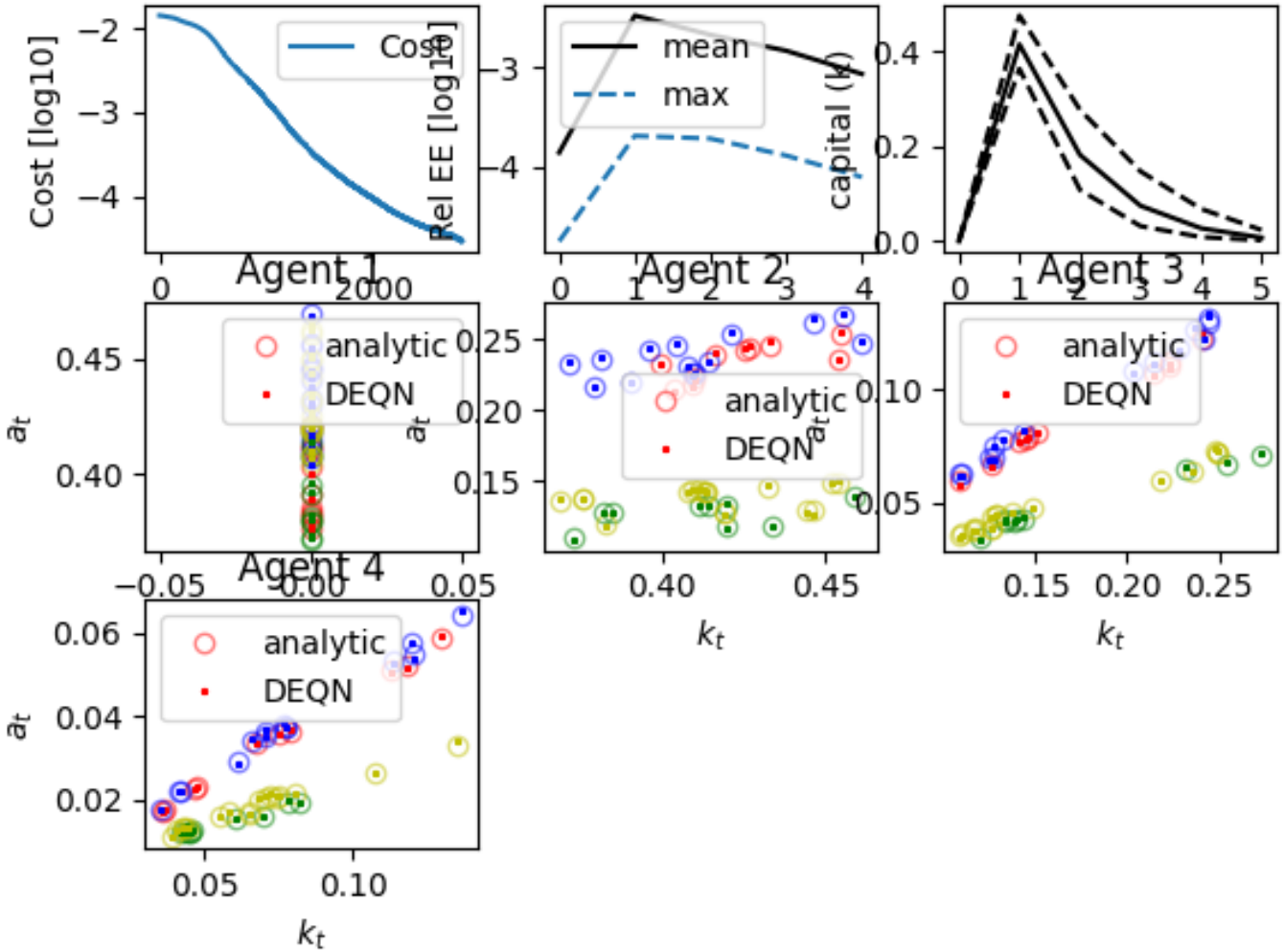


图: DEQN using MMP method

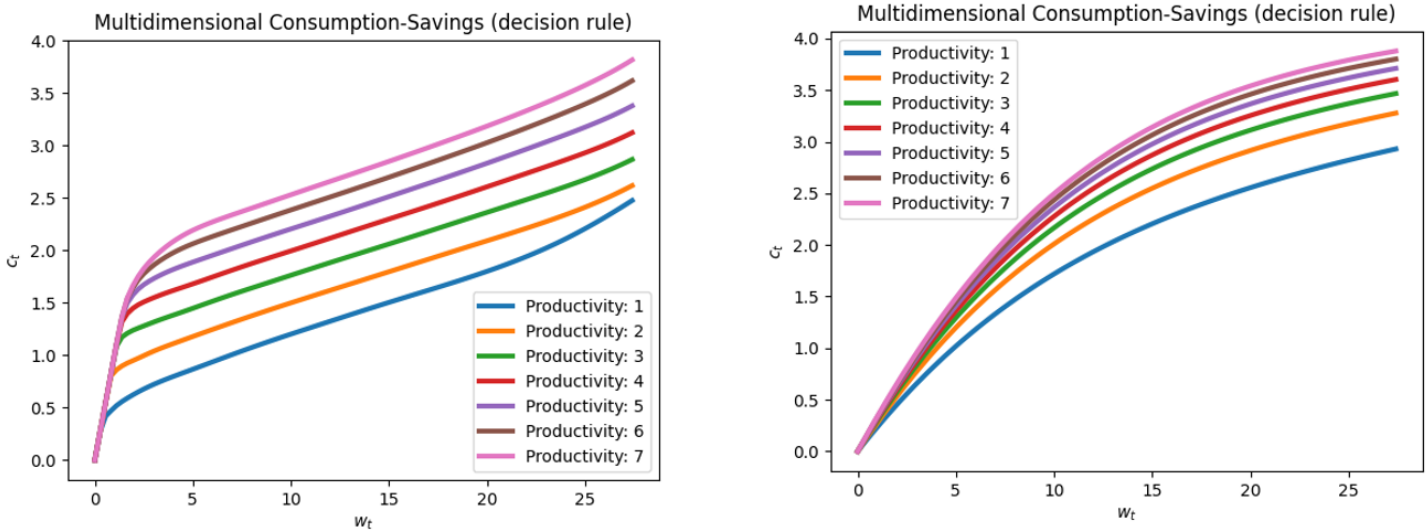


图: MMP using DEQN method (right side)

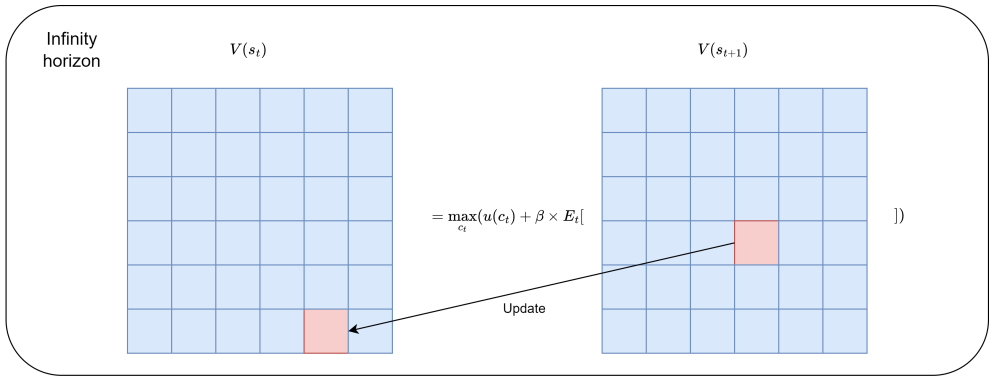
Moreover

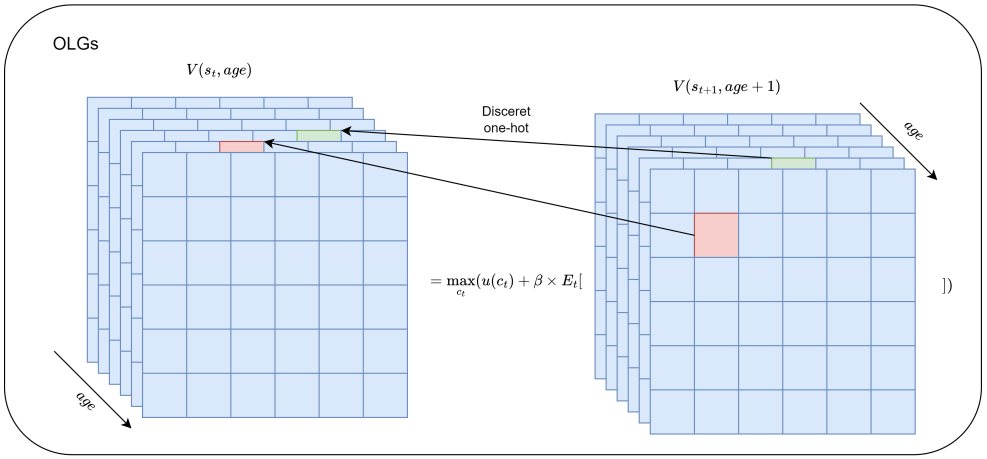
- MMP
  - Two-assets choice
  - More assets or action(labor) choice
  - Disceret choices
- DEQN
  - Traditional model(Brook Mirman 1972).
  - The Climate in Climate Economics (RES, 2023).
  - More on simulation?

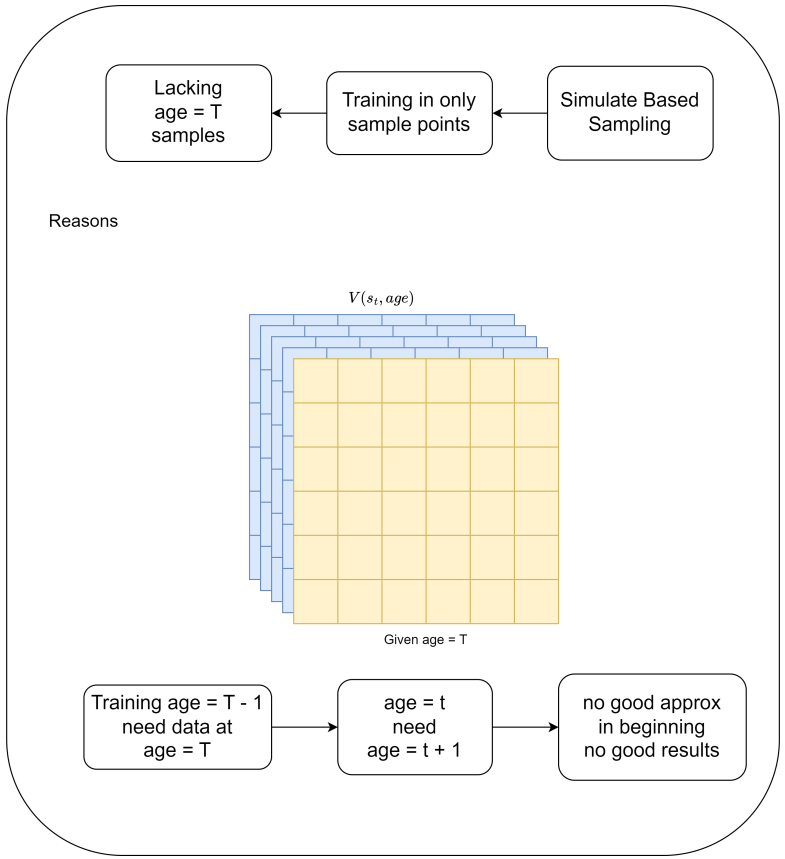
Try Value Residual Method in MMP on OLG models

- No!
- Bad Bad performance.
- Why?

Try Value Method in MMP on OLG models







谢谢大家