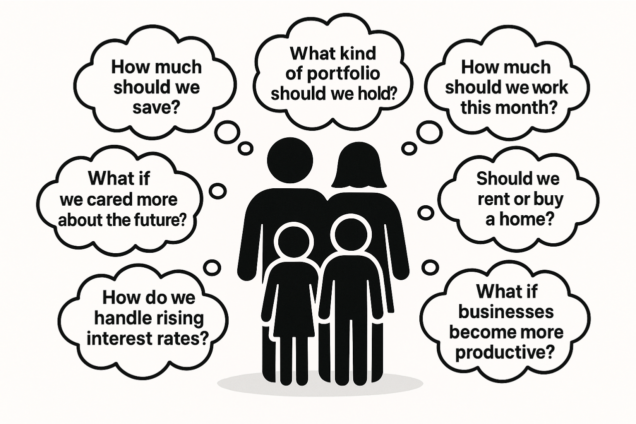
Generative Economic Modeling

Hanno Kase[1], Matthias Rottner[2], Fabio Stohler[3]

11th Annual Conference of The Society for Economic Measurement

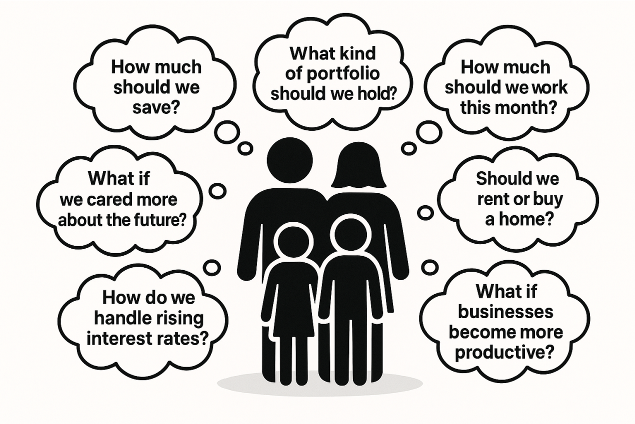
[1]European Central Bank
[2]Bank for International Settlements, Deutsche Bundesbank
[3]University of Bonn

The Curse of Dimensionality Limits Economic Analysis



Full model

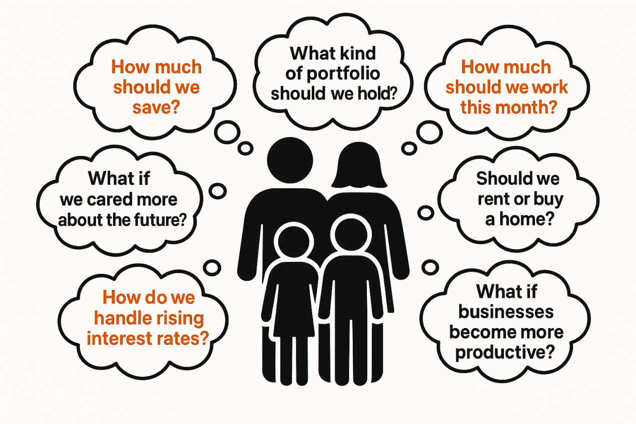
The Curse of Dimensionality Limits Economic Analysis



Full model

- Full model is intractable

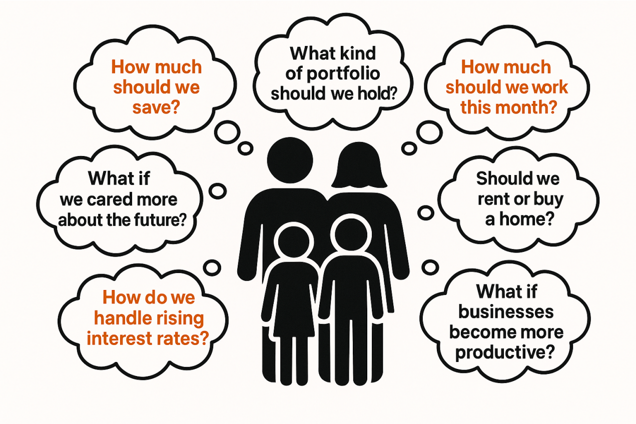
The Curse of Dimensionality Limits Economic Analysis



Simplified **submodel**

- Full model is intractable
- Forces researchers to use **submodel** with key model components
- Conventional methods (VFI & PFI) work well for simplified submodels

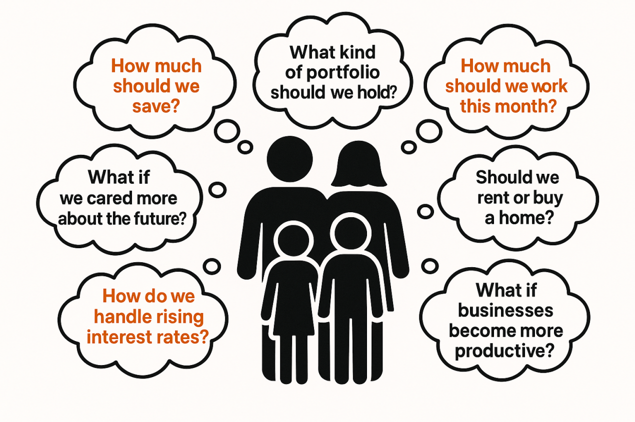
The Curse of Dimensionality Limits Economic Analysis



Simplified **submodel**

- Full model is intractable
- Forces researchers to use **submodel** with key model components
- Conventional methods (VFI & PFI) work well for simplified submodels
- **Question:** Can many simplified models approximate the full model?

The Curse of Dimensionality Limits Economic Analysis



Simplified **submodel**

- Full model is intractable
- Forces researchers to use **submodel** with key model components
- Conventional methods (VFI & PFI) work well for simplified submodels
- **Question:** Can many simplified models approximate the full model?

$\Rightarrow$ **G**enerative **E**conomic **M**odeling:

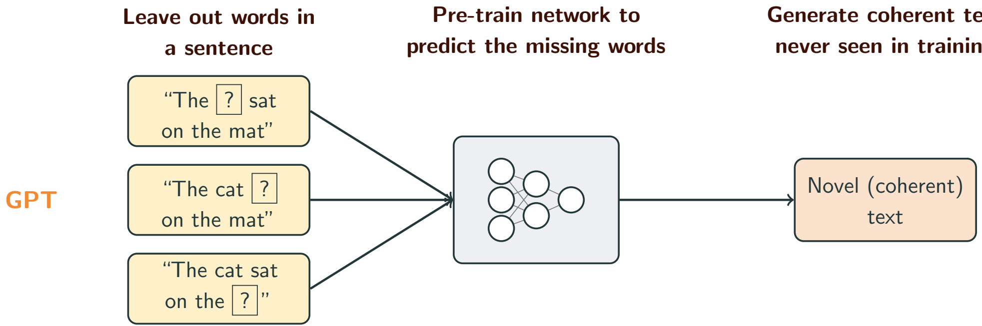
Conventional Methods + Deep Learning

From Generative AI to Generative Economic Modeling

Generative AI produces new content it was never explicitly shown, e.g. **LLMs**, which at their core are GPTs (**G**enerative **P**re-trained **T**ransformers)

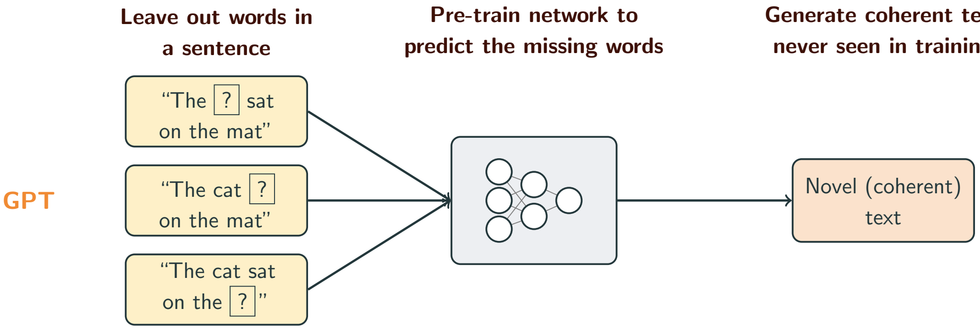
From Generative AI to Generative Economic Modeling

Generative AI produces new content it was never explicitly shown, e.g. **LLMs**, which at their core are GPTs (**G**enerative **P**re-trained **T**ransformers)



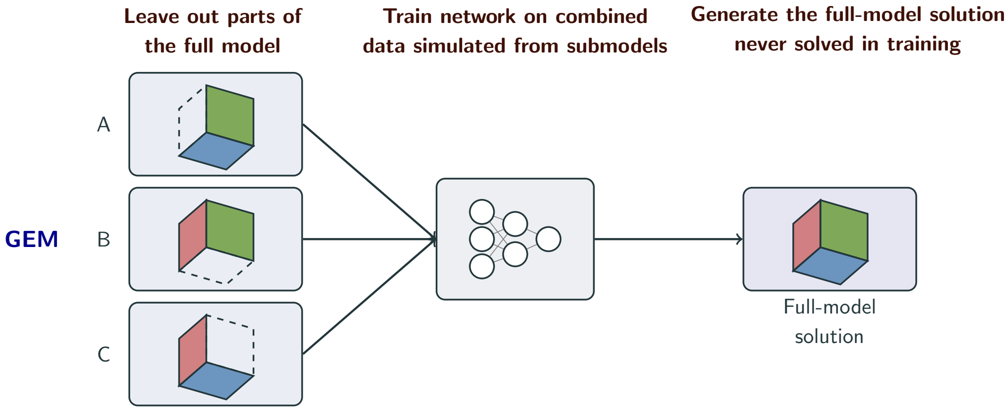
From Generative AI to Generative Economic Modeling

Generative AI produces new content it was never explicitly shown, e.g. **LLMs**, which at their core are GPTs (**G**enerative **P**re-trained **T**ransformers)



Question: Can we put the same **generative capacity** to work on economic models?

GEM Is Trained on Submodels like GPT Is Pre-Trained on Text



An **LLM** learns from text with words left out — and creates text it never saw.
GEM learns from models with parts left out — and solves a model it never solved.

Results and Roadmap for the Talk

- **Proof of concept** on a ladder of increasingly complex models
  - Asset pricing, analytical and nonlinear RBC, Krusell-Smith, and OLG economy
  - Prediction and Euler equation errors on par with a network trained on the full model
- **Application** to a Heterogeneous Agent New Keynesian (**HANK**) model
  - Nonlinear financial shocks with heterogeneous agents, 125 million state combinations
  - Interaction of multiple aggregate shocks with precautionary behavior
- **Why and when** the method works
  - Completeness and overlap of submodels, with a Taylor-expansion intuition
  - A built-in Euler equation error check, available even when no true solution exists

Location in the Literature

Deep learning to solve dynamic economic models

- Azinovic, Gaegauf, and Scheidegger (2022), Chen, Didisheim, and Scheidegger (2023), Fernández-Villaverde, Nuño, and Perla (2024), Han, Yang, and E (2021), Kase, Melosi, and Rottner (2022), Maliar, Maliar, and Winant (2021), Payne, Rebei, and Yang (2024), and Yang et al. (2025) $\Rightarrow$ **Generative approach**

Broader literature on conventional solution methods

- Judd (1998), Miranda and Fackler (2004), Fehr and Kindermann (2017), Heer and Maussner (2024) $\Rightarrow$ **Conventional solution methods & deep learning**

Generative AI for economic research

- Korinek (2023), Curti and Kazinnik (2023), Athey et al. (2024), Fish, Gonczarowski, and Shorrer (2024), Dell (2025) $\Rightarrow$ **Focus on economic modeling**

How does GEM work?

Illustrating Generative Economic Modeling

Solving a **generic model**

$$\mathbb{X}_t = f(\mathbb{X}_{t-1}, \nu_t \mid \Theta)$$

1. Solving **submodels** $\tilde{f}^{\,i}$

$$\tilde{\mathbb{X}}_t^a = \tilde{f}^{\,a}(\tilde{\mathbb{X}}_{t-1}^a, \tilde{\nu}_t^a \mid \tilde{\Theta}^{\,a})$$
$$\tilde{\mathbb{X}}_t^b = \tilde{f}^{\,b}(\tilde{\mathbb{X}}_{t-1}^b, \tilde{\nu}_t^b \mid \tilde{\Theta}^{\,b})$$
$$\tilde{\mathbb{X}}_t^c = \tilde{f}^{\,c}(\tilde{\mathbb{X}}_{t-1}^c, \tilde{\nu}_t^c \mid \tilde{\Theta}^{\,c})$$

Each **submodel** keeps different features

Solving a **generic model**

$$\mathbb{X}_t = f(\mathbb{X}_{t-1}, \nu_t \mid \Theta)$$

1. Solving **submodels** $\tilde{f}^{\,i}$
2. Simulating data and combining into one dataset

$\mathbb{X}_t$	ν_t^1	ν_t^2	ν_t^3	
$*$	$*$	$*$	0.0	(submodel a)
$*$	$*$	0.0	$*$	(submodel b)
$*$	0.0	$*$	$*$	(submodel c)

Simulate submodels and combine simulations in one dataset

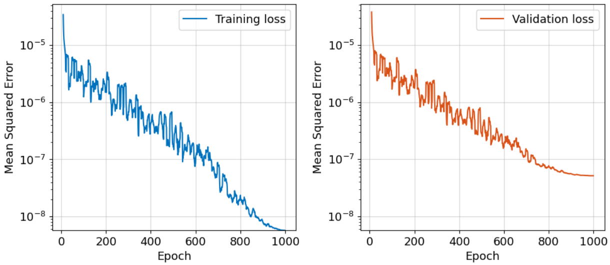
Illustrating Generative Economic Modeling

Solving a **generic model**

$$\mathbb{X}_t = f(\mathbb{X}_{t-1}, \nu_t \mid \Theta)$$

1. Solving **submodels** $\tilde{f}^{\,i}$
2. Simulating data and combining into one dataset
3. Training the neural network

$$f(\cdot) \approx \bar{f}_{DNN} = \arg\min_{\bar{W}} \mathbb{L}(\bar{W} \mid \bar{\mathbb{D}})$$



Train a neural network on the combined dataset $\bar{\mathbb{D}}$

Illustrating Generative Economic Modeling

Flow Chart

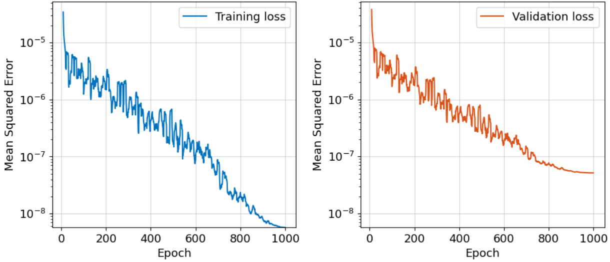
Solving a **generic model**

$$\mathbb{X}_t = f(\mathbb{X}_{t-1}, \nu_t \mid \Theta)$$

$$f(\cdot) \approx \bar{f}_{DNN} = \arg\min_{\bar{W}} \mathbb{L}(\bar{W} \mid \bar{\mathbb{D}})$$

1. Solving **submodels** $\tilde{f}^{\,i}$
2. Simulating data and combining into one dataset
3. Training the neural network

Trained network **generalizes to the full model** we never solve NN



Train a neural network on the combined dataset $\bar{\mathbb{D}}$

Proof of Concept

Proof of Concept Across Models and an Application

Proof of concept, a ladder of models of increasing complexity:

- **Asset pricing model**: analytically tractable with a closed-form solution
- **Analytical RBC**: representative agent with closed-form policy functions
- **Nonlinear RBC model**: state-dependent capital adjustment costs, solved globally
- **Heterogeneous-agent RBC model**: with the wealth distribution as a state

Application:

- **HANK model with financial frictions**:
  125 million state combinations with nonlinearity through financial frictions

Accuracy Across All Models: Euler Equation Errors

Analytical Nonlinear

Model	$\log_{10}$(MSE)	$\log_{10}$(MAE)	$\log_{10}$(EEE)
Asset Price	-4.85	-3.22	-3.15
Analytical RBC	-6.08	-3.47	-3.96
Nonlinear RBC	-3.30	-2.05	-2.46
Krusell-Smith	-7.10	-4.00	-2.11
Global HANK	—	—	-1.55

Mean Squared Error (MSE), Mean Absolute Error (MAE), Euler Equation Error (EEE)

- GEM remains reasonably accurate across the models we consider
- The **Euler equation error** is available even if **no true solution exists**

Is it generative?

An OLG Growth Model with Multiple Steady States

An OLG Growth Model with Multiple Steady States

A two-generation OLG economy:

- with log utility, the young save a **constant fraction** of output

$$K_{t+1} = \underbrace{\tfrac{\beta}{1+\beta}(1-\alpha)}_{\text{savings share}} \; Y(Z_t, K_t)$$

An OLG Growth Model with Multiple Steady States

A two-generation OLG economy:

- with log utility, the young save a **constant fraction** of output

$$K_{t+1} = \underbrace{\tfrac{\beta}{1+\beta}(1-\alpha)}_{\text{savings share}}\ Y\Big(\underbrace{\big(\textstyle\prod_j z_t^{(j)}\big)^{1/n}}_{\text{composite TFP}}, K_t\Big)$$

- three sectors draw binary shocks $\Rightarrow$ composite TFP $Z_t = \big(\prod_j z_t^{(j)}\big)^{1/n}$

An OLG Growth Model with Multiple Steady States

A two-generation OLG economy:

- with log utility, the young save a **constant fraction** of output

$$K_{t+1} = \underbrace{\tfrac{\beta}{1+\beta}(1-\alpha)}_{\text{savings share}}\ Z_t\, K_t^{\alpha}\ \underbrace{\Omega(Z_t K_t)}_{\text{externality}}$$

- three sectors draw binary shocks $\Rightarrow$ composite TFP $Z_t = \left(\prod_j z_t^{(j)}\right)^{1/n}$
- **productivity externality** $\Omega(Z_t K_t)$: efficiency jumps $A_L \to A_H$ once effective capital $Z_t K_t$ crosses a threshold $\Rightarrow$ **S-shaped** phase diagram

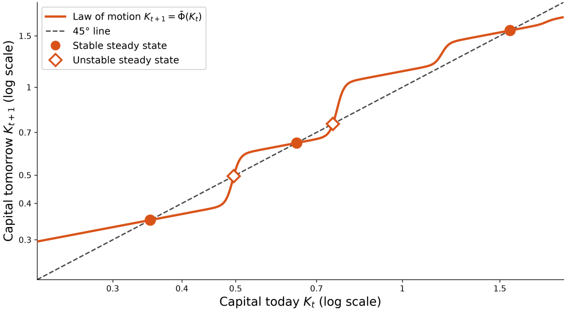
An OLG Growth Model with Multiple Steady States

A two-generation OLG economy:

- with log utility, the young save a **constant fraction** of output

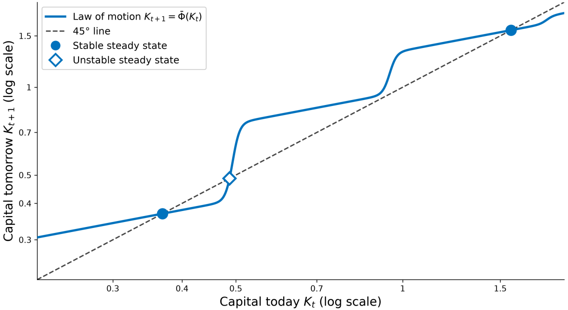
$$K_{t+1} = \underbrace{\tfrac{\beta}{1+\beta}(1-\alpha)}_{\text{savings share}} \; Z_t \, K_t^{\alpha} \; \underbrace{\Omega(Z_t K_t)}_{\text{externality}}$$

- three sectors draw binary shocks $\Rightarrow$ composite TFP $Z_t = \left(\prod_j z_t^{(j)}\right)^{1/n}$
- **productivity externality** $\Omega(Z_t K_t)$: efficiency jumps $A_L \to A_H$ once effective capital $Z_t K_t$ crosses a threshold $\Rightarrow$ **S-shaped** phase diagram



Complete three-sector economy: **five** stochastic steady states (3 stable, 2 unstable)

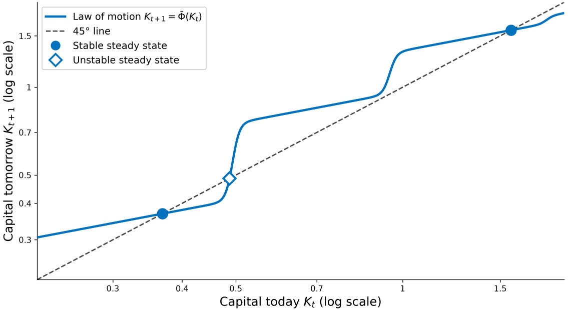
Submodels Never See All the Steady States



Two-sector **submodel**: only **three** steady states

- Submodel exposes network to **only three** steady states
- Two extra interior steady states are **never in the training data**

Submodels Never See All the Steady States



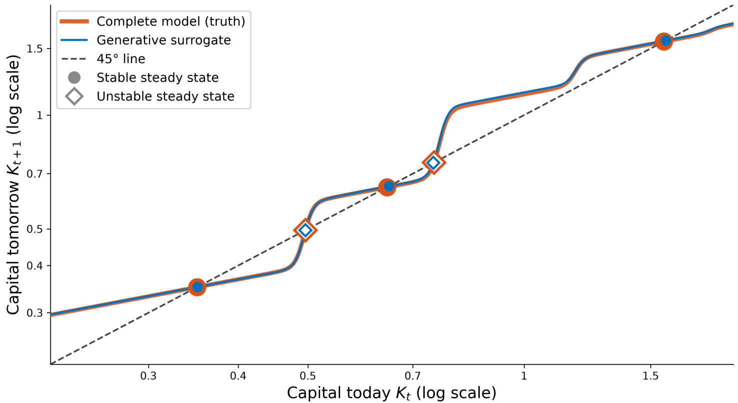
Two-sector **submodel**: only **three** steady states

- Submodel exposes network to **only three** steady states
- Two extra interior steady states are **never in the training data**

$\Rightarrow$ As **generative AI** produces outputs it never saw, GEM must recover steady states it never saw

The Surrogate Recovers the Full Law of Motion

The Surrogate Recovers the Full Law of Motion



- Train surrogate **on the two-sector submodels**, each with just three steady states
- The **approximated** law of motion (**generative**) matches the **true** one (**complete**), including the **two extra steady states** it never saw in training

Why & When It Works

Why Does It Work?

Taylor expansion: Stack states $\mathbb{X}_{t-1}$ and shocks ν_t into features $s_{j,t}$, and expand:

$$f(\mathbb{X}_{t-1}, \nu_t \mid \Theta) \approx f(\bar{\mathbb{X}}) + \sum_j \underbrace{f_j\, s_{j,t}}_{\text{recovered}} + \frac{1}{2} \underbrace{\sum_{(j,k)\ \text{co-occur}} f_{jk}\, s_{j,t}\, s_{k,t}}_{\text{recovered via overlap}}$$

$$+ \frac{1}{2} \underbrace{\sum_{(j,k)\ \text{split}} f_{jk}\, s_{j,t}\, s_{k,t}}_{\text{missed: residual}} + \dots$$

with coefficients $f_j = \partial f / \partial s_j$ and interaction coefficients $f_{jk} = \partial^2 f / \partial s_j \partial s_k$, at $\bar{\mathbb{X}}$.

- Submodels $\tilde{f}^{\,i}$ pin down f_j and interaction f_{jk} among features it contains
- Overlap lets most pairs (j, k) co-occur in some submodel, so their f_{jk} is recovered
- Only cross-interactions are missed, and in most economic models these are **small**
- The residual error **shrinks** as the number of features per submodel grows

When Does the Methodology Apply?

Two requirements on the set of submodels:

- Submodels are general-equilibrium models, solvable with conventional methods
- **Completeness**: every feature appears in **at least one** submodel
- **Overlap**: features in **at least two** submodels, so network can stitch them together

When Does the Methodology Apply?

Two requirements on the set of submodels:

- Submodels are general-equilibrium models, solvable with conventional methods
- **Completeness**: every feature appears in **at least one** submodel
- **Overlap**: features in **at least two** submodels, so network can stitch them together

Built-in validation, even when the true model is unsolvable:

- The **Euler equation error** gives a direct, ex-post accuracy check
- Also guides **design and weighting** of submodels and choice of architecture

Generative Economic Modeling: Classical Methods Meet Deep Learning

Core Steps of GEM:

1. Solve a collection of **simplified models**
2. Simulate **synthetic data** from each specification
3. Train a **neural network** on the aggregated dataset

Takeaways for applied use:

- **Extends** existing solution methods, so no new toolkit is required
- Builds directly on established **conventional methods**
- Neural networks used **only for training** on simulated data
- Can **pretrain** networks, providing a general starting point for solving other models

Thank you!

github.com/Fabio-Stohler/Generative-Economic-Modeling

Literature I

Athey, S., Imbens, G. W., Metzger, J., & Munro, E. (2024). **Using wasserstein generative adversarial networks for the design of monte carlo simulations.** *Journal of Econometrics*, *240*.

Azinovic, M., Gaegauf, L., & Scheidegger, S. (2022). **Deep equilibrium nets.** *International Economic Review*, *63*(4), 1471–1525.

Canzoneri, M. B., Cumby, R. E., & Diba, B. T. (2007). **Euler equations and money market interest rates: A challenge for monetary policy models.** *Journal of Monetary Economics*, *54*(7), 1863–1881.

Chen, H., Didisheim, A., & Scheidegger, S. (2023). **Deep surrogates for finance: With an application to option pricing.** *Available at SSRN 3782722*.

Literature II

Curti, F., & Kazinnik, S. (2023). **Let's face it: Quantifying the impact of nonverbal communication in fomc press conferences.** *Journal of Monetary Economics*, *139*, 110–126.

Dell, M. (2025). **Deep learning for economists.** *Journal of Economic Literature*, *63*(1), 5–58.

Fernández-Villaverde, J., Nuño, G., & Perla, J. (2024). **Taming the curse of dimensionality: Quantitative economics with deep learning (tech. rep.).** National Bureau of Economic Research.

Fish, S., Gonczarowski, Y. A., & Shorrer, R. I. (2024). **Algorithmic collusion by large language models.** *arXiv e-prints*, Article arXiv:2404.00806, arXiv:2404.00806.

Han, J., Yang, Y., & E, W. (2021). **Deepham: A global solution method for heterogeneous agent models with aggregate shocks (tech. rep.).** arXiv.

Literature III

Heer, B., & Maussner, A. (2024). **Dynamic general equilibrium modeling.** Springer.

Judd, K. L. (1998). **Numerical methods in economics.**

Kase, H., Melosi, L., & Rottner, M. (2022). **Estimating nonlinear heterogeneous agents models with neural networks (tech. rep. No. DP17391).** CEPR Discussion Paper No. 17391. Paris & London, CEPR Press.

King, R. G., Plosser, C. I., & Rebelo, S. T. (1988). **Production, growth and business cycles.** *Journal of Monetary Economics*, *21*(2-3), 195–232.

Korinek, A. (2023). **Generative ai for economic research: Use cases and implications for economists.** *Journal of Economic Literature*, *61*(4), 1281–1317.

Maliar, L., Maliar, S., & Winant, P. (2021). **Deep learning for solving dynamic economic models..** *Journal of Monetary Economics*, *122*, 76–101.

Literature IV

Miranda, M. J., & Fackler, P. L. (2004). **Applied computational economics and finance.** MIT press.

Payne, J., Rebei, A., & Yang, Y. (2024). **Deep learning for search and matching models.**

Yang, Y., Wang, C., Schaab, A., & Moll, B. (2025). **Structural reinforcement learning for heterogeneous agent macroeconomics.**

Appendix

Asset Pricing: Full Analytical Solution

Back

Analytical solution as an expression of conditional mean and variance:

$$q_t = \beta \exp\Big(-(a_{11} + \gamma a_{21})\,\pi_t \; - \; (a_{12} + \gamma a_{22})\,\Delta c_t \\ + \tfrac{1}{2}\big[\underbrace{(\eta_{11} + \gamma\eta_{21})^2}_{\epsilon_t^a} + \underbrace{(\eta_{12} + \gamma\eta_{22})^2}_{\epsilon_t^\zeta} + \underbrace{(\eta_{13} + \gamma\eta_{23})^2}_{\epsilon_t^\mu}\big]\Big)$$

All Submodel Solutions

Back

Each submodel drops exactly one shock:

$$q_t^{\setminus a} = \beta \exp\left(-(a_{11}+\gamma a_{21})\,\pi_t \;-\; (a_{12}+\gamma a_{22})\,\Delta c_t + \frac{1}{2}\left[(\eta_{12}+\gamma\eta_{22})^2 + (\eta_{13}+\gamma\eta_{23})^2\right]\right)$$

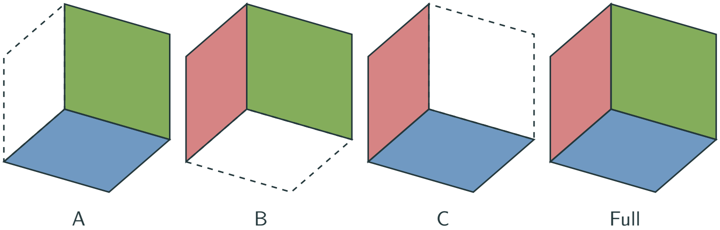
$$q_t^{\setminus \zeta} = \beta \exp\left(-(a_{11}+\gamma a_{21})\,\pi_t \;-\; (a_{12}+\gamma a_{22})\,\Delta c_t + \frac{1}{2}\left[(\eta_{11}+\gamma\eta_{21})^2 + (\eta_{13}+\gamma\eta_{23})^2\right]\right)$$

$$q_t^{\setminus \mu} = \beta \exp\left(-(a_{11}+\gamma a_{21})\,\pi_t \;-\; (a_{12}+\gamma a_{22})\,\Delta c_t + \frac{1}{2}\left[(\eta_{11}+\gamma\eta_{21})^2 + (\eta_{12}+\gamma\eta_{22})^2\right]\right)$$

- coefficients on π_t and Δc_t learned from each partial dataset
- coefficients from shocks learned as a shift from partial datasets with that shock

Overlap of Simplified Models

Back

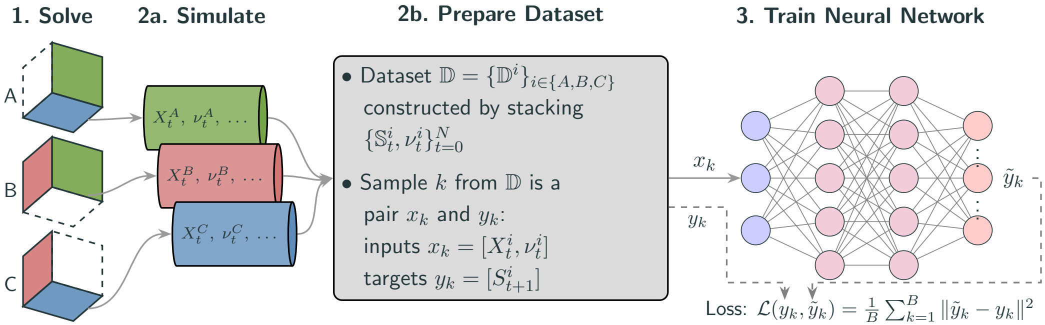


Graphical illustration for $n_\nu = 3$ and $n_{\tilde{\nu}} = 2$

- Model with shocks ν of length n_ν, simplified models with shocks $\tilde{\nu}$ of length $n_{\tilde{\nu}}$

$$\binom{n_\nu}{n_{\tilde{\nu}}} = \frac{n_\nu!}{n_{\tilde{\nu}}!(n_\nu - n_{\tilde{\nu}})!}$$

Illustration of Our Approach



Flow chart of the generative economic modeling method

Training Setup and Evaluation Criteria

Back

Train one NN on the full model and another on the combined simulated data.

- **Training Setup:**
  - Network architecture: 5 hidden layers, each with 128 neurons
  - Activation function: CELU
  - Optimizer: AdamW, minimizing mean squared error (MSE)
  - Learning rate: cosine annealing schedule, decaying from 10^{-3} to 10^{-7}

- **Evaluation Metric: Euler Equation Error (EEE)**

$$EEE = u'(\underbrace{\bar{f}^{c}_{DNN}(\mathbb{X}_t, \nu_t|\Theta)}_{c_t}) - \beta \mathbb{E}_t \left[\underbrace{\bar{f}^{R}_{DNN}(\mathbb{X}_{t+1}, \nu_{t+1}|\Theta)}_{R_{t+1}} u'(\underbrace{\bar{f}^{c}_{DNN}(\mathbb{X}_{t+1}, \nu_{t+1}|\Theta)}_{c_{t+1}}) \right]$$

Proposition

If all households are ex-ante and ex-post identical, depreciation is deterministic and full, $\delta(u_t) = 1$*, capacity utilization is fixed at* $u_t = 1$*, there are no capital adjustment costs* $\phi = 0$*, the discount factor shock is inactive* $\sigma_\zeta^2 = 0$*, and per period felicity is of King, Plosser, and Rebelo (1988) (KPR)-form given by* $u(C_t, N_t) = \ln C_t - \omega \frac{N_t^{1+\gamma}}{1+\gamma}$*. Then the policy functions of the representative household are*

$$N_t(\mu_t) = \left[\frac{\mu(1-\tau^L)(1-\alpha)}{\mu_t \omega (1+\tau^C)(\mu - (1-\tau^K)\alpha\beta)} \right]^{\frac{1}{1+\gamma}}, \tag{1}$$

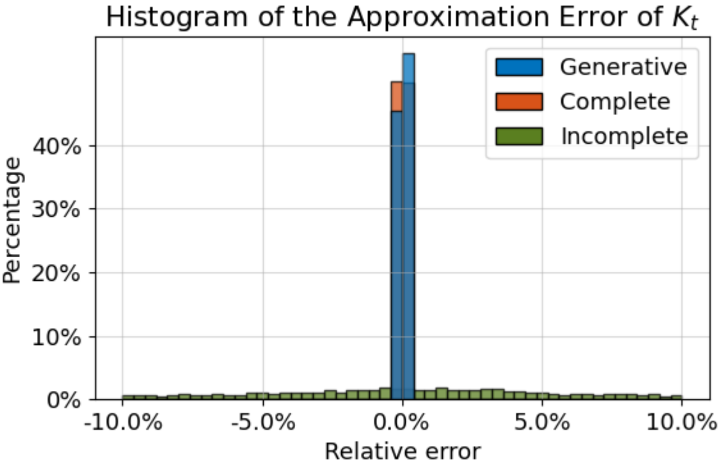
$$C_t(K_t, A_t, Z_t, \mu_t) = \left(1 - \frac{\alpha\beta}{\mu_t}\right) Y_t\left(K_t, A_t, Z_t, N_t(\mu_t)\right), \tag{2}$$

$$\text{and} \quad K_{t+1}(K_t, A_t, Z_t, \mu_t) = \frac{\alpha\beta}{\mu_t} Y_t\left(K_t, A_t, Z_t, N_t(\mu_t)\right), \tag{3}$$

where $Y_t(K_t, A_t, Z_t, N_t(\mu_t))$ *denotes the Cobb-Douglas production function. Given the policy functions of the household, the prices in the economy can be determined by the first-order conditions of the firm.*

Generative Economic Modeling with Analytical Solution: Errors

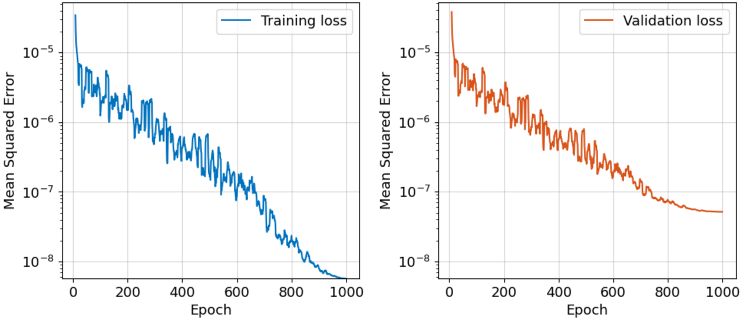
Back



Error distribution of surrogate model for analytical model

Generative Economic Modeling with Analytical Solution: Loss

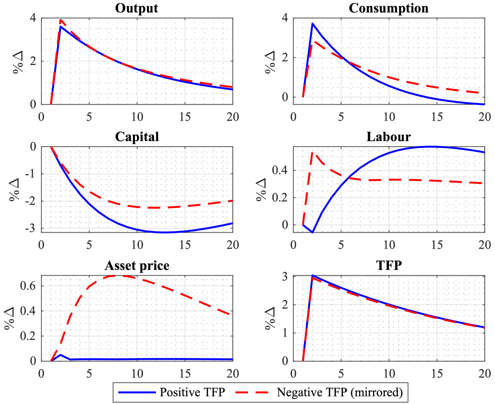
Back



Loss in training and validation data over iterations

Generative Economic Modeling with Nonlinear Rep. Agent Model: IRFs

Back

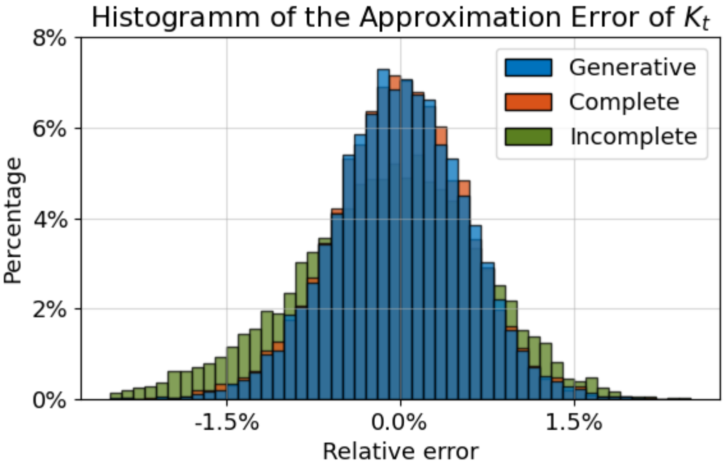


IRFs and nonlinear propagation of the TFP shock

- Nonlinear medium-sized RBC with state-dependent capital adjustment costs
- Model is solved with global solution method (time iteration)
- Generalized IRFs illustrate nonlinear effect of capital adjustment costs

Generative Economic Modeling with Nonlinear Rep. Agent Model: Errors

Back



Error distribution of surrogate model for nonlinear Rep. Agent

Analytical Asset Pricing Model as a Benchmark

Analytically tractable asset pricing model of Canzoneri, Cumby, and Diba (2007):

$$q_t = \beta \mathbb{E}_t \Big[\exp\big(-\pi_{t+1} - \gamma \Delta c_{t+1} \big) \Big]$$

Intuition: today's bond price = expected discounted payoff

Analytical Asset Pricing Model as a Benchmark

Analytically tractable asset pricing model of Canzoneri, Cumby, and Diba (2007):

$$q_t = \beta \mathbb{E}_t \Big[\exp\big(- \pi_{t+1} - \gamma \Delta c_{t+1} \big) \Big]$$

Intuition: today's bond price = expected discounted payoff

VAR(1) for variables $y_t = [\pi_t, \Delta c_t]'$ with shocks $\epsilon_t = [\epsilon_t^a, \epsilon_t^\zeta, \epsilon_t^\mu]'$:

$$y_t = A y_{t-1} + \eta \epsilon_t$$

Analytical Asset Pricing Model as a Benchmark

Analytically tractable asset pricing model of Canzoneri, Cumby, and Diba (2007):

$$q_t = \beta \mathbb{E}_t \Big[\exp\big(-\pi_{t+1} - \gamma \Delta c_{t+1} \big) \Big]$$

Intuition: today's bond price = expected discounted payoff

VAR(1) for variables $y_t = [\pi_t, \Delta c_t]'$ with shocks $\epsilon_t = [\epsilon_t^a, \epsilon_t^\zeta, \epsilon_t^\mu]'$:

$$y_t = A y_{t-1} + \eta \epsilon_t$$

$\Rightarrow$ the model has a **closed-form solution**, our laboratory benchmark

Full Solution

Analytical Asset Pricing Model as a Benchmark

Analytically tractable asset pricing model of Canzoneri, Cumby, and Diba (2007):

$$q_t = \beta \mathbb{E}_t \Big[\exp\big(- \pi_{t+1} - \gamma \Delta c_{t+1} \big) \Big]$$

Intuition: today's bond price = expected discounted payoff

VAR(1) for variables $y_t = [\pi_t, \Delta c_t]'$ with shocks $\epsilon_t = [\epsilon_t^a, \epsilon_t^\zeta, \epsilon_t^\mu]'$:

$$y_t = A y_{t-1} + \eta \epsilon_t$$

$\Rightarrow$ the model has a **closed-form solution**, our laboratory benchmark

Full Solution

Now **assume it were too complicated to solve**

Solving Submodels Instead of the Full Model

All Submodels

Each **submodel** switches *one* shock off, e.g. dropping shock a:

$$q_t^{\setminus a} = \beta \exp\Big(-(a_{11} + \gamma a_{21})\,\pi_t \;-\; (a_{12} + \gamma a_{22})\,\Delta c_t + \tfrac{1}{2}\big[\underbrace{(\eta_{11} + \gamma\eta_{21})^2}_{\text{shock } a \text{ off } = 0} + (\eta_{12} + \gamma\eta_{22})^2 + (\eta_{13} + \gamma\eta_{23})^2 \big] \Big)$$

Solving Submodels Instead of the Full Model

Each **submodel** switches *one* shock off, e.g. dropping shock a:

$$q_t^{\setminus a} = \beta \exp\Big(-(a_{11} + \gamma a_{21})\,\pi_t \;-\; (a_{12} + \gamma a_{22})\,\Delta c_t$$
$$+ \tfrac{1}{2}\Big[\underbrace{(\eta_{11} + \gamma\eta_{21})^2}_{\text{shock } a \text{ off } = 0} + (\eta_{12} + \gamma\eta_{22})^2 + (\eta_{13} + \gamma\eta_{23})^2 \Big] \Big)$$

- coefficients on π_t and Δc_t are learned from *every* submodel
- shock terms are learned wherever that shock is switched on
- the network combines them to recover the **full model it never saw**

Illustrating GEM: Asset Pricing Example

Step-by-Step

Solving **Asset Pricing** model:

1. Solving (sub)models

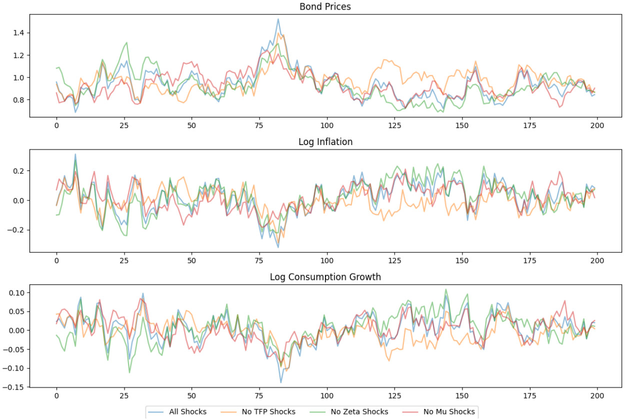
$$q_t^{\backslash a} = \ldots$$
$$q_t^{\backslash \zeta} = \ldots$$
$$q_t^{\backslash \mu} = \ldots$$

Illustrating GEM: Asset Pricing Example

Step-by-Step

Solving **Asset Pricing** model:

1. Solving (sub)models
2. Simulating data



Simulation of economies with different combinations of aggregate shocks

Illustrating GEM: Asset Pricing Example

Step-by-Step

Solving **Asset Pricing** model:

1. Solving (sub)models
2. Simulating data and combining into one dataset

q_t	π_t	Δc_t	ϵ_t^a	ϵ_t^ζ	ϵ_t^μ
0.99	-0.02	-0.01	-0.12	1.04	0.0
1.06	-0.04	0.02	1.63	1.85	0.0
⋮		⋮		⋮	
1.01	-0.08	0.02	-0.12	0.0	-1.73
0.99	-0.03	-0.01	1.63	0.0	0.45
⋮		⋮		⋮	
1.06	-0.11	0.01	0.0	-1.0	-1.72
0.94	-0.01	-0.07	0.0	-0.39	-0.86

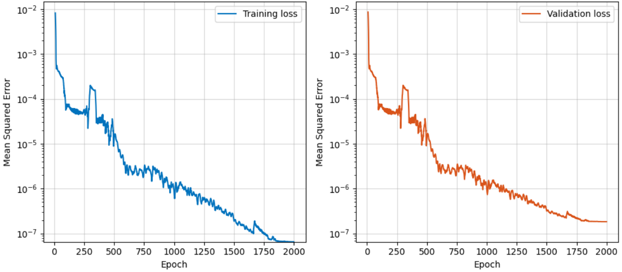
Simulation snapshots from combined dataset for q_t, π_t, Δc_t, ϵ_t^a ϵ_t^ζ, and ϵ_t^μ

Illustrating GEM: Asset Pricing Example

Step-by-Step

Solving **Asset Pricing** model:

1. Solving (sub)models
2. Simulating data and combining into one dataset
3. Training the neural network



Loss function from training the NN

Illustrating GEM: Asset Pricing Example

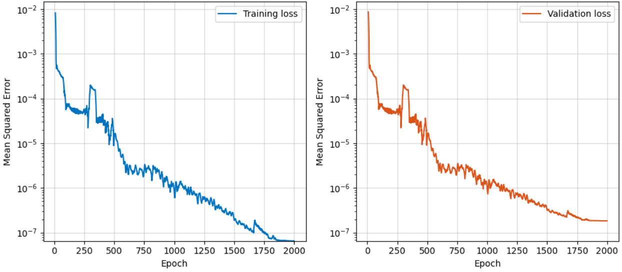
Step-by-Step

Solving **Asset Pricing** model:

1. Solving (sub)models
2. Simulating data and combining into one dataset
3. Training the neural network

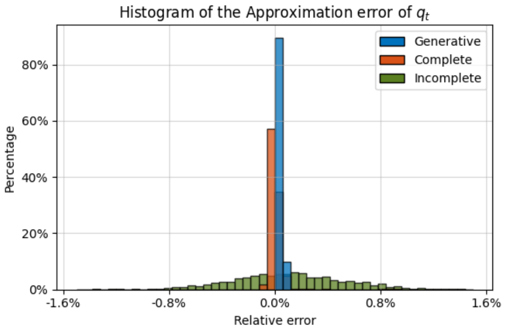
Evaluate against **NNs** trained on data **generated by true model**

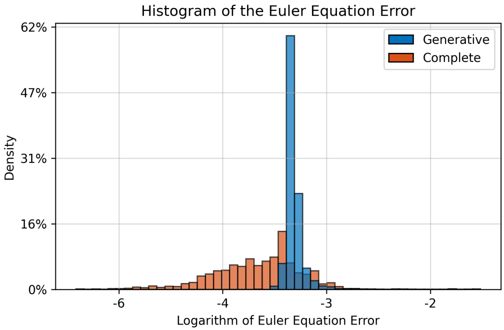
NN Specification   Evaluation Criteria



Loss function from training the NN

Evaluating Generative Economic Modeling: Predicted Values vs. Actual Values





Prediction error distribution and Euler Equation Errors

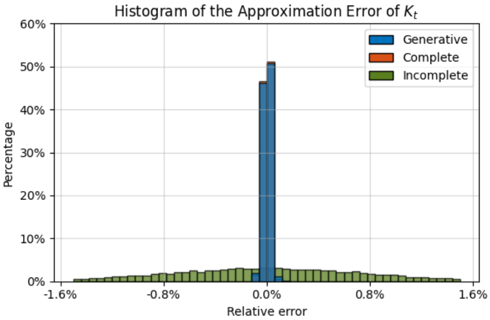
- **Generative**: extrapolates with a NN trained on simulated data from submodels
- **Complete**: Train a NN on simulated data from true model

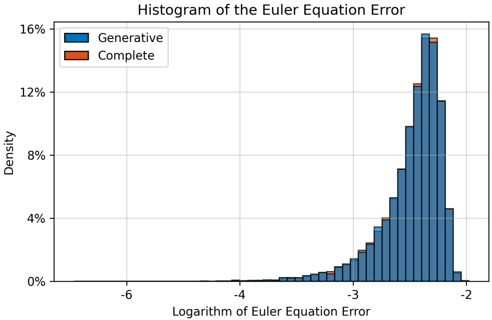
Proof of Concept Using an Extended Stochastic RBC Model

RBC Model variants with increasing complexity

1. Baseline: tractable, analytically solvable RBC variant
2. Medium-scale: includes nonlinearities in capital adjustment cost friction
3. Heterogeneous agents: partial insurance against income risk

Evaluating Generative Economic Modeling: Heterogeneous Agent Model

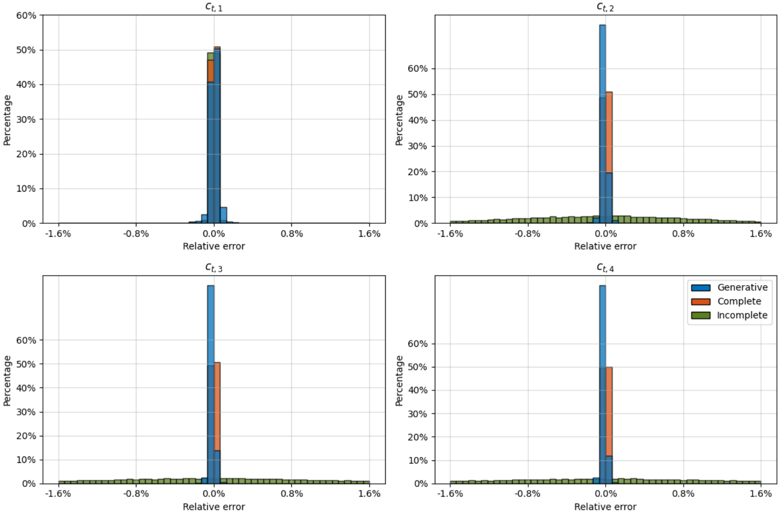




Graphical evaluation of model fit:

Prediction error distribution and Euler Equation Errors

Errors for Individual Policy Functions



Comparison of error distributions for consumption function at individual grid points

The Full OLG Model: From Primitives to the Law of Motion

Primitives.

Households (two generations, log utility). The young supply one unit of labor, earn the wage w_t, and save a constant fraction of income; the old consume savings plus return:

$$s_t = \tfrac{\beta}{1+\beta}\, w_t.$$

Production (n sectors, one composite shock). Each sector draws an independent binary shock $z_t^{(j)} \in \{z_L, z_H\}$, combined into a composite TFP via the geometric mean,

$$Z_t = \Big(\textstyle\prod_j z_t^{(j)}\Big)^{1/n}, \qquad Y_t = Z_t\, \Omega(Z_t K_t)\, K_t^{\alpha}.$$

Externality (S-shaped, Hill). Aggregate efficiency depends on effective capital $Z_t K_t$:

Equilibrium and the law of motion.

Competitive firms pay labor its marginal product, so the wage is the labor share of output, $w_t = (1-\alpha)\, Y_t$. Capital-market clearing equates the savings of the young to next-period capital, $K_{t+1} = s_t$. Substituting,

$$K_{t+1} = \lambda\, Z_t\, K_t^{\alpha}\, \Omega(Z_t K_t), \qquad \lambda = \frac{\beta(1-\alpha)}{1+\beta}.$$

Averaging over the shocks gives the **mean dynamics**

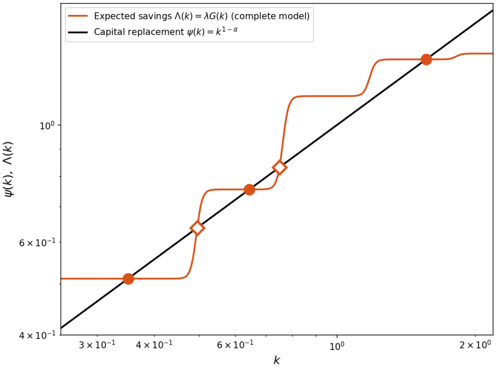
$$\bar{\Phi}(K) = \lambda\, K^{\alpha} G(K), \quad G(K) = \textstyle\sum_s \pi(s)\, Z(s)\, \Omega(Z(s)K$$

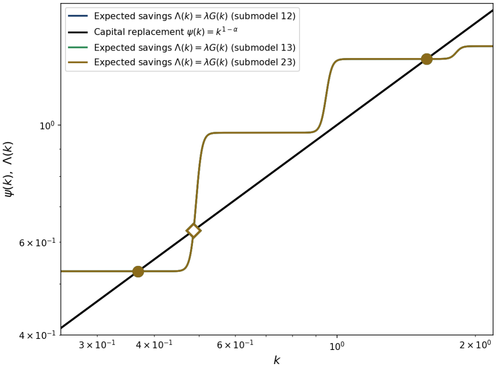
and stochastic steady states solve $\bar{\Phi}(K) = K$.

Multiplicity scales with the number of sectors.

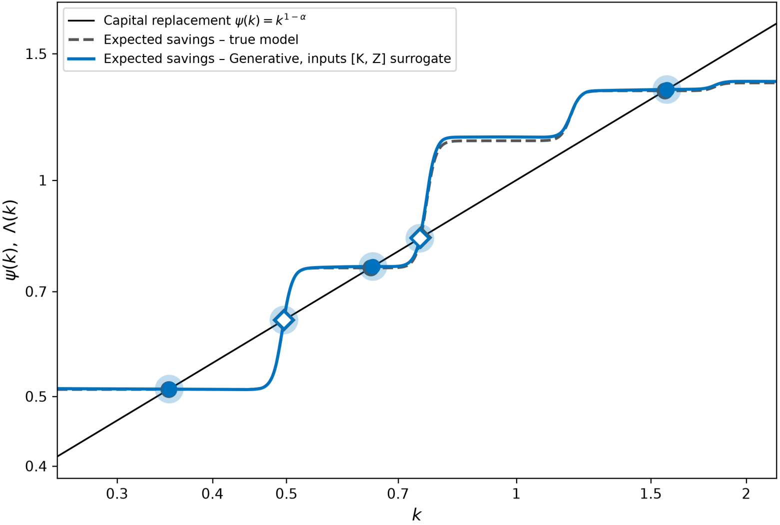
Sectors	Steady states

OLG: Law of Motion, Full Model vs. Submodels





OLG: Mean Dynamics of the Generative Surrogate



A HANK Model with Financial Frictions

The setup, briefly:

- **Firms** produce with labor under monopolistic competition, subject to a **cash-in-advance constraint** (TFP & financial shocks)
- **Households** consume, save, supply labor; face **income risk** (preference shocks)
- **Government** levies taxes, issues bonds, follows a Taylor rule (monetary shocks)

A HANK Model with Financial Frictions

The setup, briefly:

- **Firms** produce with labor under monopolistic competition, subject to a **cash-in-advance constraint** (TFP & financial shocks)
- **Households** consume, save, supply labor; face **income risk** (preference shocks)
- **Government** levies taxes, issues bonds, follows a Taylor rule (monetary shocks)

Why this model, and why it is otherwise unsolvable:

- **125 million state combinations**: a global solution is typically infeasible
- The two payoffs we are after:
  - (i) financial shocks **amplify nonlinearly** in their size
  - (ii) more aggregate risk $\Rightarrow$ stronger **precautionary saving** $\Rightarrow$ dampened transmission
- Is heterogeneity relevant for **optimal policy?**

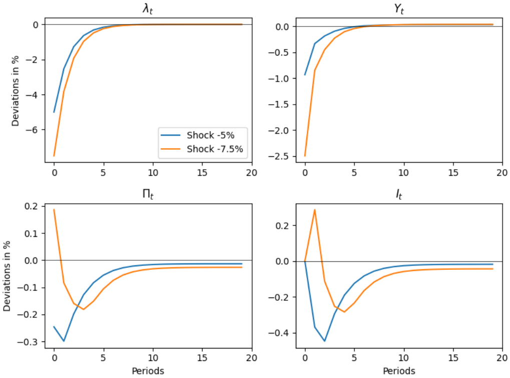
Accuracy of Solution of Financial HANK Model

Model	$\log_{10}(\text{MSE})$	$\log_{10}(\text{MAE})$	$\log_{10}(\text{EEE})$
Asset Price	-4.85	-3.22	-3.15
Analytical RBC	-6.08	-3.47	-3.96
Nonlinear RBC	-3.30	-2.05	-2.46
Krusell-Smith	-7.10	-4.00	-2.09
Global HANK	—	—	-1.55

Mean Squared Error (MSE), Mean Absolute Error (MAE), and Euler Equation Error (EEE). —: not available, the full model cannot be solved.

- No true model to benchmark against, which is the whole point of the method
- The Euler Equation Error is the only accuracy check available

Scenario Analysis: Financial Shocks Trigger Nonlinear Recessions



Generalized impulse responses to a financial shock of varying size

Acts as a supply shock:

- Financial shock limits firms' ability to finance inputs $\Rightarrow$ output falls
- Inflation rises, pushing up nominal and real rates $\Rightarrow$ demand contracts further

Global solution needed for the constraint:

- Economy's response is **nonlinear** in the size of the financial shock