Spooky Boundaries at a Distance:

Inductive Bias and Dynamic Macroeconomic Models

Mahdi Ebrahimi Kahou[1] Jesús Fernández-Villaverde[2]

Sebastián Gómez-Cardona[3] Jesse Perla[4] Jan Rosa[4]

August 23, 2025

Abstract

When studying the short-run dynamics of economic models, it is crucial to consider boundary conditions that govern long-run forward-looking behavior, such as transversality conditions. We demonstrate that machine learning (ML), specifically deep learning, can automatically satisfy these conditions due to its inherent inductive bias toward finding flat solutions to functional equations. This characteristic enables ML algorithms to solve for transition dynamics, ensuring that long-run boundary conditions are approximately met. ML can even select the correct path in cases of steady-state multiplicity.

Keywords: Machine learning; inductive bias; rational expectations; transitional dynamics; transversality.

JEL codes: **C1, E1.**

[1]Bowdoin College, [2]University of Pennsylvania, [3]Morningstar, and [4]University of British Columbia, Vancouver School of Economics. We want to thank Marlon Azinovic, Misha Belkin, Jess Benhabib, Anmol Bhandari, Doga Bilgin, Matthew Botsch, Rachael Childers, Keyvan Eslami, Deni Goktas, Amy Greenwald, Loukas Karabarbounis, Hiro Kasahara, Hanno Kase, Kyle Herkenhoff, Kevin Leyton-Brown, Pierlauro Lopez, Vadim Marmer, Galo Nuño, Jonathan Payne, Michael Peters, Geoff Pleiss, Mark Schmidt, and Kevin Song. We also would like to thank Annabella Stoll-Dansereau for outstanding research assistance. All the code for the paper is available at https://github.com/HighDimensionalEconLab/transversality.

1 Introduction

Steady states play a paradoxical role in dynamic economic models. They are approached only asymptotically, and short-run dynamics often differ sharply from steady-state values. Yet, steady states are essential for solving these models: long-run assumptions grounded in economic reasoning, such as transversality conditions, are typically required to ensure internally consistent short-run dynamics.

However, imposing these conditions often complicates solving models, particularly those with many dimensions. Generally, these conditions require solving the model over a broad range of possible values of the state variables. For example, recursive formulations necessitate accurate solutions for arbitrary values of the state variables, even though the solution may only be relevant from a single initial condition, or require the careful design of a hypercube where the stable solution of the model lies.

This paper presents two key contributions. First, we show that long-run boundary conditions can be met without strictly enforcing them as constraints on the model's dynamics. In particular, we illustrate how machine learning (ML) methods allow us to run short-run simulations while still satisfying boundary conditions, even in cases with multiple steady states, thanks to the inductive bias of ML algorithms. Inductive bias, a concept widely discussed in the philosophy of science and ML, reflects a preference for simplicity and parsimony (similar to Ockham's razor) when fitting a general model with limited observations.

Inductive bias is also closely related to the double descent phenomenon, which describes the ability of highly parameterized models, such as deep neural networks (DNNs), to escape the classical bias-variance trade-off and achieve minimal errors in fitting and forecasting (see

Belkin et al., 2019, Belkin, 2021, and Wilson, 2025). We show how DNNs, with four orders of magnitude more parameters than grid points (the "data" in this context), produce solutions to dynamic economic models with minimal errors and that adhere to long-run constraints.

Interestingly, these highly parameterized DNNs often yield "simple" solutions, where "simple" in a functional space is defined more subtly than by merely counting parameters. Instead, we want to think of "simple" as a function that is small with respect to an appropriate functional seminorm.[1]

Why is having a "simple" solution important? Assumptions such as the transversality condition rule out explosive trajectories of the model's state (or co-state) variables because they violate the boundary conditions that discipline agents' forward-looking forecasts. Measured as functions, these explosive trajectories have large seminorms, whereas trajectories that satisfy the long-run boundary conditions have small seminorms. Since inductive bias favors solutions with smaller seminorms, ML tends to select those that satisfy long-run conditions, even without being explicitly programmed to search for them. This mechanism helps explain the accuracy of ML methods, including deep learning, in solving high-dimensional models, even those with multiple steady states and hysteresis.

Our second contribution is to argue that inductive bias can serve as a micro-foundation for modeling forward-looking agents. Instead of relying on ad hoc learning rules, we propose equipping agents with a general ML model, such as a richly parameterized DNN, and allowing them to learn from a few observations. Inductive bias ensures that agents will learn a solution that (i) is easy to compute, (ii) exhibits minimal errors, and (iii) satisfies the necessary long-

[1] In sequential and deterministic models, the solutions are the functions themselves, whereas in stochastic or recursive models, the solutions are the policies. In both cases, larger seminorms correspond to more divergent trajectories. Alternative perspectives on simplicity and inductive bias –consistent with, yet more general than, the minimum-norm interpretation– include PAC-Bayes (see Wilson, 2025, for details).

run constraints.

In this way, we may enable economists to construct approximations of forward-looking agents while limiting “additional ‘free parameters,’ unrestricted by theory,” which was a central goal of the rational expectations revolution (Sargent, 2024). Thus, our work builds on the tradition of Sargent (1993, p.16) and Evans and Honkapohja (2001, p.69-70), who connected perfect-foresight models, transversality conditions, stability, and bounded rationality. Our exploration of how ML can construct self-consistent expectations also echoes the ideas found in Bianchi et al. (2022).

After formally introducing inductive bias, we apply it to understanding transversality in three canonical models that constitute the foundation of much of modern macroeconomics: the linear asset pricing model, the neoclassical growth model, and the New Keynesian model. Building on the established knowledge of these models, we demonstrate how ML methods regularize solutions to achieve min-norms that satisfy boundary conditions without directly calculating long-run behavior. Moreover, the solutions exhibit excellent short-run accuracy and can select the right path in the case of multiple steady states.

In the case of the neoclassical growth model, we also highlight the connection between our results and the classical turnpike theorems (see McKenzie, 1976 and Marimón, 1989). These theorems establish that models with long –but finite– horizons share almost identical short- to medium-run dynamics with infinite-horizon models. ML methods seem to “understand” turnpike theorems.

Indeed, all three applications transparently underscore that long-run boundary conditions –such as transversality and no-bubble conditions– arise from economic assumptions essential for model consistency.[2] These are not merely technical conditions; they are intrinsic to the economics of rational expectations. While such boundary conditions that discipline forward expectations may not always be explicitly stated, they are implicitly present when solving infinite-horizon control problems or using recursive methods.

In linear models, boundary conditions are often articulated in terms of stability. For example, in Blanchard and Kahn (1980) and Klein (2000), these conditions are satisfied by selecting the unique non-explosive solution through spectral methods. Checking the eigenvalues for a linear, time-invariant policy provides a sufficient condition to eliminate unstable trajectories that violate transversality. In global methods, boundary conditions are frequently applied implicitly (e.g., during steady-state calculations) or may appear to be bypassed, such as in collocation on a compact space. However, they remain a necessary condition for optimality (see Ekeland and Scheinkman, 1986 and Kamihigashi, 2005), and paying attention to them is a key step in the successful design of a collocation.

In low dimensions, where we have a strong prior on the relevant regions of the state space, economists can artfully tinker to ensure that a compact hypercube is placed at the appropriate location and does not contain the solutions violating transversality. Moreover, by plotting the dynamics of the model, we can see when simulations diverge (see Fernández-Villaverde et al., 2016, p.10).

However, this process is not feasible in high dimensions, as we cannot constrain ourselves to a compact hypercube and may not have a good prior on the location of a steady state. Even evaluating whether transversality conditions are fulfilled for a given policy is computationally

[2] For example, Sargent and Wallace (1973) introduce an asymptotic boundary condition that “the money supply [is] not expected to increase too swiftly.” Similarly, Blanchard and Kahn (1980) emphasize the need to “rule out exponential growth of the expectation.” Knife-edge stability is also a common feature of monetary models that assume perfect foresight or rational expectations, as discussed in Obstfeld and Rogoff (1983).

infeasible, because it requires iterating the policy function for all initial conditions.

Notice here the connection to the issue of stability in *all* numerical methods, even in small models. Simple forward iterations can accumulate numerical errors and be numerically unstable when the solution is only "approximately" stable.[3]

Thus, our paper lays the theoretical groundwork for using deep learning to find equilibria in dynamic models. Examples of this burgeoning literature include Ebrahimi Kahou et al. (2021), Maliar et al. (2021), Azinovic et al. (2022), Han et al. (2022), Kase et al. (2022), Barnett et al. (2023), Fernández-Villaverde et al. (2023), Jungerman (2023), Duarte et al. (2024), and Payne et al. (2024). Interestingly, these studies do not directly impose transversality conditions, and none explicitly address the issue. Alternatively, Ebrahimi Kahou et al. (2024) focus on deterministic optimal control problems in continuous time and ridgeless kernel methods. This setup facilitates some proofs. However, that paper does not address stochastic shocks or recursive environments, and it does not focus on deep learning.

The remainder of the paper is organized as follows. Section 2 introduces the concept of inductive bias and its role in ML. Section 3 presents the linear asset pricing model. Section 4 presents the neoclassical growth model. Section 5 presents the New Keynesian model, and Section 6 concludes. A Supplementary Material document provides additional results.

2 Inductive Bias in ML

Let us start by examining the role of inductive bias in solving functional equations in economics. Let $\mathcal{X}$ be a space, and write an economic model as a set of functional equations $\ell(x, f) = 0$ for all $x \in \mathcal{X}$, where $f : \mathcal{X} \to \mathbb{R}$. For example, in a growth model, $\ell(x, f)$ might

[3] This is part of the appeal of perturbative solutions, which are provably stable even in high dimensions if properly pruned, as shown by Andreasen et al., 2018.

combine the Euler equation residual at a given capital level x and the resource constraint, with f representing the investment policy function.

The ERM solution. A typical solution method approximates f with $f_\theta \in \mathcal{H}(\Theta)$, where $\mathcal{H}$ is a class of function approximations such as a DNN. More concretely, we select θ to minimize the empirical risk (ERM):

$$\min_{\theta \in \Theta} \left\{ \frac{1}{N} \sum_{x \in \mathcal{D}} \|\ell(x, f_\theta)\|_2^2 \right\}, \tag{1}$$

where N points $\mathcal{D} \subset \mathcal{X}$ are taken from either a grid or a sample.[4] We solve for θ using a gradient-based method, which does not scale exponentially with $\dim(\theta)$, unlike solving the underdetermined nonlinear system $\ell(x, f_\theta) = 0$ for all $x \in \mathcal{D}$.

More importantly, if $\mathcal{H}(\Theta)$ is flexible and highly overparameterized ($|\theta| \gg N$), gradient-based optimization methods reliably find an f_θ that interpolates, i.e., $\ell(x, f_\theta) \approx 0$ for all $x \in \mathcal{D}$. However, given the overparameterization, there are many different θ such that $\ell(x, f_\theta) \approx 0$. To which of these solutions do we converge in practice?

The min-norm solution. ML approximations such as DNNs converge in practice toward the "simplest" interpolating solutions. This tendency is called the inductive bias, and its classic interpretation is that ML follows Ockham's razor: the simplest solution is the most likely. But what does "simple" mean in this context?

One framework to define "simplest" is as the solution with a min-norm (Belkin, 2021). When we minimize the ERM with a highly overparameterized approximation, we are effec-

[4] For more complicated problems, such as in the nonlinear New Keynesian model of Section 5, $\mathcal{D}$ may need to be adaptively resampled.

tively solving:

$$f_\theta^* \equiv \min_{f_\theta \in \mathcal{H}(\Theta)} \|f_\theta\|_\psi \tag{2}$$

$$\text{s.t. } \ell(x, f_\theta) = 0, \quad \text{for all } x \in \mathcal{D}, \tag{3}$$

where ψ is some functional seminorm. In other words, we can interpret solutions to the problem (1) as finding the simplest function, measured by $\|\cdot\|_\psi$, that interpolates the data.[5]

While we cannot characterize ψ without imposing further structure on $\mathcal{H}(\Theta)$, $\ell(\cdot,\cdot)$, and the optimization method used to solve problem (1), equations (2) and (3) will guide the reading of our results throughout this paper.

The double descent phenomenon. We are dealing with highly parameterized aproximations $\ell(x, f_\theta)$ such as DNNs. Classic statistics intuition suggests that having excess parameters makes it more likely to overfit, leading to large $||\ell(x, f_\theta)||_2^2$ outside of $\mathcal{D}$ and excess sensitivity to the details of $\mathcal{H}$ and to initial conditions for the optimization algorithm.

Surprisingly, this is not the case. In numerous applications, the double descent phenomenon has shown that, if Θ is large enough, the approximation f_θ not only interpolates in $\mathcal{D}$ regardless of the initial conditions, but it also extrapolates better than low-dimensional approximations. Furthermore, the exact features of $\mathcal{H}(\Theta)$ become less important, and different approximation architectures yield similar results.[6]

The importance of the double descent phenomenon for our argument is that, even if the

[5] An exact min-norm of the inductive bias is provable in some cases, such as in overparameterized linear regression and ridgeless kernel regression (Hastie et al., 2022) and kernel methods (Ebrahimi Kahou et al., 2024), and holds for limiting cases of DNNs (Belkin, 2021; Ma and Ying, 2021). Although the mapping to a seminorm is often only approximate, it nonetheless provides a useful framework for understanding inductive bias.

[6] See Smith et al. (2021), Chiang et al. (2022), and Zhang et al. (2021). Spiess et al. (2023) provide recent examples of double descent in causal inference econometrics.

approximation is highly parameterized, the actual solution is "simple," where simplicity can be interpreted through the seminorm ψ. In other words, counting the dimensionality of θ is a misleading indicator of the parsimony of an approximating function f_θ when we measure parsimony in a function space (the space in which dynamic model solutions reside).

This observation points to a broader lesson. Parameter counting, in general, is a poor proxy for model complexity. We are not interested in the parameters in isolation, but rather in how they shape the properties of the functions we use to fit the data (Wilson, 2025, Section 5). Instead of counting parameters, complexity must be assessed within the function space of the solutions themselves. Overparameterization then ensures that f_θ is "simple" in the right sense.

Long-run expectations. Why do we care about the "simple" solutions f_θ toward which ML algorithms are biased? Because, as we will show in our applications, these "simple" solutions satisfy the long-run boundary conditions of the economic model, such as transversality conditions, without the need to impose them explicitly. In particular, the "simple" solutions fluctuate the least in a well-defined sense and avoid explosive or implosive paths.

Skipping the explicit imposition of long-run boundary conditions materially expands the class of models we can solve in practice, even when $\mathcal{D}$ contains no points near a steady state. For example, a key difficulty in many projection methods is ensuring that these boundary conditions are satisfied, either explicitly or implicitly through the choice of the grid.

Furthermore, the "simple" solution is also the most plausible from a behavioral economics perspective: agents converge to policy rules that satisfy Ockham's razor criterion while remaining consistent with their constraints.

Caveat. The "simple" solution we have discussed provides very accurate approximations for short- and medium-run outcomes. However, it may be slightly inaccurate for learning long-run objects such as the steady state or the ergodic distribution of variables of interest. Nonetheless, this inaccuracy is preferable to alternative methods that might produce explosive long-run paths. The Supplementary Material explains how long-run inaccuracies can be controlled in practice through resampling, extending the time horizon, and carefully tuning the DNN hyperparameters.

Applications. To illustrate how our arguments work, the remainder of this paper examines three classic applications with well-established baselines: (i) a linear asset pricing model, which formally demonstrates the sufficiency of this condition and clarifies its connection to functional seminorms; (ii) the neoclassical growth model, the canonical forward-looking model with saddle-path dynamics, which further elucidates links to turnpike theory; and (iii) a canonical mid-size New Keynesian model, showing how these results hold in models used in daily research. For simplicity and clarity, in applications (i) and (ii), we present the results using the sequence space representation, but the Supplementary Material shows that the same principles apply to recursive representations.

3 Linear Asset Pricing

We begin with the basic model of risk-neutral asset pricing, which has long served a pedagogical role in exploring long-run boundary conditions (e.g., Ljungqvist and Sargent, 2018) and as a foundation for analyzing fiat currency, hyperinflation, and other pure bubbles. Linearity, in turn, helps illustrate the link between multiplicity, asymptotic boundary behavior, and norms in function spaces.

The model. The risk-neutral price $p(t)$ of a claim to an exogenous dividend stream $y(t)$ equals the current dividend plus the expected discounted price:

$$p(t) = y(t) + \beta p(t+1), \tag{4}$$

where $\beta \in (0,1)$ is the discount factor and $t = \{0, 1, \cdots, \infty\}$.[7] Although the model is in discrete time, we write $p : \mathbb{R} \to \mathbb{R}$, taking care to evaluate only at discrete t.

Forward-looking behavior and multiplicity. The recursive structure of equation (4) admits a family of solutions of the form:

$$p(t) = p_f(t) + \zeta \beta^{-t}, \tag{5}$$

where $\zeta \geq 0$ and $p_f(t) \equiv \sum_{\tau=0}^{\infty} \beta^{\tau} y(t+\tau)$ is the fundamental price (i.e., the present discounted value of dividends). The bubble component, $p(t) - p_f(t) = \zeta \beta^{-t}$, is explosive for all $\zeta > 0$ since $\beta < 1$. For the risk-neutral agent, this multiplicity reflects that many internally consistent, fully rational price forecasts satisfy the model's recursive structure, each associated with a different bubble size at $t = 0$, where $p(0) - p_f(0) = \zeta$.

Asymptotic boundary conditions. An interpretation of the multiplicity of solutions to equation (4) is that agents may rationally forecast many paths with bubbles that grow asymptotically, but only one path avoids long-run divergence relative to discounting. This yields the no-bubble condition:

$$0 = \lim_{t \to \infty} \beta^t p(t), \tag{6}$$

that ensures stability. Agents who impose condition (6) reject asset price forecasts that grow faster than $1/\beta$, which constrains their long-run expectations. With this condition, the

[7] To ensure present discounted values are well defined, dividends must not grow faster than β^{-1} (i.e., $\lim_{t\to\infty} |y(t+1)/y(t)| < \beta^{-1}$).

system (4) and (6) becomes well-posed and has the unique solution $p(t) = p_f(t)$.

This simple model illustrates a central computational challenge. With forward-looking agents, short-term forecasts (e.g., $p(t)$ for $t \ll t_N \equiv \max \mathcal{D}$) require imposing the asymptotic condition (6). Otherwise, any $p(0) \geq p_f(0)$ is a valid equilibrium. Thus, even if we care little about long-run dynamics, we must still consider the full sequence $\{p(t)\}_{t=0}^{\infty}$, subject to the no-bubble condition (6), to ensure internal consistency in the short run.

Stability and function norms. The no-bubble condition is a special case of a broader class of transversality conditions, which often arise as stability requirements in optimal control. A policy is stable if it does not diverge, and transversality conditions ensure that repeated applications of a policy rule satisfy this property.

Our goal is to show how these economically meaningful long-run boundary conditions can be satisfied without explicitly solving for them. Intuitively, our approach hinges on the fact that the solution satisfying the transversality condition is the stable one: among all possible solutions, ML methods select the least explosive.

To formalize this idea, consider a function seminorm, denoted $||p||_{\psi}$. An important example is the Sobolev seminorm defined on $t \in [0, T]$, where $||p||^2_{W^{1,2}} \equiv \int_0^T |p'(t)|^2 \, dt$. Functions with steeper derivatives, $p'(t)$, will have larger norms, and explosive functions exhibit diverging norms relative to flatter ones, since $\lim_{T\to\infty} |p'(T)| = \infty$.[8]

In our model, explosive solutions exhibit larger norms than those based solely on fundamentals. Consider the general solution in equation (5), recall that $\zeta \geq 0$, and apply the triangle inequality to compare its norm to that of $p_f(\cdot)$, the unique solution consistent with

[8] Restricting attention to a closed interval with finite T ensures the norm is well defined and is innocuous here. This restriction can be relaxed by modifying the norm definition, as in Van et al. (2007), who apply exponential discounting to the function space.

the no-bubble condition:

$$\|p\|_\psi \equiv \|p_f + \zeta\beta^{-t}\|_\psi \le \|p_f\|_\psi + \zeta\,\|\beta^{-t}\|_\psi. \tag{7}$$

The norm is minimized when $\zeta = 0$, and in that case $\|p_f\|_\psi = \|p\|_\psi$. To see that these are the same function, up to an equivalence class, compare the solutions: $\|p - p_f\|_\psi \equiv \|p_f + \zeta\beta^{-t} - p_f\|_\psi = \zeta\|\beta^{-t}\|_\psi$. Hence, $\zeta = 0$ implies $\|p - p_f\|_\psi = 0$.[9]

Since we only used the triangle inequality in the proof, these results hold for any seminorm ψ. However, we focus on the Sobolev seminorm because Ma and Ying (2021) establish that, under certain conditions, DNNs and first-order optimization methods (the approach that we follow in this paper) are biased toward functions with small Sobolev seminorms.

The coincidence of solutions. The previous argument showed that the unique solution to the model with the asymptotic boundary condition is the smallest (and hence flattest, given relevant norms that penalize gradients) solution to equation (4). This is also the solution ML methods prefer due to their inductive bias toward flatter functions. In a fortunate coincidence, "small" solution functions are both the preferred choice of ML methods and the outcome consistent with economic assumptions on agents' long-run expectations.

The numerical solution. Next, we examine how this observation holds in practice. First, define a highly parameterized DNN approximation $p_\theta \in \mathcal{H}(\Theta)$. For $\mathcal{X} = [0, \infty)$, choose $\mathcal{D} \equiv \{t_1, \cdots, t_N\} \subset \mathcal{X}$ and minimize (4) numerically:

$$\min_{\theta\in\Theta} \frac{1}{N} \sum_{t\in\mathcal{D}} \left[p_\theta(t) - y(t) - \beta p_\theta(t+1)\right]^2, \tag{8}$$

[9] If we have a seminorm, rather than a norm, then it only provides an equivalence class, $\|p_f - p\|_\psi = 0$ rather than ensuring p and p_f are identical pointwise. While this is not an issue in this particular example, it can be relevant when considering two functions in an equivalence class that fulfill the model equations, and long-run boundary conditions could have economically relevant differences. In that case, one needs to adapt the approach accordingly.

where the baseline example generates dividends as $y(t+1) = c + (1+g)y(t)$, with $y(0) = y_0$, $c > 0$, and $g \geq -1$.

Since this is a particular case of problems (1) and (3), we can interpret problem (8) as:

$$\min_{p_\theta \in \mathcal{H}(\Theta)} \|p_\theta\|_\psi$$

$$\text{s.t. } p_\theta(t) = y(t) + \beta p_\theta(t+1), \quad \text{for all } t \in \mathcal{D},$$

where we know from equation (7) that this result holds for any seminorm.[10]

Calibration. We set the parameter values of the model at $\beta = 0.9$, $c = 0.01$, and $y_0 = 0.08$, and we consider $g = -0.1$ and $g = 0.02$. The first g leads to prices converging to a deterministic steady state (DSS), while the second leads to prices growing indefinitely. However, since $g < \beta^{-1} - 1$, there is a well-defined price on a balanced growth path (BGP). We do not provide g to the algorithm or calculate a BGP, but allow our DNN to scale exponentially: when $g > 0$, we set $p_\theta(t) = \exp(\phi t)\mathrm{NN}(t;\theta_{\mathrm{NN}})$, where $\theta \equiv \{\phi, \theta_{\mathrm{NN}}\}$, $\phi \in \mathbb{R}$, and $\mathrm{NN}(t;\cdot)$ is a DNN to be defined.

Results. We fit problem (8) using DNNs, where $p_\theta(t) = \mathrm{NN}(t;\theta)$ or $p_\theta(t) = \exp(\phi t)\mathrm{NN}(t;\theta_{\mathrm{NN}})$. DNNs illustrate how our arguments work with a maximally flexible functional form. For instance, DNNs encompass many functional forms familiar to economists, such as splines and orthogonal polynomials.

Our baseline solution uses a DNN with four hidden layers of 128 nodes each, *Tanh* as the activation function, and a final layer of *Sofplus*. In other words, we use 30 grid points with 50K parameters, overparameterizing by about four orders of magnitude.[11]

[10] See Blanc et al. (2020), Damian et al. (2021), and Ma and Ying (2021) for more on characterizing the approximate function norms of the inductive bias. In some limiting cases, it can be proven to be $W^{1,2}$.

[11] The results are not especially sensitive to the design of the DNN as long as it is sufficiently overparameterized. We use the L-BFGS optimizer for its robustness and speed, and compute the gradients of the

For the grid, we use $\mathcal{D} = \{0, 1, 2, \ldots, 29\}$. While our focus is on the short term, we also plot an extrapolation region for $t > 30$ to assess how well the DSS is forecasted and to gauge stability. Since when $g = 0.02$, there is a BGP rather than a DSS, we check whether the DNN can learn g, i.e., whether $\log(1 + g) \approx \phi$ given the $p_\theta(t) = \exp(\phi t)\mathrm{NN}(t; \theta_{\mathrm{NN}})$ approximation.

To check robustness, we rerun the optimizer from different initial conditions for θ and report the median, 10th, and 90th percentiles. The primary metric is the relative error, $\varepsilon_p(t) \equiv (p_\theta(t) - p_f(t))/p_f(t)$.

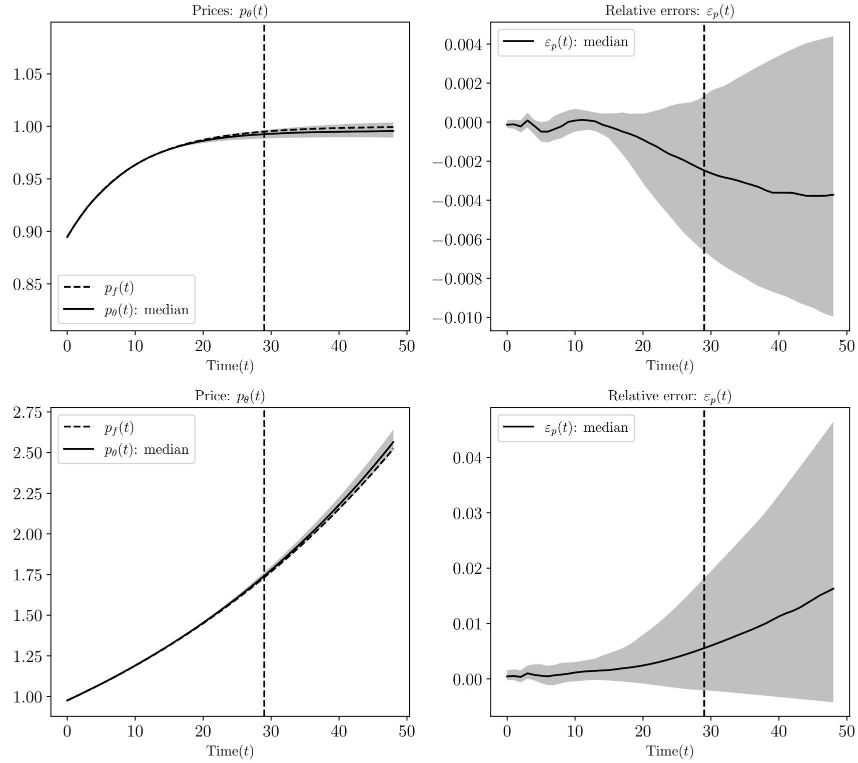

Figure 1: Solutions to problem (8) for an ensemble of 100 initial conditions for θ. Top row for $g = -0.1$, and bottom row for $g = 0.02$.

Figure 1 plots our results. The dotted line represents the closed-form baseline, the solid line depicts the median, and the shaded region illustrates the 10th to 90th percentiles. The top left panel compares the solutions for $g = -0.1$, while the top right panel shows the

objective function (8) using all of $\mathcal{D}$.

corresponding relative errors. The bottom left panel contrasts the solutions for $g = 0.02$, and the bottom right panel plots the relative errors.

The inductive bias toward the min-norm solution delivers highly accurate approximations in the short run. The approximations also perform well in the extrapolation region for $t > 30$, even though this is not our primary objective. The difference between $p_\theta(t)$ and $p_f(t)$ is imperceptible in the short run, as shown in the two left panels.

For the stationary case with $g = -0.1$, the relative errors up to $t = 10$ are within numerical precision on average and remain well below 0.1% even at the 90th percentile. In the non-stationary case, up to $t = 10$, the solutions are close to numerical precision, and the 90th percentile of errors stays around 0.5%, with the 90th percentile reaching nearly 2% at $t = 30$, as shown in the bottom right panel. Even an imperfect characterization of the long run does not generate substantial short-run errors, even without knowledge of g.

Recall that, by leveraging our economic understanding of the problem's structure, we allowed the DNN to rescale itself using $p_\theta(t) = \exp(\phi t)\mathrm{NN}(t; \theta_{\mathrm{NN}})$, where $\phi \in \mathbb{R}$ is learned. In this context, ML selects the ϕ that minimizes the norm of the $\mathrm{NN}(t; \theta)$ function, creating a strong bias toward the closed-form solution $\phi = \log(1 + g)$, as this choice minimizes the explosiveness of $\mathrm{NN}(t; \theta_{\mathrm{NN}})$. Other solutions with different ϕ values require more explosive $\mathrm{NN}(t; \theta_{\mathrm{NN}})$ to compensate, resulting in higher norms. This highlights that successful outcomes depend on integrating economic insights into the design of $\mathcal{H}$ (as, in any case, one should do with *any* numerical method).

Behavioral interpretation. We can view our solution method as a behavioral approximation of agents with an inductive bias. In this interpretation, agents train their DNNs with only 30 observations and regard asset pricing trajectories that lead to more stable solutions as the most likely. While the solution does not exactly select a stable path (as shown by the slight divergences in the long run in the top panels of Figure 1) it matches the short run to numerical precision, i.e., a behavioral agent prices assets well.

Even more surprisingly, the policies are approximately time-consistent. If we re-optimized at $t = 5$, for example, the agent would similarly choose a path with low error in the short run that is (slightly) unstable in the long run.

4 The Neoclassical Growth Model

The neoclassical growth model is a classic example of the importance of transversality conditions in ruling out suboptimal paths. As in our previous example, we analyze these conditions and their connection to inductive bias. To push the argument further, we also examine a version of the model with multiple DSSs. We will show how, despite the nonlinear nature of the growth model and the requirement to satisfy both initial values and boundary conditions, the inductive bias delivers the required stable solutions.

Model. We follow the standard treatment of the neoclassical growth model in Ljungqvist and Sargent (2018) with a log utility. Then, we can jump straight to the dynamic system of equations that characterizes the optimal path:

$$k(t+1) = z(t)^{1-\alpha} f\left(k(t)\right) + (1-\delta)k(t) - c(t), \tag{9}$$

$$c(t+1) = \beta c(t)\left[z(t+1)^{1-\alpha} f'\big(k(t+1)\big) + 1 - \delta\right], \tag{10}$$

$$\lim_{t\to\infty} \beta^t c(t)^{-1} k(t+1) = 0, \tag{11}$$

given $\beta \in (0,1), \delta \in (0,1)$, and an initial condition $k(0) = k_0$. Equation (9) is the law of motion derived from the resource constraint, equation (10) is a forward-looking Euler equation, and equation (11) is a forward-looking transversality condition that prevents capital from accumulating too fast relative to the marginal utility of consumption, $c(t)^{-1}$.

Technology, $z(t)$, follows $z(t+1) = (1+g)z(t)$ given a growth rate $0 \leq g < 1/\beta - 1$ and initial condition $z(0) = z_0$. Our baseline production function is $f(k) = k^{\alpha}$ for $\alpha \in (0,1)$, which implies a unique DSS (or BGP) and transition dynamics toward it.

Forward-looking behavior and saddle-path stability. Given only k_0 and equations (9) and (10), the model admits multiple DSSs with associated transition dynamics unless the transversality condition is imposed. In the simple case $z(t) = 1$ for all t, there are two sets of possible paths: one contains all suboptimal capital paths $k_{\max}(t; c_0)$ and their corresponding consumption paths, characterized by suboptimal initial consumption choices c_0. In this set, the limit approaches the global maximum of output where $f'(k^*_{\max}) = \delta k^*_{\max}$ and $c^*_{\max} = 0$. The other set consists of $k(t)$ and $c(t)$ converging to an interior k^* and, c^*, where the economy follows the saddle path with positive consumption.[12]

Stability and function norms. The transversality condition eliminates the first set of paths above, since it rules out trajectories with maximal output and no consumption in the limit.

But this path also has a higher seminorm than the saddle-path solution because $k^*_{\max} \gg k^*$. Since the output-maximizing trajectories $k_{\max}(t)$ and the saddle-path optimal trajectory $k(t)$ both grow from the same initial condition $k(0) = k_0$, the trajectories leading to $k^*_{\max}$

[12] There is a trivial third steady state (ignored here) with $k = 0$ that is not an attracting basin, cannot be reached unless $k_0 = 0$, and is unstable to any perturbation of k_0. The case with $g > 0$ in the TFP process is similar but requires rescaling by the geometric growth rate.

are steeper, with $\|k_{\max}\|_\psi > \|k\|_\psi$ for norms that penalize gradients, such as $\psi = W^{1,2}$.[13]

The min-norm solution. To illustrate the previous argument, we solve the model without applying the long-run boundary condition and rely on the inductive bias of the ML algorithms to select the min-norm solution.

First, define a DNN approximation $k_\theta \in \mathcal{H}(\Theta)$. Next, for $\mathcal{X} = [0, \infty)$ choose $\mathcal{D} \equiv \{t_1, \cdots, t_N\} \subset \mathcal{X}$ and minimize (10) subject to $k(0) = k_0$:

$$\min_{\theta \in \Theta} \left[\frac{1}{N} \sum_{t \in \mathcal{D}} \left(\frac{c(t+1; k_\theta)}{c(t; k_\theta)} - \beta \big[z(t+1)^{1-\alpha} f'\big(k_\theta(t+1)\big) + (1-\delta) \big] \right)^2 + \left(k_\theta(0) - k_0\right)^2 \right], \quad (12)$$

where $z(t) = z(0)(1+g)^t$, and consumption is defined as a function of k_θ through the feasibility constraint $c(t; k_\theta) = z(t)^{1-\alpha} f(k_\theta(t)) + (1-\delta)k_\theta(t) - k_\theta(t+1)$. As before, we will use (2) to interpret solutions to problem (12) as the interpolating solution that minimizes some norm, $||k_\theta||_\psi$.

Results. The baseline parameters are $f(k) \equiv k^\alpha$, $\beta = 0.9$, $\alpha = 0.33$, $\delta = 0.1$, $k_0 = 0.4$, $z(0) = 1$, and $g = 0$. We choose $\mathcal{D} = \{0, 1, 2, \ldots, 29\}$ and minimize equation (12) using $k_\theta(t) = \mathrm{NN}(t; \theta)$ for a DNN with four hidden layers, 128 nodes in the hidden layers, *Tanh* as the activation function, and a final layer of *Softplus*.[14]

We solve the ERM using L-BFGS and all of $\mathcal{D}$. The relative errors are $\varepsilon_k(t) \equiv \frac{k_\theta(t) - k(t)}{k(t)}$ and $\varepsilon_c(t) \equiv \frac{c_\theta(t) - c(t)}{c(t)}$. Our goal is to ensure that inductive bias leads to low errors in the short to medium run, even if there may be a small degree of instability in the long run.

Figure 2 presents our results. The left panel plots the median of the approximate solutions for capital and consumption, $k_\theta(t)$ and $c_\theta(t)$, against a benchmark (dashed lines)

[13] In the Supplemental Material, we see this even more starkly by inspecting the evolution of the co-state variable (i.e., the marginal utility).

[14] In the case with $g > 0$, we scale the DNN exponentially with $k_\theta(t) = \exp(\phi t)\mathrm{NN}(t; \theta_{\mathrm{NN}})$, where $\theta \equiv \{\phi, \theta_{\mathrm{NN}}\}$. We do not provide the approximation with g and let it decide whether to normalize the solution.

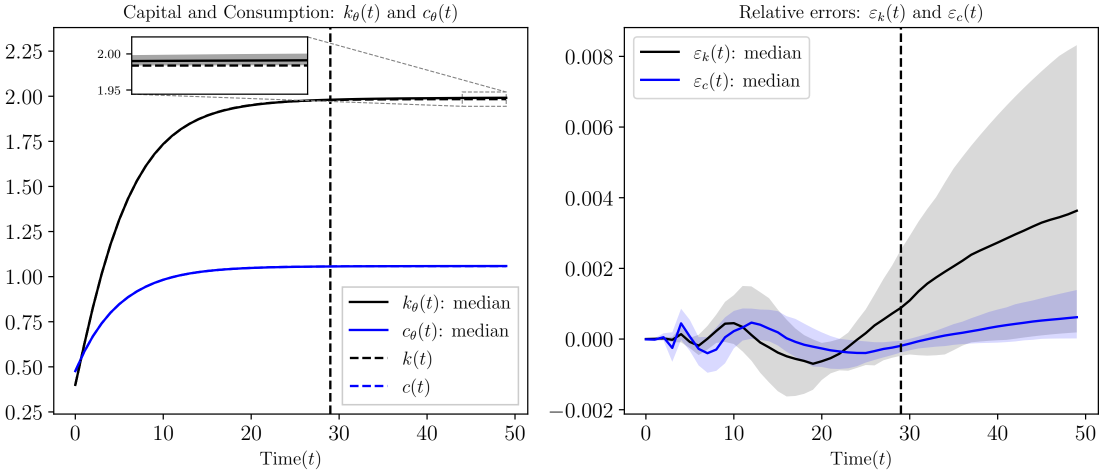

Figure 2: Ensemble of 100 initial conditions for θ solving (12) with $g = 0$ and $k_0 = 0.4$.

obtained with value function iteration.[15] Even without the transversality condition (11), the approximation always yields the correct dynamics. The right panel shows the median errors and the shaded region from the 10th to the 90th percentiles for the errors relative to the baseline, $\varepsilon_k(t)$ and $\varepsilon_c(t)$.

The errors are close to numerical precision in the short to medium run. In the long-run extrapolation region, where $T > 30$, the relative error grows slowly, with a median error of less than 0.1% for $k_\theta(t)$. This confirms that (i) inductive bias selects the correct trajectories consistent with the transversality condition; (ii) small long-run extrapolation errors do not propagate into large short-run errors; and (iii) the solution is almost stable.

In the long run ($t > 29$), the solution slightly deviates from the steady state. This issue can be resolved by extending the time horizon and re-solving the optimization problem starting from $t = 30$.

[15] We use a capital grid with 50 points over the interval $[0.9k_0, 1.1k^*]$, where k^* denotes the steady-state capital and k_0 is the initial level of capital. A tolerance of 10^{-9} is used for the convergence criterion of the value function.

Results with a BGP. Next, we solve the same model with $g = 0.02$ and show that the inductive bias still leads to the correct solution. As before, when we solve a version with a BGP, we neither manually detrend nor provide g to the approximation.

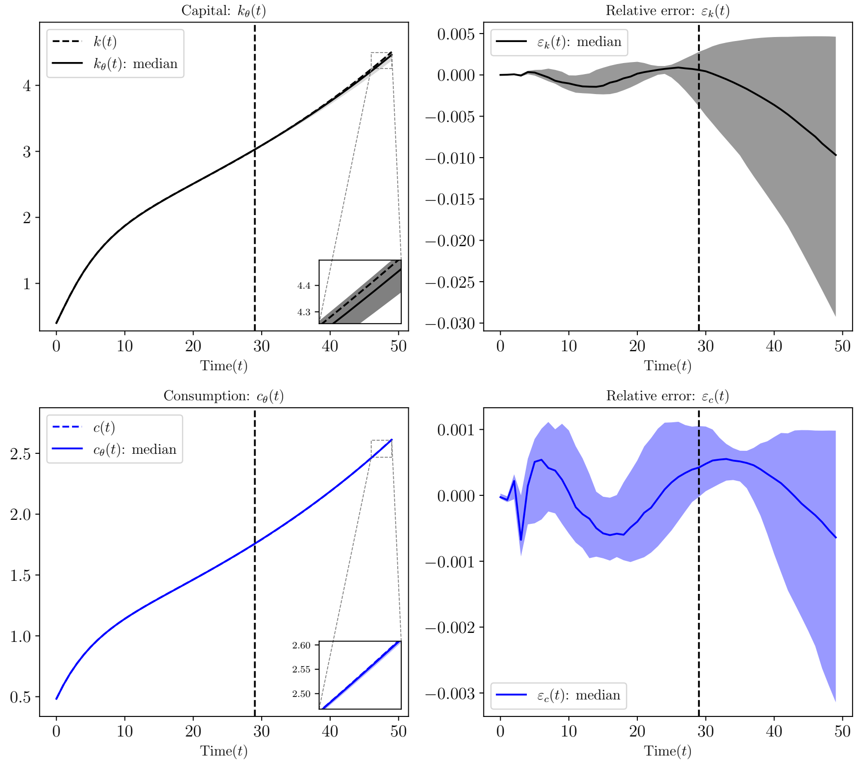

Figure 3: Ensemble of 100 initial conditions for θ solving (12) with $g = 0.02$ and $k_0 = 0.4$.

Figure 3 plots the results. The left panels show the median of the approximate capital and consumption paths, $k_\theta(t)$ and $c_\theta(t)$, with the baseline solutions as dashed lines. The right panels show the median and the shaded region from the 10th to the 90th percentiles for the errors relative to the baseline, $\varepsilon_k(t)$ and $\varepsilon_c(t)$.

The short- to medium-run errors are extremely small, and even in the extrapolation region, the median error is roughly 0.1% for consumption at $t = 50$. The inductive bias leads the approximation to choose a min-norm solution and an appropriate rescaling, despite not being given the growth rate, the transversality condition, or the BGP. The intuition is the same as in the asset pricing example with growth: if $k(t)$ is approximated by $\exp(\phi t)\mathrm{NN}(t;\theta_{\mathrm{NN}})$, the ϕ that yields the smallest norm for $\mathrm{NN}(t;\theta_{\mathrm{NN}})$ is $\phi \approx \log(1+g)$. All other ϕ produce explosive $\mathrm{NN}(t;\theta_{\mathrm{NN}})$ functions as $t \to \infty$. See the Supplementary Material for more details.

State-space formulation. The Supplementary Material shows that the transversality condition (11) is also required in a recursive state-space formulation of the problem. In that case, the inductive bias toward min-norm solutions applies to the policy function, selecting policy functions with smaller gradients where k_t does not explode.

Robustness. The Supplementary Material reports a wide set of robustness tests. Most notably, it shows that we obtain nearly as accurate short- to medium-run forecasts with a sparser and irregular grid –even with as few as nine points– using the same number of parameters for the DNN. We also obtain excellent solutions when fitting a misspecified functional form of a BGP version.

Turnpikes. An interpretation of inductive bias is that agents turn toward solutions on the turnpike because the turnpike trajectory is the unique path that does not diverge, and it is the "easiest" to characterize.

Turnpike theorems (see McKenzie, 1976, and Marimón, 1989) show that models with long (but finite) horizons have almost the same short- to medium-run dynamics as infinite-horizon models. Even if trajectories have local transition dynamics at $t = 0$ and the terminal point T, for a large enough time frame, it is optimal to remain close to the time-invariant steady state except close to 0 and T.

Multiple DSSs and hysteresis. How would inductive bias move the system toward the correct DSS for a given initial condition when multiple DSSs exist?

To investigate this question, we solve the same model as before but replace the production function with $f(k) \equiv \max\{k^\alpha,\, b_1 k^\alpha - b_2\}$ for $b_1 > 1$ and $b_2 > 0$, following Skiba (1978). In this case, there are two sets of steady states, (k_1^*, c_1^*) and (k_2^*, c_2^*), each with its domain of attraction. As before, without imposing transversality, there exists a $(k_{\max}^*, 0)$ solution with vanishing consumption, but inductive bias rules it out.

Let $a = 0.5$, $g = 0$, $b_1 = 3$, and $b_2 = 2.5$. This calibration yields steady states $k_1^* = 2.75$ and $k_2^* = 4$. The model is solved as before by choosing $\mathcal{D}$ and minimizing function (12), with the only change being the new $f(\cdot)$ and $f'(\cdot)$. In particular, we do not give the algorithm any indication that multiple DSSs exist.[16]

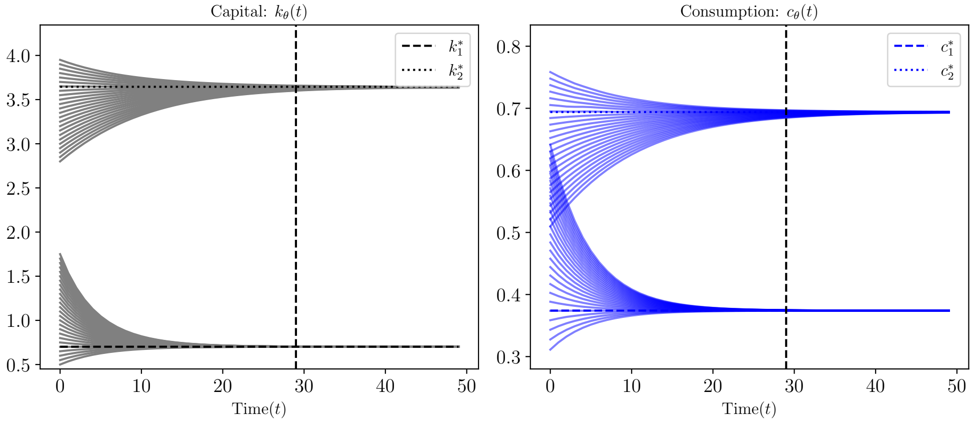

Figure 4: Solutions to (9) and (10) with a convex-concave production function.

Figure 4 shows the results for various initial conditions crossing between the two domains of attraction. The left panel plots the capital paths for different initial capital levels, and the right panel shows the corresponding consumption paths. The inductive bias selects solutions

[16] We now solve the ERM problem with the Adam optimizer, which is slower than L-BFGS, but introduces a stronger inductive bias.

that converge to the correct DSS, even when multiplicity is present. ML accurately generates transition dynamics from each initial condition toward the appropriate domain of attraction.

ML selects the correct DSS for two reasons. First, the discontinuity in the marginal product of capital would create a discontinuity in the Euler equation (10) if trajectories crossed between the two regions. These discontinuities lead to large changes in $k(t)$, which the inductive bias avoids. Second, any trajectory moving toward the wrong steady state would require steeper transitions, as it must first approach the correct domain of attraction. The success of this experiment suggests that it is possible to solve complex economic models with significant hysteresis and multiple DSSs.

5 A Nonlinear New Keynesian Model

Our final application illustrates how inductive bias operates in a more complex setting: a canonical nonlinear New Keynesian model based on Fernández-Villaverde and Rubio-Ramírez (2006) and Fernández-Villaverde and Guerrón-Quintana (2021). To keep the exposition relatively brief and the results more transparent, we shut down some of the shocks included in this class of models. None of these omissions affects our point: the key mechanism relevant for our argument –the transversality conditions of the representative household and the presence of physical capital– remains present.

The representative household. The economy is populated by a representative household that has a lifetime utility function over consumption, c_t, and hours worked, l_t:

$$\mathbb{E}_0 \sum_{t=0}^{\infty} \beta^t \left[\log(c_t) - \psi \frac{l_t^{1+\eta}}{1+\eta} \right],$$

subject to the budget constraint $c_t + k_t + \frac{b_{t+1}}{p_t} = w_t l_t + (1 + r_t - \delta)k_{t-1} + R_{t-1}\frac{b_t}{p_t} + T_t + F_t$ and the standard transversality conditions $\lim_{t\to\infty} \beta^t c_t^{-1} k_t = \lim_{t\to\infty} \beta^t c_t^{-1} \frac{b_{t+1}}{p_t} = 0$.

Beyond consumption, the household saves in physical capital, k_t, and buys government debt, b_{t+1}.[17] The household earns income by supplying labor at the real wage w_t and renting capital at the real rental price r_t (δ is the depreciation rate). Additional resources include government debt with a nominal gross interest rate R_t, lump-sum transfers from the government T_t, and profits from the firms F_t.

The final good producer. A perfectly competitive final good producer combines intermediate goods with the technology $y_t = \left(\int_0^1 y_{it}^{\frac{\varepsilon-1}{\varepsilon}} \, \mathrm{d}i\right)^{\frac{\varepsilon}{\varepsilon-1}}$, where ε is the elasticity of substitution.

Given the intermediate goods prices p_{it} and the final good price p_t, the demand function for each intermediate good i is $y_{it} = \left(\frac{p_{it}}{p_t}\right)^{-\varepsilon} y_t$, where y_t is the aggregate demand in the economy (to be defined below). Integrating over i and using the zero-profit condition for the final good producer, we obtain $p_t = \left(\int_0^1 p_{it}^{1-\varepsilon} \, \mathrm{d}i\right)^{\frac{1}{1-\varepsilon}}$.

Intermediate good producers. Each intermediate good producer i has a production function $y_{it} = e^{z_t} k_{it-1}^{\alpha} l_{it}^{1-\alpha}$, where k_{it-1} is the capital rented by the firm, l_{it} is the labor input, and z_t is the technology shock that evolves as $z_t = \rho z_{t-1} + \sigma_z \epsilon_{z,t}$, where $\epsilon_{z,t} \sim \mathcal{N}(0, 1)$.

Intermediate good producers solve a two-stage problem. First, taking the input prices w_t and r_t as given, firms rent l_{it} and k_{it-1} to minimize the (marginal) real cost:

$$\mathrm{mc}_t = \left(\frac{1}{1-\alpha}\right)^{1-\alpha} \left(\frac{1}{\alpha}\right)^{\alpha} \frac{w_t^{1-\alpha} r_t^{\alpha}}{e^{z_t}}.$$

The marginal cost is the same for all intermediate good producers, as they all have access to the same technology and take input prices as given.

[17] The household can also trade in Arrow securities, but we omit them from the budget constraint since their equilibrium net supply is zero.

Second, each producer chooses the price that maximizes discounted real profits subject to Calvo pricing. In each period, a fraction $1-\vartheta$ of firms can change their prices. All other firms keep their old prices. The marginal valuation of future profits, $\beta^{\tau}\frac{\lambda_{t+\tau}}{\lambda_t}$, comes from the fact that the household owns the firm and we have complete markets: λ_t is the Lagrangian multiplier of the household's budget constraint. For this problem to be well posed, we assume that $(\beta\vartheta)^{\tau}\lambda_{t+\tau}$ goes to zero sufficiently fast in relation to inflation.

After some algebra, this pricing problem is characterized by two auxiliary variables:

$$\begin{aligned} g_t^{(1)} &= \lambda_t \mathrm{mc}_t y_t + \beta\vartheta\mathbb{E}_t\left[\Pi_{t+1}^{\epsilon} g_{t+1}^{(1)}\right], \\ g_t^{(2)} &= \lambda_t \Pi_t^* y_t + \beta\vartheta\mathbb{E}_t\left[\Pi_{t+1}^{\epsilon-1}\left(\frac{\Pi_t^*}{\Pi_{t+1}^*}\right) g_{t+1}^{(2)}\right], \end{aligned}$$

where $\Pi_t^* \equiv \frac{p_t^*}{p_t}$ is the ratio of the reset price of firms that can change their prices over p_t and the price index evolves as $1 = \vartheta\Pi_t^{\epsilon-1} + (1-\vartheta)(\Pi_t^*)^{1-\epsilon}$.

The government. The government sets the nominal interest rate following a Taylor rule:

$$R_t = R\left(\frac{\Pi_t}{\Pi}\right)^{\gamma_\Pi}\left(\frac{y_t}{y}\right)^{\gamma_y} e^{\sigma_m \epsilon_{m,t}}.$$

The variable R is the steady-state nominal gross return of capital (equal to the steady-state real gross return of capital plus the target level of inflation Π), and y is the steady-state level of output. The term $\epsilon_{m,t} \sim \mathcal{N}(0,1)$ is a random shock to monetary policy.

Aggregation. Aggregate demand is given by $y_t = c_t + k_t - (1-\delta)k_{t-1}$, and aggregate supply is $y_t^s = \frac{1}{v_t} e^{z_t} k_{t-1}^{\alpha} l_t^{1-\alpha}$, where $v_t = \int_0^1 \left(\frac{p_{it}}{p_t}\right)^{-\epsilon} di$ is the loss of output created by price rigidities and the resulting misallocation of inputs. By Calvo pricing, v_t evolves according to $v_t = \vartheta\Pi_t^{\epsilon} v_{t-1} + (1-\vartheta)(\Pi_t^*)^{-\epsilon}$. All equilibrium conditions are listed in the Supplementary Material.

State variables. The state space in this problem is four-dimensional: capital (k), the inefficiency (or loss) caused by price rigidities (v, which we refer to simply as "inefficiency"), technology (z), and the monetary policy shock (ϵ_m).

Since the numerical ranges of the state variables differ significantly, it is useful to normalize them to avoid training issues such as slow convergence, poor generalization, and numerical instability (LeCun et al., 2002). A convenient normalization divides the state variables by their DSS value (or, if this value is zero, by a multiple of their standard deviations): $s \equiv \left[\frac{k-k^*}{k^*}, \quad \frac{v-v^*}{v^*}, \quad \frac{z}{\zeta_z \sigma_z}, \quad \frac{\epsilon_m}{\zeta_m \sigma_m}\right]$. Other normalizations are possible, and the researcher should choose the one most convenient for her goals.

Calibration. Table 1 reports our calibration. All parameter values are standard in the literature, and we skip their discussion for the sake of brevity.

Symbol	**Definition**	**Value**
Π	Inflation target	1.005
β	Discount factor	0.99
δ	Depreciation rate	0.025
ϑ	Calvo parameter	0.8
ϵ	Elasticity of substitution	10.0
α	Capital share	0.33
η	Labor exponent in the utility function	1.0
γ_Π	Taylor rule coefficient on inflation	2.0
γ_y	Taylor rule coefficient on output	0.5
ρ	Persistence of technology shock	0.95
σ_z	Std. dev. of technology shock	0.007
σ_m	Std. dev. of monetary policy shock	0.001
ψ	Labor disutility weight	0.766

Table 1: Calibration

The DNNs. We use highly parameterized DNNs, $\Psi_\theta(\cdot)$, to approximate the policy functions for labor, $l(s)$, capital, $k'(s)$, the Lagrangian multiplier, $\lambda(s)$, gross inflation, $\Pi(s)$, and the expected discounted marginal cost of intermediary firms, $g^1(s)$. Once these functions are approximated, the remaining variables, such as consumption $c(s)$ and wages $w(s)$, can be determined using the equilibrium conditions, as explained in the Supplementary Material.

As was the case with the state variables, the variations in $l(\cdot)$, $k'(\cdot)$, $\lambda(\cdot)$, $\Pi(\cdot)$, and $g^{(1)}(\cdot)$ are substantial. For example, their DSS values are given by $(l^*, k^*, \lambda^*, \Pi, g^{(1)^*}) = (1.0, 24.2, 0.44, 1.005, 6.8)$. Thus, we also normalize them to ensure that the relevant gradients are well-scaled across all output dimensions. It also helps the networks learn more efficiently by preventing large-magnitude outputs from dominating the loss function or slowing down convergence.

More formally:

$$\Psi_\theta(s) = \begin{bmatrix} l_\theta(s) \\ k'_\theta(s) \\ \lambda_\theta(s) \\ \Pi_\theta(s) \\ g^1_\theta(s) \end{bmatrix} = \begin{bmatrix} l^* \cdot (1 + \mathrm{NN}_l(s; \theta_l)) \\ k^* \cdot (1 + \mathrm{NN}_{k'}(s; \theta_{k'})) \\ \lambda^* \cdot (1 + \mathrm{NN}_\lambda(s; \theta_\lambda)) \\ \Pi \cdot (1 + \mathrm{NN}_\Pi(s; \theta_\Pi)) \\ g^{(1)^*} \cdot (1 + \mathrm{NN}_{g^{(1)}}(s; \theta_{g^{(1)}})) \end{bmatrix}$$

where $\theta \equiv \{\theta_l, \theta_{k'}, \theta_\lambda, \theta_\Pi, \theta_{g^{(1)}}\}$ contains 433,000 parameters, and we have the following architectures:

- $\mathrm{NN}_l(s; \theta_l)$: 3 hidden layers, with a Sigmoid Linear Unit (*SiLU*) activation function, and the final layer uses *Tanh*. Each layer has 128 nodes.

- $\mathrm{NN}_{k'}(s; \theta_{k'})$: 4 hidden layers, with *SiLU* activation function, and the final layer uses *Tanh*. Each layer has 200 nodes.

- $\mathrm{NN}_\lambda(s; \theta_\lambda)$: 3 hidden layers, with *SiLU* activation function, and the final layer uses *Tanh*. Each layer has 128 nodes.

- $\text{NN}_{\Pi}(s; \theta_{\Pi})$: 4 hidden layers, with *SiLU* activation function, and the final layer uses *Tanh* (see the Supplementary Material for an explanation of this and related choices in the architecture design). Each layer has 200 nodes.

- $\text{NN}_{g^{(1)}}(s; \theta_{g^{(1)}})$: 4 hidden layers, with *SiLU* activation function, and the final layer uses *Linear*. Each layer has 200 nodes.

The ERM. Using equilibrium equations (11), (12), (13), (14), and (21) in the Supplementary Material, we can define the following residual functions:

$$\mathcal{R}_1(s, \Psi_\theta) \equiv 1 - \frac{\beta}{\lambda_\theta(s)} \mathbb{E}\left[\frac{\lambda_\theta(s') R(s; \Psi_\theta)}{\Pi_\theta(s')} \,\middle|\, s \right],$$

$$\mathcal{R}_2(s, \Psi_\theta) \equiv 1 - \frac{\beta}{\lambda_\theta(s)} \mathbb{E}\left[\lambda_\theta(s') \left(1 + r(s'; \Psi_\theta) - \delta\right) \mid s \right],$$

$$\mathcal{R}_3(s, \Psi_\theta) \equiv 1 - \left(\frac{\lambda_\theta(s) \text{mc}(s; \Psi_\theta) y(s; \Psi_\theta)}{g_\theta^{(1)}(s)} + \frac{\beta \vartheta}{g_\theta^{(1)}(s)} \mathbb{E}\left[\Pi_\theta(s')^{\varepsilon} g_\theta^{(1)}(s') \,\middle|\, s \right] \right),$$

$$\mathcal{R}_4(s, \Psi_\theta) \equiv 1 - \left(\frac{\lambda_\theta(s) \Pi^*(s; \Psi_\theta) y(s; \Psi_\theta)}{g_\theta^{(2)}(s)} + \frac{\beta \vartheta}{g_\theta^{(2)}(s)} \mathbb{E}\left[\Pi_\theta(s')^{\varepsilon - 1} \left(\frac{\Pi^*(s; \Psi_\theta)}{\Pi^*(s'; \Psi_\theta)} \right) g_\theta^{(2)}(s') \,\middle|\, s \right] \right),$$

$$\mathcal{R}_5(s, \Psi_\theta) \equiv 1 - \frac{y(s; \Psi_\theta) + (1 - \delta)k - k'_\theta(s)}{c(s; \Psi_\theta)},$$

where expectations are computed using Gaussian quadrature.

Using these residual functions, we define the theoretical loss function to be minimized as $\ell(s, \Psi_\theta) \equiv \sum_{i=1}^{5} \mathcal{R}_i^2(s, \Psi_\theta)$. For the sake of brevity, we omit the min-norm formulation of the problem discussed in the previous two applications.

Let $\mathcal{D} \equiv \{s_1, \cdots, s_N\}$ denote a training set consisting of N points from the state space (the details of this sampling will be explained below). Thus, the ERM approximation to the theoretical loss function is:

$$\mathcal{L}(\mathcal{D};\Psi_\theta) \equiv \frac{1}{N}\sum_{s_j\in\mathcal{D}} \ell(s_j,\Psi_\theta), \tag{13}$$

with an associated optimization problem:

$$\min_{\theta\in\Theta} \mathcal{L}(\mathcal{D};\Psi_\theta). \tag{14}$$

Sampling method. Since we must deal with a four-dimensional state space, using a grid-based approach to determine $\mathcal{D}$ is computationally infeasible. For instance, using just 20 points per dimension would require $N = 20^4$ total points.

Instead, we design the following dynamic sampling:

1. **Linear samples**: We linearize the model around its DSS and, given a random sequence of shocks $\{\epsilon^i_{z,t}, \epsilon^i_{m,t}\}^T_{t=1}$, use the linearized solution to generate the associated sequence of states $S^i \equiv \{s^i_1, \cdots, s^i_T\}$. We repeat this procedure for M different realizations of the shock sequence to construct a training set: $\mathcal{D}_{\text{linear}} \equiv \{S^1, \cdots, S^M\}$.

2. **Deterministic samples**: Using $\mathcal{D}_{\text{linear}}$, we solve the optimization problem in (14) but without shocks and, hence, being able to omit the expectation operators. We use this solution to construct a new training set, denoted by $\mathcal{D}_{\text{Deterministic}}$.

3. **Initial training**: Using $\mathcal{D}_{\text{Deterministic}}$, we solve the stochastic problem described in (14), now with the shocks back in. Let $\theta^*_0 \equiv \text{argmin}_{\theta\in\Theta}\,\mathcal{L}(\mathcal{D}_{\text{Deterministic}};\Psi_\theta)$ denote the solution to this optimization problem. Using the trained $\Psi_{\theta^*_0}(\cdot)$ and simulated shocks, we construct a new training set $\mathcal{D}^{(0)}_{\text{Stochastic}}$.

4. **Iterative sampling**: Using $\mathcal{D}^{(0)}_{\text{Stochastic}}$, we solve the stochastic problem (14) again. Let

θ_1^* denote the solution to this optimization problem. Using the trained neural networks $\Psi_{\theta_1^*}(\cdot)$ and simulated shocks, we construct a new training set $\mathcal{D}^{(1)}_{\text{Stochastic}}$. We use a variant of adaptive sampling: around the points with the highest loss values, we simulate a new point. This process increases the importance of those points in the optimization problem.

5. **Convergence**: We repeat step 4 until convergence, which is defined as the loss $\mathcal{L}(\cdot;\cdot)$ falling below a predefined tolerance threshold.

Results. Figure 5 plots a simulated path of the capital, net inflation (in percentage points), labor, inefficiency due to price rigidities, consumption, and expected discounted marginal cost once our solution has converged.

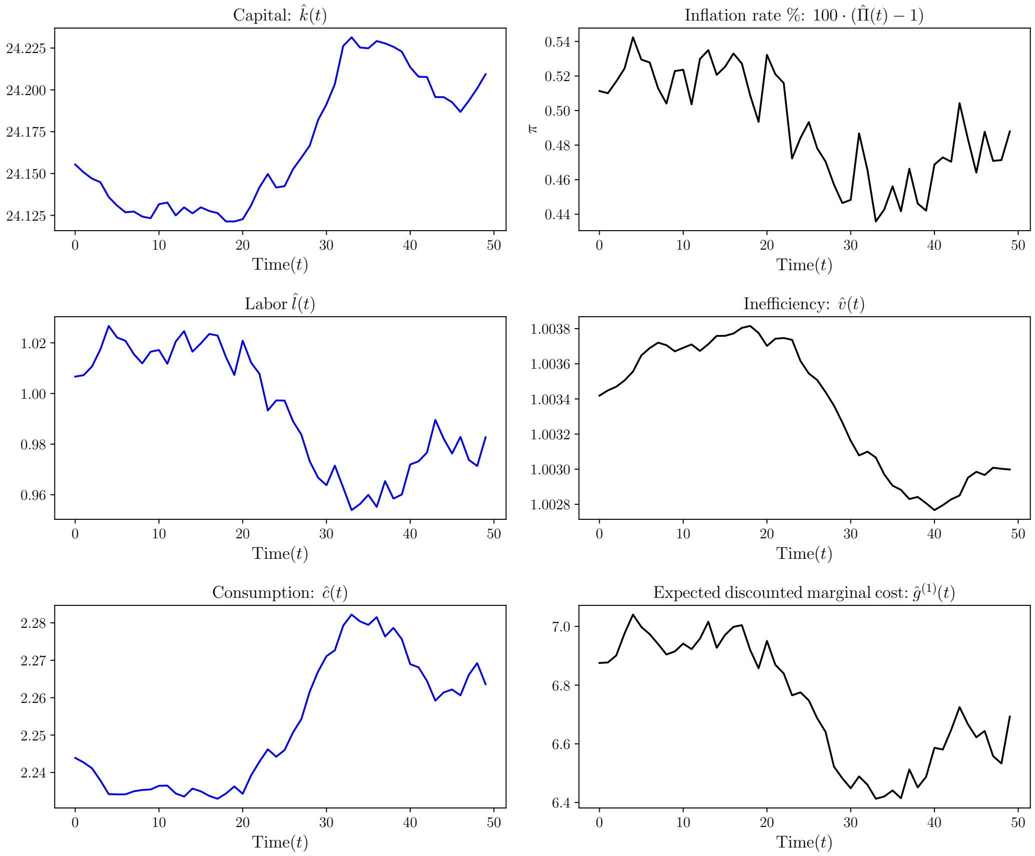

Figure 5: The path for $\widehat{k}(t)$, net inflation (in percentage points), $\widehat{l}(t)$, $\widehat{v}(t)$, $\widehat{c}(t)$, and $\widehat{g}^{(1)}(t)$.

Since, at its core, the New Keynesian model is a neoclassical growth model with nominal rigidities added to it, the intuition from Section 4 still applies here: the DNN solution picks the path where the marginal utility of consumption remains bounded and, consequently, where the valuation of prices by firms and their reset policy is also bounded.

Comparison with linear solution. Figure 6 compares the DNN and linear solutions for $\widehat{k}(t)$, net inflation (in percentage points), $\widehat{l}(t)$, $\widehat{v}(t)$, $\widehat{c}(t)$, and $\widehat{g}^{(1)}(t)$. The solid lines show the median solution across 100 different random initializations of the DNNs.

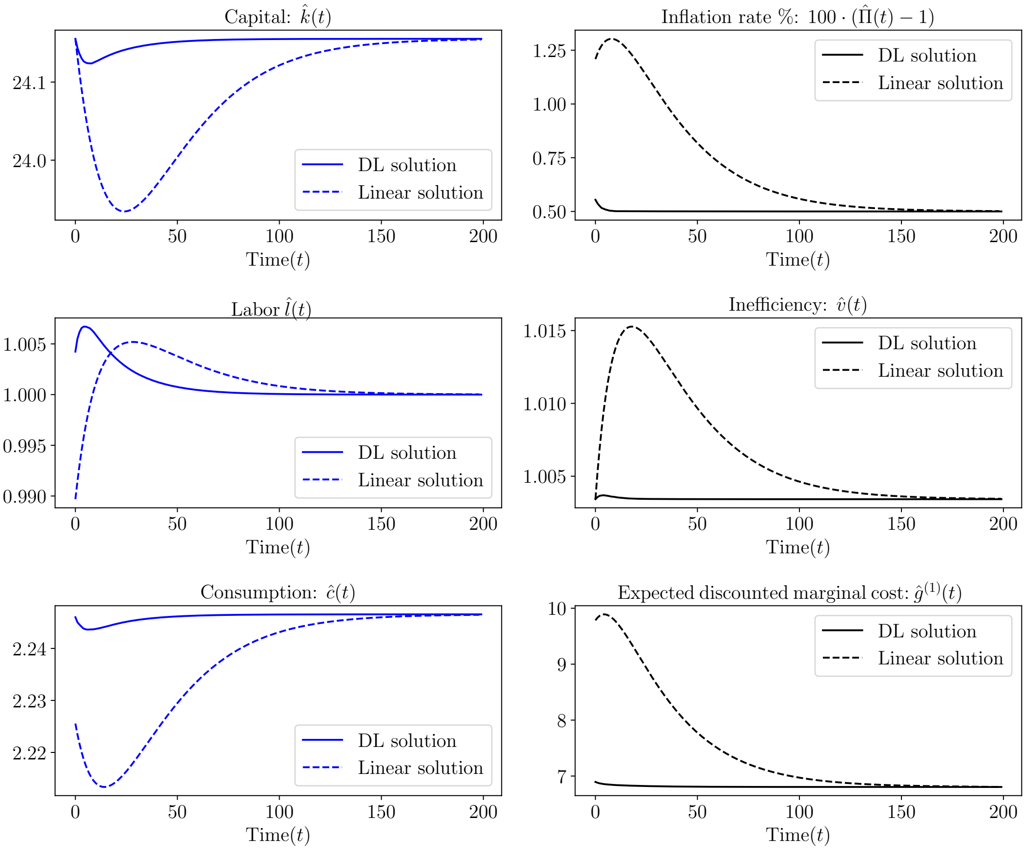

Figure 6: IRFs for $\widehat{k}(t)$, net inflation (in percentage points), $\widehat{l}(t)$, $\widehat{v}(t)$, $\widehat{c}(t)$, and $\widehat{g}^{(1)}(t)$ in response to a technology shock of $z_0 = -\sigma_z$. The DNN solution is labeled as "DL solution," and the linear solution is labeled as "Linear solution."

The figure shows that the differences in impulse-response functions (IRFs) between the two solutions are materially significant. More specifically, we consider a sequence of shocks of the form $(z_0, 0, \cdots, 0)$, where $z_0 = -\sigma_z$.

There are two notable differences between the IRFs of the DL and linear solutions: (1) inflation and (2) expected discounted marginal cost. The linear solution predicts inflation up to twice as high as the DL solution.

This difference is due to the high degree of nonlinearity in the equilibrium condition determining the ratio of the reset price of firms that can change their prices over p_t: $\Pi_t^* = \left(\frac{1-\vartheta\Pi_t^{\varepsilon-1}}{1-\vartheta}\right)^{\frac{1}{1-\varepsilon}}$ (see the Supplementary Material for the derivation). Since $\varepsilon \geq 1$, the exponent is a negative non-integer. As a result, inflation cannot exceed the upper bound $\overline{\Pi} \equiv \left(\frac{1}{\vartheta}\right)^{\frac{1}{\varepsilon-1}}$, and we have $\lim_{\Pi_t \to \overline{\Pi}} \Pi_t^* = \infty$ (with our calibration, $\overline{\Pi} \approx 1.025$). Because intermediate good firms' price setting depends on inflation and its expectations, this discrepancy also appears in the expected discounted marginal cost and revenue, i.e., $g^{(1)}$ and $g^{(2)}$, which explains the divergence in $\widehat{g}^{(1)}(t)$ between the two solutions.

Since DNN algorithms involve non-convex optimization, it is also important to ensure that we approximately find the same policy functions (i.e., if θ_1^* and θ_2^* are found from different random seeds, then $g_{\theta_1^*}(\cdot) \approx g_{\theta_2^*}(\cdot)$ for relevant regions of the state space). Thus, we solve the optimization problem using 100 different random initializations.

Figure 7 shows the loss function (13) for the DNN and linear solutions for the same experiment illustrated in Figure 6, i.e., the equilibrium path generated by a technology shock $(z_0, 0, \cdots, 0)$, where $z_0 = -\sigma_z$.

In both solutions, calculating the loss function requires computing an expectation operator. This involves evaluating $l(\cdot)$, $k'(\cdot)$, $\lambda(\cdot)$, $\Pi(\cdot)$, and $g^{(1)}(\cdot)$ across different states (particularly over ϵ_z and ϵ_m). In the DNN solution, these functions are generated by the

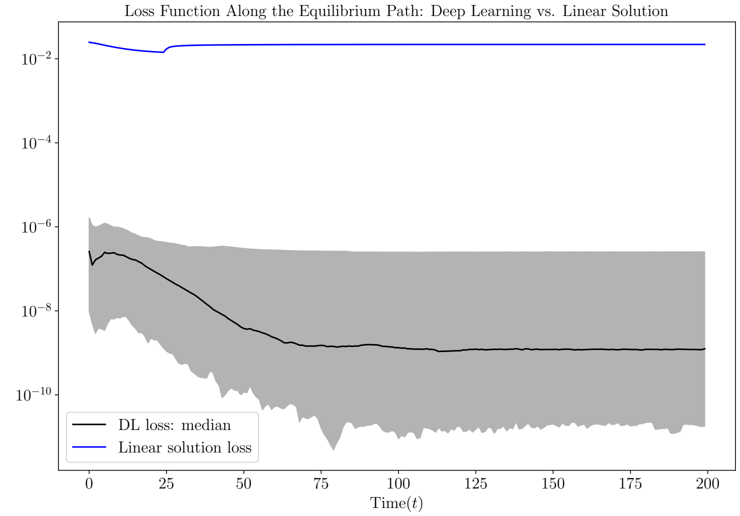

Figure 7: Comparison of the loss function in equation (13), for the DL solution ("DL Loss") and the linear solution across 100 different random seeds. The shaded region represents the 10th and 90th percentiles across seeds.

DNNs themselves. In contrast, in the linear solution, they are linear functions of the state variables, expressed as log deviations from the steady state. Therefore, even at the DSS and after a complete training of the DNN, both solutions yield a non-zero loss.

The results show that (1) using the loss function as a measure of accuracy, the DNN solution is 10^4–10^5 times more accurate than the linear solution, and (2) despite the non-convex nature of the optimization problem, the performance and accuracy of the DNN solution are not sensitive to the initialization of the neural network parameters.

This robustness to different random seeds is a direct consequence of a robust property of deep learning optimization, called "mode connectivity." This property implies that different local minima (or "modes") found when training DNNs are connected by simple, low-loss paths in the space of weights. Intuitively, this means that we do not have isolated minima, but rather flat valleys, and that all the solutions within those valleys deliver roughly the same accuracy. See Belkin (2021) and Wilson (2025) for details.

Taking stock. Our results in this section show that we can apply our approach to an otherwise standard New Keynesian model and obtain much higher accuracy than a standard linearized solution. This higher accuracy leads to meaningfully different economic answers, such as in the reported IRFs of Figure 6.

Whether this additional accuracy exceeds that of other nonlinear solution methods is not relevant for our purposes (and would depend on the details of the other method, e.g., how many Chebyshev polynomials one uses or how one choses collocation points). Our point is that the New Keynesian model can be solved with extremely high accuracy while ignoring its transversality conditions, thanks to the inductive bias of ML, and that this is of interest in many real-life research applications.

6 Conclusion

This paper has presented a theoretical framework and three applications to examine how ML methods can solve short-term transition dynamics while remaining consistent with forward-looking expectations. The central insight is that transversality conditions –essential boundary conditions that ensure the consistency of forward-looking expectations– can be approximately satisfied through the inductive bias present in ML algorithms. This suggests that ML may help solve high-dimensional problems that would otherwise be infeasible due to the curse of dimensionality.

References

Andreasen, M. M., J. Fernández-Villaverde, and J. F. Rubio-Ramírez (2018): "The pruned state-space system for non-linear DSGE models: Theory and empirical applications," *Review of Economic Studies*, 85, 1–49.

Azinovic, M., L. Gaegauf, and S. Scheidegger (2022): "Deep Equilibrium Nets," *International Economic Review*, 63, 1471–1525.

Barnett, M., W. Brock, L. P. Hansen, R. Hu, and J. Huang (2023): "A deep learning analysis of climate change, innovation, and uncertainty," Papers 2310.13200, arXiv.org.

Belkin, M. (2021): "Fit without fear: Remarkable mathematical phenomena of deep learning through the prism of interpolation," *Acta Numerica*, 30, 203–248.

Belkin, M., D. Hsu, S. Ma, and S. Mandal (2019): "Reconciling modern machine-learning practice and the classical bias–variance trade-off," *Proceedings of the National Academy of Sciences of the United States of America*, 116, 15849–15854.

Bianchi, F., S. C. Ludvigson, and S. Ma (2022): "Belief distortions and macroeconomic fluctuations," *American Economic Review*, 112, 2269–2315.

Blanc, G., N. Gupta, G. Valiant, and P. Valiant (2020): "Implicit regularization for deep neural networks driven by an Ornstein-Uhlenbeck like process," *Proceedings of Machine Learning Research vol*, 125, 1–31.

Blanchard, O. J. and C. M. Kahn (1980): "The solution of linear difference models under rational expectations," *Econometrica*, 48, 1305–1311.

Chiang, P.-y., R. Ni, D. Y. Miller, A. Bansal, J. Geiping, M. Goldblum, and T. Goldstein (2022): "Loss landscapes are all you need: Neural network generalization can be explained without the implicit bias of gradient descent," in *The Eleventh International Conference on Learning Representations*.

Damian, A., T. Ma, and J. D. Lee (2021): "Label noise SGD provably prefers flat global minimizers," *Advances in Neural Information Processing Systems*, 34, 27449–27461.

Duarte, V., D. Duarte, and D. Silva (2024): "Machine learning for continuous-time finance," Working paper, CESifo.

Ebrahimi Kahou, M., J. Fernández-Villaverde, J. Perla, and A. Sood (2021): "Exploiting symmetry in high-dimensional dynamic programming," Working Paper 28981, National Bureau of Economic Research.

Ebrahimi Kahou, M., J. Yu, J. Perla, and G. Pleiss (2024): "How inductive bias in machine learning aligns with optimality in economic dynamics," Tech. Rep. 2406.01898, arXiv.org.

Ekeland, I. and J. A. Scheinkman (1986): "Transversality conditions for some infinite horizon discrete time optimization problems," *Mathematics of Operations Research*, 11, 216–229.

Evans, G. W. and S. Honkapohja (2001): *Learning and Expectations in Macroeconomics*, Princeton University Press.

Fernández-Villaverde, J. and P. A. Guerrón-Quintana (2021): "Estimating

DSGE models: Recent advances and future challenges," *Annual Review of Economics*, 13, 229–252.

Fernández-Villaverde, J., J. Rubio-Ramírez, and F. Schorfheide (2016): "Solution and estimation methods for DSGE models," in *Handbook of Macroeconomics*, ed. by J. B. Taylor and H. Uhlig, Elsevier, vol. 2, 527–724.

Fernández-Villaverde, J. and J. F. Rubio-Ramírez (2006): "A baseline DSGE model," Unpublished manuscript. Available at https://www.sas.upenn.edu/~jesusfv/benchmark_DSGE.pdf.

Fernández-Villaverde, J., S. Hurtado, and G. Nuño (2023): "Financial frictions and the wealth distribution," *Econometrica*, 91, 869–901.

Han, J., Y. Yang, and W. E (2022): "DeepHAM: A global solution method for heterogeneous agent models with aggregate shocks," Tech. Rep. 2112.14377, arXiv.org.

Hastie, T., A. Montanari, S. Rosset, and R. J. Tibshirani (2022): "Surprises in high-dimensional ridgeless least squares interpolation," *Annals of Statistics*, 50, 949.

Jungerman, W. (2023): "Dynamic Monopsony and Human Capital," Working Paper.

Kamihigashi, T. (2005): "Necessity of the transversality condition for stochastic models with bounded or CRRA utility," *Journal of Economic Dynamics and Control*, 29, 1313–1329.

Kase, H., L. Melosi, and M. Rottner (2022): "Estimating nonlinear heterogeneous agents models with neural networks," Discussion Paper 17391, CEPR.

Klein, P. (2000): "Using the generalized Schur form to solve a multivariate linear rational expectations model," *Journal of Economic Dynamics and Control*, 24, 1405–1423.

LeCun, Y., L. Bottou, G. B. Orr, and K.-R. Müller (2002): "Efficient backprop," in *Neural networks: Tricks of the trade*, Springer, 9–50.

Ljungqvist, L. and T. J. Sargent (2018): *Recursive Macroeconomic Theory*, MIT Press, 4 ed.

Ma, C. and L. Ying (2021): "On linear stability of SGD and input-smoothness of neural networks," *Advances in Neural Information Processing Systems*, 34, 16805–16817.

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

Marimón, R. (1989): "Stochastic turnpike property and stationary equilibrium," *Journal of Economic Theory*, 47, 282–306.

McKenzie, L. W. (1976): "Turnpike theory," *Econometrica*, 841–865.

Obstfeld, M. and K. Rogoff (1983): "Speculative hyperinflations in maximizing models: Can we rule them out?" *Journal of Political Economy*, 91, 675–687.

Payne, J., A. Revei, and Y. Yang (2024): "Deep learning for search and matching models," Tech. Rep. 4768566, SSRN.

Sargent, T. J. (1993): *Bounded Rationality in Macroeconomics*, Oxford University Press.

——— (2024): "Macroeconomics after Lucas," Working Paper.

Sargent, T. J. and N. Wallace (1973): “The stability of models of money and growth with perfect foresight,” *Econometrica*, 1043–1048.

Skiba, A. K. (1978): “Optimal growth with a convex-concave production function,” *Econometrica*, 527–539.

Smith, S. L., B. Dherin, D. Barrett, and S. De (2021): “On the origin of implicit regularization in stochastic gradient descent,” in *International Conference on Learning Representations*.

Spiess, J., G. Imbens, and A. Venugopal (2023): “Double and single descent in causal inference with an application to high-dimensional synthetic control,” Working Paper 31802, National Bureau of Economic Research.

Van, C. L., R. Boucekkine, and C. Saglam (2007): “Optimal control in infinite horizon problems: a Sobolev space approach,” *Economic Theory*, 32, 497–509.

Wilson, A. G. (2025): “Position: Deep learning is not so mysterious or different,” in *Forty-second International Conference on Machine Learning Position Paper Track*.

Zhang, C., S. Bengio, M. Hardt, B. Recht, and O. Vinyals (2021): “Understanding deep learning (still) requires rethinking generalization,” *Communications of the ACM*, 64, 107–115.

Spooky Boundaries at a Distance: Supplementary Material

Mahdi Ebrahimi Kahou[1] Jesús Fernández-Villaverde[2]
Sebastián Gómez-Cardona[3] Jesse Perla[4] Jan Rosa[4]

August 23, 2025

This Supplementary Material contains additional material to the paper "Spooky Boundaries at a Distance." We refer to the main paper for notation.

1 Robustness for the neoclassical growth model

This section describes further robust analysis for the neoclassical growth model in Section 4 of the main text.

Sparse grids. In our baseline example, we choose $\mathcal{D} \equiv \{0, \dots, 29\}$ and minimize equation (12) of the main text to find a $k_\theta(t)$ where $|\theta| \approx 40,000$. Here, we use a sparser set of grid points and interpolate when $t \notin \mathcal{D}$, while keeping the rest of the algorithm unchanged. In particular, consider a grid with more points close to the area with high curvature and fewer closer to the steady state, $\mathcal{D}^{\text{Sparse 1}} \equiv \{0, 1, 2, 4, 6, 8, 12, 16, 20, 24, 29\}$, and another grid with fewer points spread evenly over the domain, $\mathcal{D}^{\text{Sparse 2}} \equiv \{0, 1, 4, 8, 12, 16, 20, 24, 29\}$.

Figure 1.1 shows the results of these two experiments for an ensemble of 100 random seeds. The left panel compares the benchmark solution $k(t)$ with $k_\theta(t)$ for $\mathcal{D}^{\text{Sparse 1}}$ and $\mathcal{D}^{\text{Sparse 2}}$. The right panel compares the benchmark $c(t)$ against the corresponding $c_\theta(t)$. In both cases, the shaded areas show the 10th and 90th percentiles.

[1] Bowdoin College, [2] University of Pennsylvania, [3] Morningstar, and [4] University of British Columbia, Vancouver School of Economics.

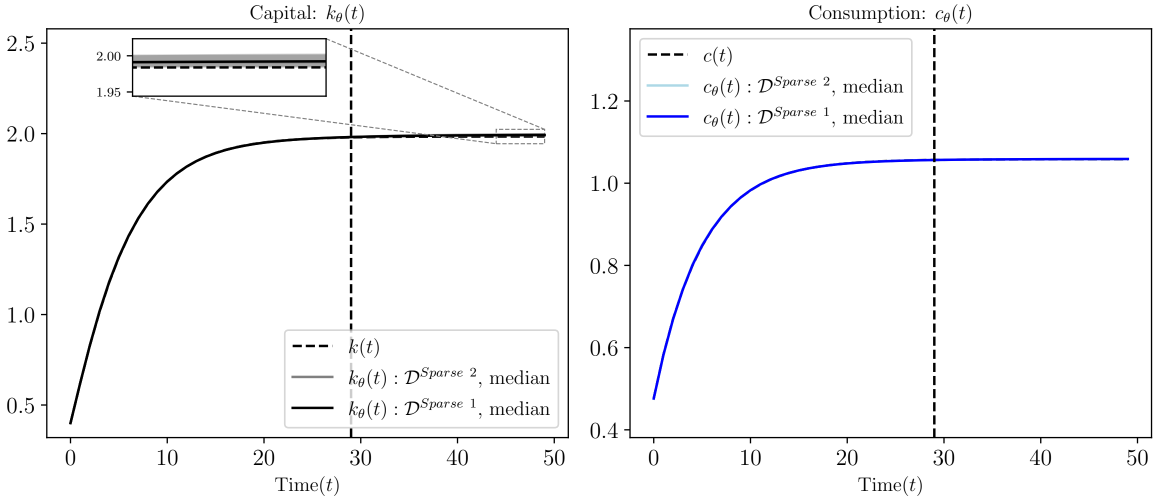

Figure 1.1: Solutions to equation (12) of the main text with $\mathcal{D}^{\text{Sparse 1}}$ and $\mathcal{D}^{\text{Sparse 2}}$.

The distribution of the relative error of $k_\theta(t)$ is small, even in the extrapolation region. In the case of $c_\theta(t)$, the error is so small that the 10th and 90th percentile ranges are not visible. This experiment establishes that we can achieve very accurate solutions with sparse grids, even though the problem remains overparameterized by around four orders of magnitude. ML algorithms do not intrinsically require a large amount of data as long as they have a strong inductive bias.

Solving on a short horizon. A challenge in solving for transition dynamics of models with classic algorithms, such as shooting methods, is the difficulty in choosing the T at which point the solution is close to a deterministic steady state (DSS). If T is too small, we move toward the DSS too quickly. If T is too large, numerical instabilities can accumulate as the solution iterates forward. Choosing the value of T is an art and requires a good prior on the speed of convergence for a particular model.

To test whether this concern holds with our methods, we solve our model by minimizing equation (12) of the main text with the same algorithm as in our baseline case, but choose $\mathcal{D} \equiv \{0, 1, \cdots, 9\}$. Not only are there few grid points, but the $t_N = 9$ is far below the point of convergence to the DSS.

Figure 1.2 shows the results of this experiment for an ensemble of 100 random seeds. The left panel shows the median of the approximate capital paths, $k_\theta(t)$, and the benchmark solution. The right panel shows the median of the approximate consumption paths, $c_\theta(t)$, and the benchmark solution. The shaded areas represent the 10th and 90th percentiles.

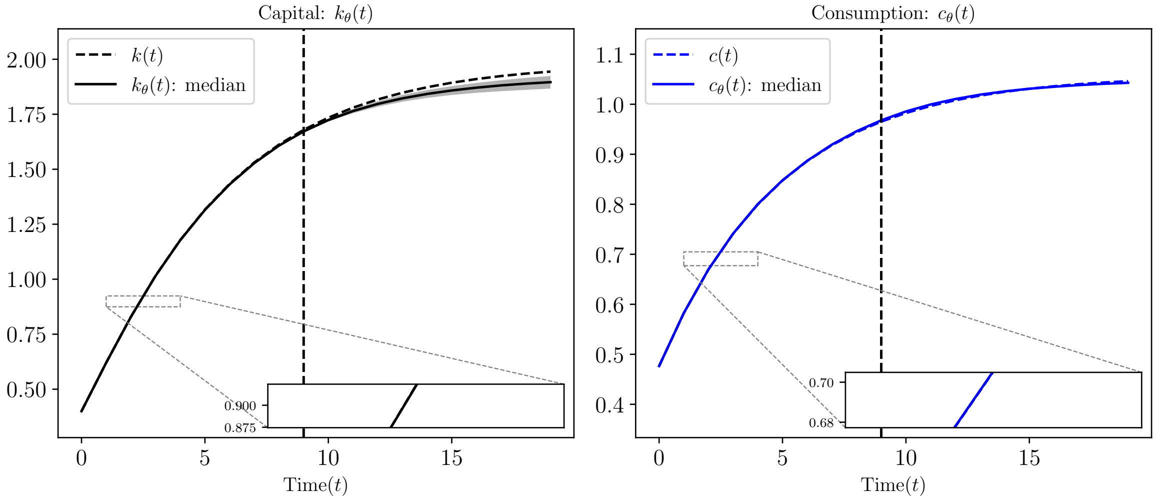

Figure 1.2: Solutions to equation (12) with $\mathcal{D} \equiv \{0, 1, \cdots, 9\}$.

The conclusion is that for the short- to medium-run dynamics, the solutions are very accurate, and the lack of grid points close to the DSS does not feed back into large errors in the short run (as it would with a shooting method). The extrapolation errors are larger than in the baseline case, but getting the long run right was not the goal of the exercise. This experiment suggests that ML methods relying on inductive bias are not very sensitive to choosing data close to the DSS, as long as they are not used to extrapolate too far out of the sample.

Learning the scaling factor. When designing the deep neural network (DNN) with a balanced growth path (BGP), we added a learnable rescaling: $k_\theta(t) = \exp(\phi t)\mathrm{NN}(t; \theta_{\mathrm{NN}})$, where $\theta \equiv \{\phi, \theta_{\mathrm{NN}}\}$. Given a $\mathcal{D}$ with a large maximum value t_N, the min-norm solution for $\mathrm{NN}(t; \theta_{\mathrm{NN}})$ is achieved by setting $\phi = \log(1 + g)$, at which point $\mathrm{NN}(t; \theta_{\mathrm{NN}})$ can be non-explosive.

However, if t_N is relatively small, we would not expect the approximation to exactly choose $\phi = \log(1+g)$. A smaller ϕ might yield a lower norm $\mathrm{NN}(t; \theta_{\mathrm{NN}})$ for interpolating a particular $\mathcal{D}$. How well, then, does the algorithm learn g?

Figure 1.3 plots a histogram of the approximated g in each of the 100 runs of the DNN (each with a different seed) and compares them to the true growth rate, $g = 0.02$. The results show that the min-norm is biased toward smaller growth rates, as expected given it is trained on a bounded $\mathcal{D}$. However, the solutions remain extremely accurate. The variations in ϕ within Figure 1.3 are compensated by changes in $\mathrm{NN}(t; \theta_{\mathrm{NN}})$. A very accurate approximation of the growth rate is not necessary to achieve accurate short- and medium-run dynamics.

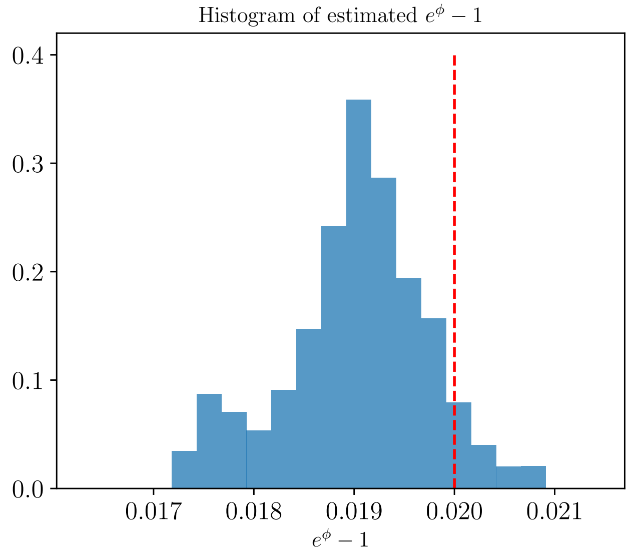

Figure 1.3: The distribution of the learned $e^{\phi}-1$ for the ensemble of 100 seeds used in solving equation (12); $g = 0.02$, shown as the dashed line.

Learning a misspecified DNN In the main text, we used economic insights to choose a DNN that included a term for exponential growth. A natural question is: Is it still helpful to suggest a problem structure when designing the DNN if the suggestion is misspecified?

To analyze this case, we solve a version where the scaling is assumed to be linear rather than exponential. In particular, $k_\theta(t) = t \cdot \mathrm{NN}(t;\theta) + k_0$. The linear scaling allows some degree of growth, but as $t_N \to \infty$, the $\mathrm{NN}(t;\theta)$ would still need to have an infinite norm in order to capture the true dynamics of the BGP.

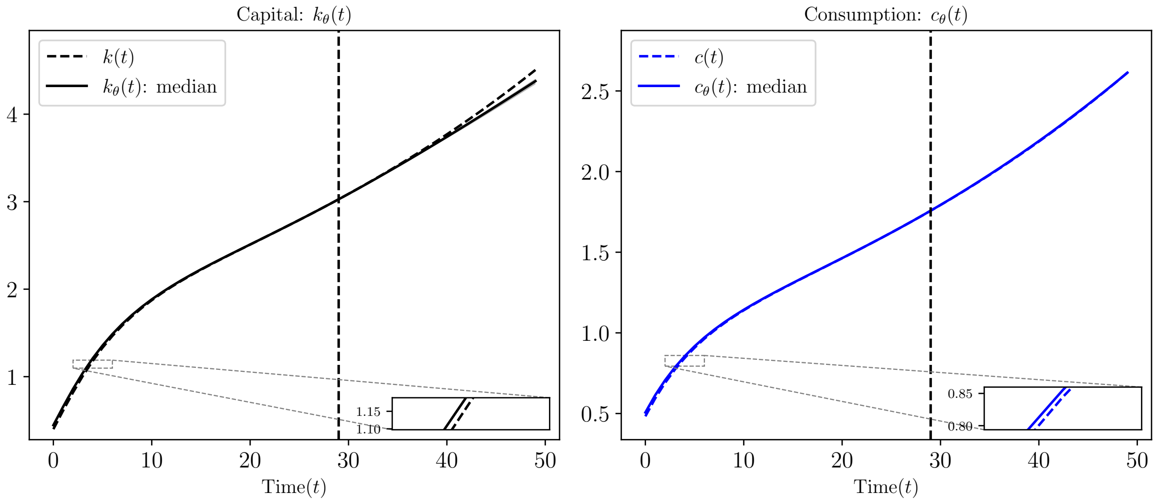

Figure 1.4: Solutions to equation (12) with the misspecified $k_\theta(t) = t \cdot \mathrm{NN}(t;\theta) + k_0$ and $g = 0.02$.

Figure 1.4 displays the solutions to the equation with this specification for 100 random seeds. The left panel shows the benchmark and the median of the solution for capital, while the right panel does the same for consumption. Although the 10th and 90th percentiles are included, they are so close to each other that they remain indistinguishable even after zooming in.

Compared to the well-specified case, the long-run extrapolation slowly diverges (and would continue to do so for any finite t_N), but this does not cause any issues for the short- and medium-run dynamics.

Function norms and the transversality condition. When relying on the inductive bias of the function norms in lieu of the transversality condition, we must argue that $\|k_{\max}\|_\psi > \|k\|_\psi$ for a large class of norms ψ.

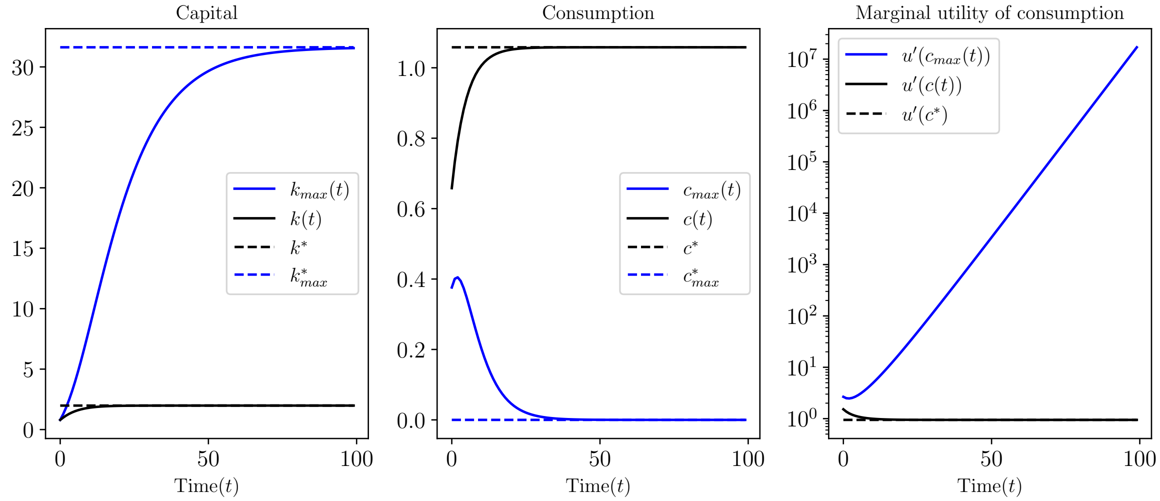

Figure 1.5: Comparison between the optimal solution and those violating the transversality condition.

To see this, Figure 1.5 plots the two solutions to the under-determined system. The blue curves show a set of capital, consumption, and marginal utility paths, denoted respectively by $k_{\max}(t)$, $c_{\max}(t)$, and $u'(c_{\max}(t))$, that violate the transversality condition. The black curves show the optimal paths that satisfy the transversality condition and eventually converge to k^*, c^*. Focusing on the left panel, we see that the path of the $k_{\max}(t)$ function has much steeper changes than that of $k(t)$. Therefore, for a large class of norms and seminorms, which penalize either the average level or gradients, we have $\|k_{\max}\|_\psi > \|k\|_\psi$.

The middle and right panels of Figure 1.5 also provide intuition on when these methods can fail. While $\|k_{\max}\|_\psi > \|k\|_\psi$ for a large class of norms given the big spread between k^* and $k^*_{\max}$, this is not the case for $c(t)$. If a norm penalizes the gradients (e.g., $\int_0^T |c'(t)|\, dt$), then the norms of $\|c_{\max}\|_\psi$ and $\|c\|_\psi$ would be similar. If the level enters the norm, it may even bias the solution toward the wrong answer (i.e., where $c^*_{\max} = 0$). The right panel shows the other extreme, where using the marginal utility makes an even starker difference between the two solutions. We will revisit this issue in the next section and offer guidance on how to avoid these problems.

2 State-Space Formulation

This section describes the recursive state-space formulation of the neoclassical growth model, in contrast to our baseline sequence-space representation. Inductive bias will serve a similar role in providing a sufficiency condition for transversality, but those conditions will involve norms of the policy functions rather than the trajectories themselves. All model primitives and parameters remain the same as in the baseline.

Model. For the state space $(k, z) \in \mathbb{R}^2_+$, equations (9) and (10) in the main text become:

$$u'(c) = u'(c')\beta\left[z'^{1-\alpha} f'(k') + 1 - \delta\right] \tag{1}$$

$$k' = z^{1-\alpha} f(k) + (1-\delta)k - c, \tag{2}$$

where k', c', and z' are the next period capital, consumption, and technology, respectively, and $u(c) = \log c$.

The transversality condition takes the form now:

$$\lim_{T\to\infty} \beta^T u'\left(c_T(k_0, z_0)\right) k_{T+1}(k_0, z_0) = 0 \quad \text{for all } (k_0, z_0) \in \mathbb{R}^2_+. \tag{3}$$

In this notation, $k_{T+1}(k_0, z_0)$ requires iterating the $k'(\cdot,\cdot)$ policy and $z' = (1+g)z$ law of motion $T+1$ times from (k_0, z_0). Consumption, $c_T(k_0, z_0)$, is found by first iterating to find (k_T, z_T) and then using equation (2) to calculate $c_T = z_T^{1-\alpha} f(k_T) + (1-\delta)k_T - k'(k_T, z_T)$.

Transversality with classic methods. The iteration of the policy $k'(\cdot,\cdot)$ in equation (3) links stability and transversality. If $k'(\cdot,\cdot)$ were explosive (e.g., $|\nabla_k k'(k, z)| > 1$ for k and z above some threshold), capital would grow until it asymptotically approached the capital that maximizes the BGP (or $k^*_{\max}$ if $g = 0$) via equation (1). This would lead to an infinite marginal utility of consumption in equation (3), thereby violating transversality.

In practice, classical methods do not apply the transversality condition as a limit, but enforce it indirectly in several ways:

- For sequence-space methods, a DSS is found (possibly after detrending the BGP), which is then used as a terminal boundary condition with shooting methods. These approaches implicitly use the transversality condition when solving for the correct DSS.
- Linear rational expectations models and linear-quadratic control select the non-explosive root via spectral methods.
- With global solution methods such as projection and collocation, transversality is implicitly fulfilled by restricting the state space domain. For example, in the growth model, we might approximate with Chebyshev polynomials on a compact hypercube $[k_{\min}, \bar{k}] \times [z_{\min}, \bar{z}]$. If $\bar{k} < k^*_{\max}$ and $k_{\min} < k^*$, then policy functions violating transversality are rejected since they cannot satisfy the Euler equation before hitting the boundaries. Alternatively, bounding $c \geq c_{\min} > 0$ implicitly enforces transversality by ensuring $u'(c) \leq u'(c_{\min}) < \infty$.

The min-norm solution. We approximate the capital policy, $k'_\theta(\cdot,\cdot) \in \mathcal{H}(\Theta)$, using a DNN. Choose $\mathcal{D} \subset \mathbb{R}^2_+$ with N points and minimize the equivalent of equation (12):

$$\min_{\theta \in \Theta} \frac{1}{N} \sum_{(k,z)\in\mathcal{D}} \left[\frac{u'\big(c(k,z;k'_\theta)\big)}{u'\big(c(k'_\theta(k,z),(1+g)z;k'_\theta)\big)} - \beta\big[(1+g)zf'\big(k'_\theta(k,z)\big) + 1 - \delta\big] \right]^2. \tag{4}$$

Consumption $c\big(k,z;k'_\theta\big) \equiv f(k) + (1-\delta)k - k'_\theta(k,z)$ is defined through the feasibility constraint for a given policy for capital $k'_\theta(\cdot,\cdot)$.

Then, we can think of solutions to equation (4) as finding:

$$\min_{k'_\theta \in \mathcal{H}(\Theta)} ||k'_\theta||_\psi \tag{5}$$

$$\text{s.t.} \quad \frac{u'\big(c(k,z;k'_\theta)\big)}{u'\big(c(k'_\theta(k,z),(1+g)z;k'_\theta)\big)} = \beta\big[(1+g)zf'\big(k'_\theta(k,z)\big) + 1 - \delta\big], \quad \forall (k,z) \in \mathcal{D}. \tag{6}$$

The norm in problem (5) typically depends on gradients, reflecting its bias toward flat solutions. For example, it might have properties similar to $||k'_\theta||^2_{W^{1,2}} \equiv \int ||\nabla k'_\theta(k,z)||^2_2 \, dF(k,z)$, a Sobolev seminorm, for some measure F over the state space, or over a compact subset of the domain.

To informally argue why this bias selects the non-explosive solution, consider iterating the policy $k_{t+1} = k'_\theta(k_t, z_t)$. A bias toward smaller gradients, with $|\nabla_k k'_\theta(k, z)| < 1$ for large k, produces policies with smaller changes in capital, $k_{t+1} - k_t$. If a DSS exists, the iteration converges to the fixed point $k_t \approx k'_\theta(k_t, z_t)$. Forward iteration under this bias yields trajectories that satisfy the transversality condition.

Results. We solve the minimization problem (4) for $\beta = 0.9$, $\alpha = 0.33$, $\delta = 0.1$, $g = 0$, $z_0 = 1$, and $k_0 = 0.4$. In our baseline case, $\mathcal{D}$ is a uniform grid of 16 points between $k_1 = 0.8$ and $k_{N_k} = 2.5$. When $g \neq 0$, we can use a grid $\mathcal{D} \equiv \{k_1, \cdots, k_{N_k}\} \times \{z_1, \cdots, z_{N_z}\}$ of $N = N_z \times N_k$ total points, but the methods could use sampled or simulated points in the state space. The design of $\mathcal{H}(\Theta)$ is a DNN identical to the sequential version of the model, except that it takes two inputs (k, z) rather than the univariate t. As before, we solve with the L-BFGS optimization algorithm, which is fast and requires little tuning.

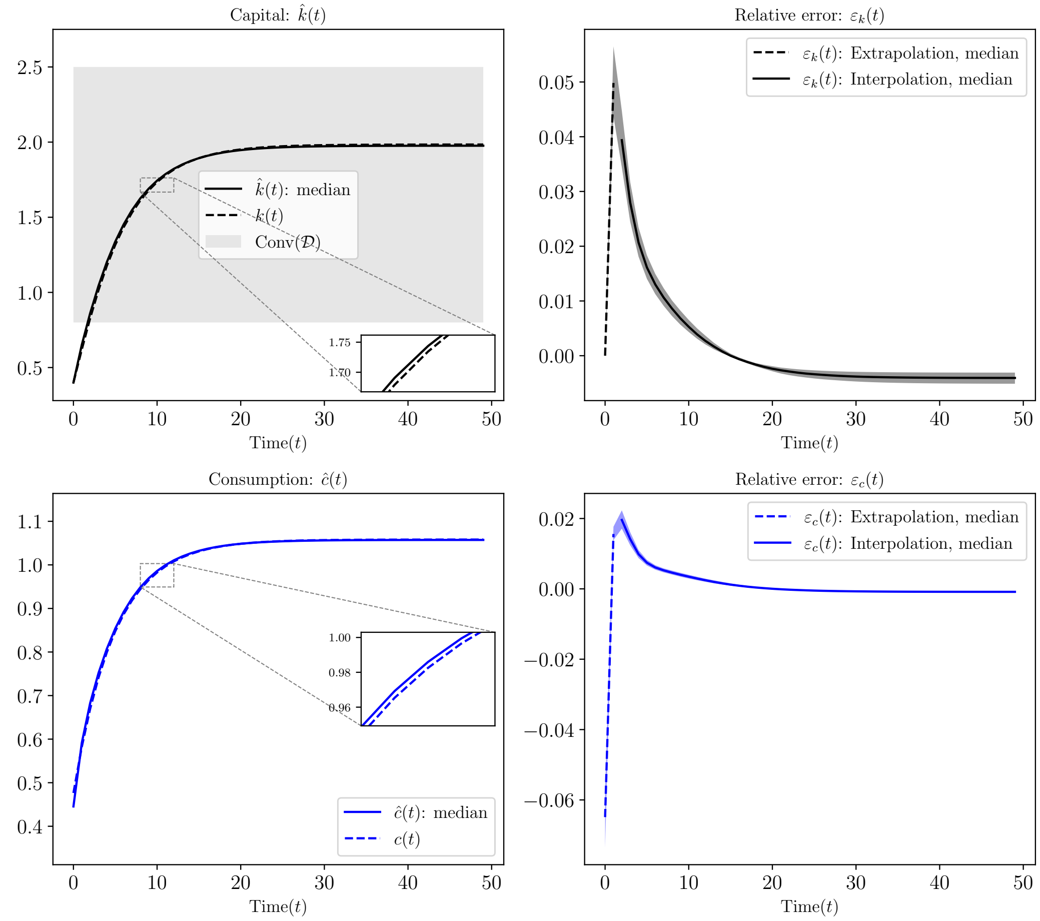

Figure 2.1: Solutions obtained by solving problem (4) for $g = 0$.

Figure 2.1 shows the median solutions for capital (top row) and consumption (bottom row) across an ensemble of 100 random seeds. The consumption path $\tilde{c}(t)$ is computed from the resource constraint given the trajectory of the state variables. Benchmark solutions, $k(t)$ and $c(t)$, are obtained using value function iteration.

The left panels plot the median of the approximate capital, $\widehat{k}(t)$, and consumption, $\widehat{c}(t)$, together with the benchmark solutions (i.e., $\widehat{k}(t)$ and $\widehat{c}(t)$ result from iterating the solution from a given initial condition). The right panels show the median relative errors for capital, $\varepsilon_k(t) \equiv (\widehat{k}(t) - k(t))/k(t)$, and consumption, $\varepsilon_c(t) \equiv (\widehat{c}(t) - c(t))/c(t)$. Shaded regions correspond to the 10th and 90th percentiles. The gray area in the top-left panel marks the interpolation region, defined as the convex hull of $\mathcal{D}$. Dashed curves indicate the median relative errors in the extrapolation region.

The results show that the inductive bias eliminates solutions that violate the transversality condition and achieves high accuracy using only 16 data points. Even when k_0 lies outside the minimum value of $\mathcal{D}$, errors remain small, and the inductive bias delivers good generalization beyond $\text{Conv}(\mathcal{D})$.

BGP. Since the solution is homothetic when $g = 0.02$, we design our DNN as $k'_\theta(k, z) \equiv z \cdot \text{NN}(k/z, z; \theta)$. We set $\mathcal{D}$ as the Cartesian product of 16 points in $[0.8, 3.5]$ for capital and 8 points in $[0.8, 1.8]$ for technology. As before, using a small $\mathcal{D}$ highlights the strength of the inductive bias. This implementation minimizes problem (4) with different choices of $\mathcal{D}$, running 100 seeds for the optimizer's initial condition.[1]

Figure 2.2 shows the results for a simulated trajectory starting from $k_0 = 0.4$ and $z_0 = 1$, compared with the benchmark solution. The left panel reports the median of the approximate capital path, $\widehat{k}(t)$, and the right panel reports the median of the approximate consumption path, $\widehat{c}(t)$. The shaded regions represent the 10th and 90th percentiles.

The results indicate that, even with a growing technology, the solution is highly accurate in the short run, and the differences from the benchmark are barely visible, even after zooming in. The long-run extrapolation is less precise than in the benchmark case (where manual rescaling was possible due to homotheticity). In sum, very accurate short- and medium-run solutions can be obtained even when the initial capital stock lies outside the interpolation region.

[1] In the exactly homothetic case, this could be further reduced to a univariate $\text{NN}(k/z; \theta)$, but we retain the z parameter to handle nearly homothetic cases and to verify that the inductive bias prevents overfitting.

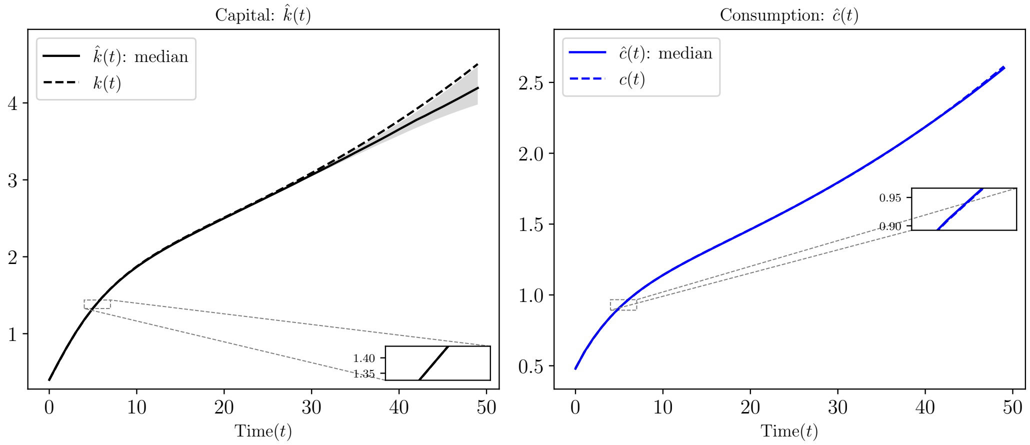

Figure 2.2: Solutions to problem 4 for $g = 0.02$.

Failures of Euler residuals minimization. Section 1 of this Supplementary Material discusses the importance of choosing a problem formulation that ensures the inductive bias toward min-norm solutions selects the path fulfilling transversality. This issue is often even more pronounced in state-space formulations, making it critical to understand before addressing high-dimensional macroeconomic problems, where transversality failures are less obvious.

We illustrate this by comparing an equivalent formulation of the neoclassical growth model, where we approximate $c_\theta(k, z)$, to our earlier results in Figures 2.1 and 2.2. As we will see, in this case, the inductive bias toward min-norm solutions consistently selects the transversality-violating path.

For simplicity, let $z = 1$ and $g = 0$, approximate $c_\theta(k)$ with a DNN, and define the implied investment choice as $k'(k; c_\theta) \equiv f(k) + (1-\delta)k - c_\theta(k)$. The equivalent of the objective in equation (4) is now:

$$\min_{\theta\in\Theta}\frac{1}{N}\sum_{k\in\mathcal{D}}\underbrace{\left[\frac{u'\big(c_\theta(k)\big)}{u'\big(c_\theta(k'(k;c_\theta))\big)}-\beta\left[f'\big(k'(k;c_\theta)\big)+1-\delta\right]\right]^2}_{\equiv\varepsilon^c_E(k)}. \tag{7}$$

Figure 2.3 compares approximating the policy function for capital $k'_\theta(\cdot)$ with approximating the consumption function $c(\cdot)$ using a DNN.[2] The left panels show the baseline k'_θ approximation from Figure 2.1, plotting the Euler error in the top panel and the policy $k'_\theta(k)$ in the bottom panel. The latter crosses the 45-degree line near $k^* \approx 2.0$, the closed-form steady state. The

[2] Primitives and parameters are identical to our baseline case. Given the parameters, the DSS solution fulfilling transversality is $k^* \approx 2.0$.

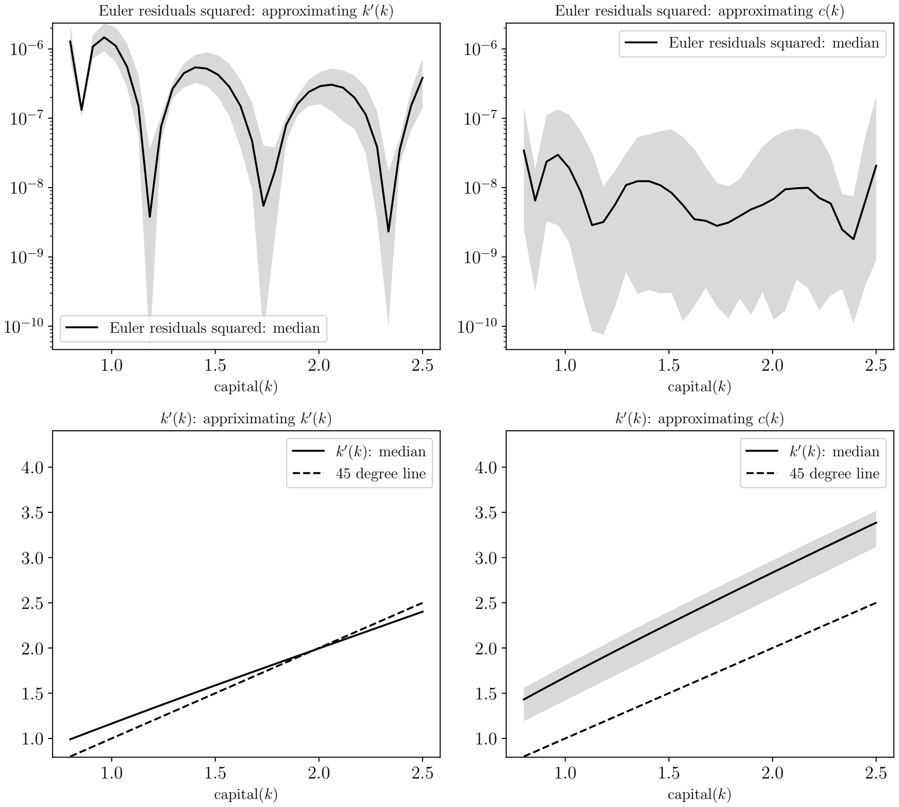

Figure 2.3: Comparison between approximating the policy function for capital $k'(k)$ vs. the consumption function $c(k)$ with a DNN.

right panels use the c_θ approximation: the top panel shows the squared Euler error from (7), and the bottom panel shows the implied policy $k'(k; c_\theta) \equiv f(k) + (1-\delta)k - c_\theta(k)$. Solid curves are medians, and shaded regions show the 10th and 90th percentiles over 100 seeds.

In both cases, Euler errors $\varepsilon_E^k(k)$ and $\varepsilon_E^c(k)$ are near numerical precision, so the optimizer finds a solution interpolating the Euler equation and implicitly fulfilling the resource constraint on $\mathcal{D}$. If anything, $\varepsilon_E^c(k)$ is smaller. However, the bottom right panel shows $k'(k; c_\theta)$ never intersecting the 45-degree line, with $\nabla_k k'(k; c_\theta) > 1$ for all k, producing explosive $\tilde{k}(t)$ paths that violate transversality.

Approximating state and co-state variables, not jump variables. Why does approximating $c_\theta(k)$ lead to solutions that violate transversality? Because the consumption path violating transversality converges to 0 and, thus, its slope is systematically smaller in absolute value. As a result, the inductive bias does not avoid these paths.

Approximating $k'_\theta(k, z)$ was not the product of luck or repeated trial and error. We approximated capital because it is the variable that appears in the transversality condition:

$$\lim_{T\to\infty} \beta^T u'\left(c_T(k_0,z_0)\right) k_{T+1}(k_0,z_0) = 0 \quad \text{for all } (k_0,z_0) \in \mathbb{R}^2_+.$$

and, therefore, we want to avoid explosive paths for capital.

Consumption also appears, but as $c(k,z)^{-1}$, i.e., as the marginal utility of consumption, which determines the shadow price of resources $\lambda(k,z) \equiv u'(c(k,z))$:

$$\lim_{T\to\infty} \beta^T u'\left(c_T(k_0,z_0)\right) k_{T+1}(k_0,z_0) = \lim_{T\to\infty} \beta^T \lambda_T\left(k_0,z_0\right) k_{T+1}(k_0,z_0) = 0 \quad \text{for all } (k_0,z_0) \in \mathbb{R}^2_+$$

However, approximating $c(k,z)$ in levels does not work because it does not guarantee that the transversality condition holds.

This reasoning teaches a more general lesson, applicable to other models and both the sequential formulation and the state-space version:

1. Low Euler (or value function) errors do not guarantee a correct solution.

2. Formulate the problem in terms of approximating state variables or co-states, not jump variables.[3]

3 The New Keynesian Model

In this section, we detail the equilibrium conditions of the New Keynesian model and how we use them to derive the equilibrium of the model once we have solved for the DNN.

Equilibrium conditions.

- First-order conditions of the household:

$$\frac{1}{c_t} = \lambda_t, \tag{8}$$

$$\psi l_t^\eta = \lambda_t w_t, \tag{9}$$

$$\lambda_t = \beta \mathbb{E}_t \left[\lambda_{t+1} \frac{R_t}{\Pi_{t+1}}\right], \tag{10}$$

$$\lambda_t = \beta \mathbb{E}_t \left[\lambda_{t+1}(1 + r_{t+1} - \delta)\right]. \tag{11}$$

[3] This last recommendation should not be a surprise. It is often the case with other solution methods that approximating state variables or co-states works better than approximating jump variables.

- First-order conditions of the firm:

$$g_t^{(1)} = \lambda_t \mathrm{mc}_t y_t + \beta\vartheta \mathbb{E}_t \left[\Pi_{t+1}^{\varepsilon} g_{t+1}^{(1)} \right], \tag{12}$$

$$g_t^{(2)} = \lambda_t \Pi_t^* y_t + \beta\vartheta \mathbb{E}_t \left[\Pi_{t+1}^{\varepsilon-1} \left(\frac{\Pi_t^*}{\Pi_{t+1}^*} \right) g_{t+1}^{(2)} \right], \tag{13}$$

$$\varepsilon g_t^{(1)} = (\varepsilon - 1) g_t^{(2)}, \tag{14}$$

$$\frac{k_{t-1}}{l_t} = \frac{\alpha}{1-\alpha} \frac{w_t}{r_t}, \tag{15}$$

$$\mathrm{mc}_t = \left(\frac{1}{1-\alpha} \right)^{1-\alpha} \left(\frac{1}{\alpha} \right)^{\alpha} \frac{w_t^{1-\alpha} r_t^{\alpha}}{e^{z_t}}. \tag{16}$$

- Price level evolution:

$$1 = \vartheta \Pi_t^{\varepsilon-1} + (1-\vartheta)(\Pi_t^*)^{1-\varepsilon}. \tag{17}$$

- Monetary policy:

$$R_t = R \left(\frac{\Pi_t}{\Pi} \right)^{\gamma_\Pi} \left(\frac{y_t}{y} \right)^{\gamma_y} e^{\sigma_m \epsilon_{m,t}}. \tag{18}$$

- Market clearing:

$$y_t = \frac{1}{v_t} e^{z_t} k_{t-1}^{\alpha} l_t^{1-\alpha}, \tag{19}$$

$$c_t = y_t + (1-\delta) k_{t-1} - k_t, \tag{20}$$

$$v_t = \vartheta \Pi_t^{\epsilon} v_{t-1} + (1-\vartheta)(\Pi_t^*)^{-\epsilon}. \tag{21}$$

Approximating other variables. Given the neural networks $\Psi_\theta(\cdot)$ and the state s, we define the following variables based on the equilibrium conditions above:

1. **Consumption:** Using equation (8), define:

$$c(s; \Psi_\theta) \equiv \frac{1}{\lambda_\theta(s)}.$$

2. **Wage:** Using equation (9), define:

$$w(s; \Psi_\theta) \equiv \frac{\psi l_\theta(s)^\eta}{\lambda_\theta(s)}.$$

3. **Expected discounted revenue of intermediary firms:** Using equation (14), define:

$$g^{(2)}(s; \Psi_\theta) \equiv \frac{\varepsilon}{\varepsilon - 1} g_\theta^{(1)}(s).$$

4. **Rental price of capital:** Using equation (15) and the wage function $w(s;\Psi_\theta)$, define:

$$r(s;\Psi_\theta) \equiv \frac{\alpha}{1-\alpha}\frac{l_\theta(s)}{k}w(s;\Psi_\theta).$$

5. **Marginal cost:** Using equation (16), along with the rental price of capital function $r(s;\Psi_\theta)$ and wage function $w(s;\Psi_\theta)$, define:

$$\text{mc}(s;\Psi_\theta) \equiv \left(\frac{1}{1-\alpha}\right)^{1-\alpha}\left(\frac{1}{\alpha}\right)^{\alpha}\frac{w(s;\Psi_\theta)^{1-\alpha}r(s;\Psi_\theta)^{\alpha}}{e^{z}}.$$

6. **Relative optimal price:** Using equation (17), define:

$$\Pi^*(s;\Psi_\theta) \equiv \left(\frac{1-\vartheta\Pi_\theta(s)^{\varepsilon-1}}{1-\vartheta}\right)^{\frac{1}{1-\varepsilon}}.$$

7. **Production:** Using $\Pi^*(s;\Psi_\theta)$ and equation (21) to compute v', and, then, using equation (19), define:

$$y(s;\Psi_\theta) \equiv \frac{e^{z}}{v'(s;\Psi_\theta)}k^{\alpha}l_\theta(s)^{1-\alpha}.$$

8. **Gross interest rate:** Using equation (18) and the production function $y(s;\Psi_\theta)$, define:

$$R(s;\Psi_\theta) \equiv R\left(\frac{\Pi_\theta(s)}{\Pi}\right)^{\gamma_\Pi}\left(\frac{y(s;\Psi_\theta)}{y}\right)^{\gamma_y}e^{\sigma_m\epsilon_{m,t}}.$$

The five neural networks in $\Psi_\theta(s)$ and the eight variables constructed using the equilibrium conditions together provide a parametric approximation for solving the complete equilibrium. Using $k'_\theta(s)$, $v'(s;\Psi_\theta)$, and the law of motion for technology, we construct the next-period state variables as:

$$s' = \left[\frac{k'-k^*}{k^*}, \frac{v'-v^*}{v^*}, \frac{z'}{\zeta_z\sigma_z}, \frac{\epsilon'_m}{\zeta_m\sigma_m}\right].$$

Architecture design. The choice of *Tanh* as activation function for our DNNs is motivated by its range of $[-1,1]$, which makes it convenient to interpret the DNN's output as percentage deviations from the DSS.

We make an exception for the final activation function for $\text{NN}_{g^{(1)}}(s;\theta_{g^{(1)}})$. During training, especially away from equilibrium paths, the output of $\text{NN}_{g^{(1)}}(s;\theta_{g^{(1)}})$ may exceed 100 percent above the DSS. Since *Tanh* saturates near its boundaries, this causes vanishing gradients and training instability. To avoid this problem, we use a linear activation in the last layer of $\text{NN}_{g^{(1)}}(s;\theta_{g^{(1)}})$.

4 Long-Run Simulations

As we mentioned in the main text, the "simple" solutions we have discussed only approximately satisfy the long-run boundary conditions. This can create problems when simulating ergodic distributions or computing steady states in some models.[4]

There are at least three strategies to address this issue. First, and most important, one can carefully tune the hyperparameters of the DNN. Even if the loss function is already sufficiently small, a better-trained DNN can provide an extra layer of precision for computing ergodic distributions.

Second, the training time horizon can be extended, using the accurate medium-run solution as the initial condition. The inductive bias of the deep learning approach then yields accurate solutions and stable convergence in the long run.

Third, the algorithm can be retrained with simulated data from a previously trained DNN, and the new approximation used to refine the economy's trajectory. This process can be repeated, with the inductive bias of the deep learning solution ensuring accurate and stable long-run results.

A skillful combination of these three strategies can provide robust solutions across a large class of models.

[4] This is not just a problem for our approach. Computing accurate ergodic distributions is a notoriously challenging task for all solution methods.