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 $\mathcal{D} \equiv \{0, \ldots, 29\}$ 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.

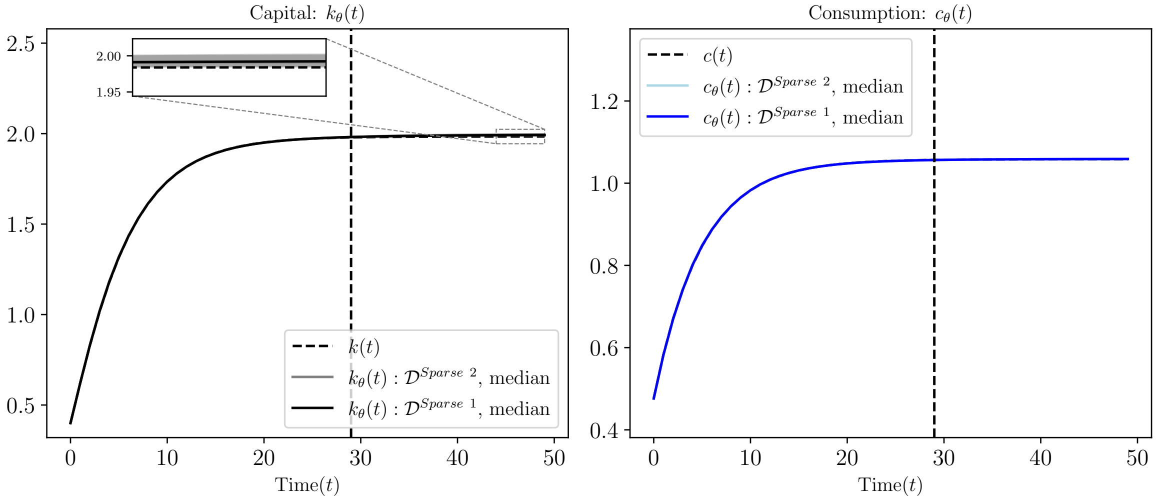

Figure 1.1: Solutions to equation (12) of the main text with $\mathcal{D}^{\text{Sparse 1}}$ and $\mathcal{D}^{\text{Sparse 2}}$.

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.

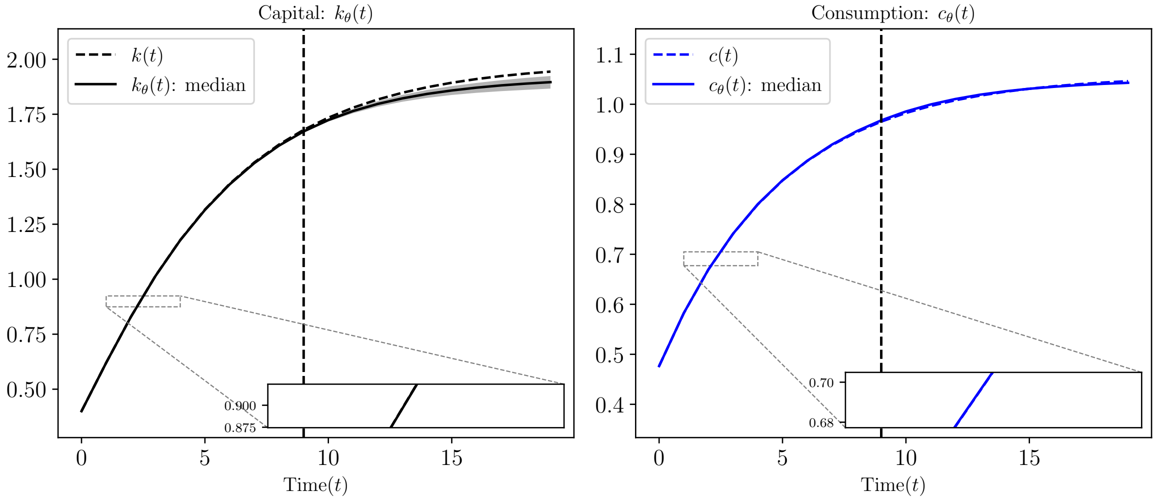

Figure 1.2: Solutions to equation (12) with $\mathcal{D} \equiv \{0, 1, \cdots, 9\}$.

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.

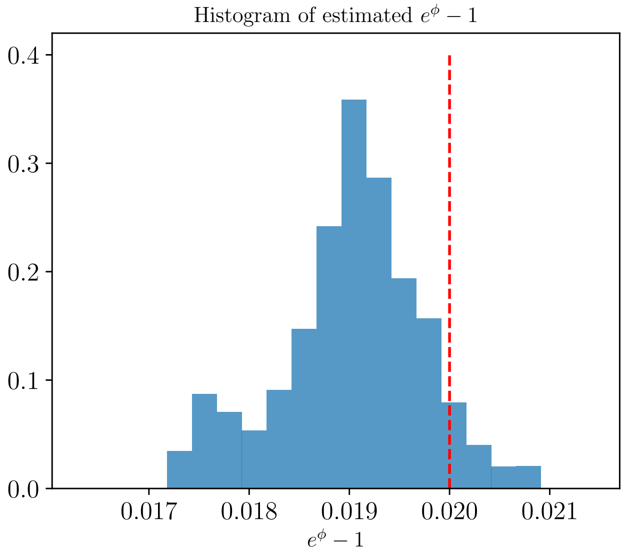

Figure 1.3: The distribution of the learned $e^{\phi} - 1$ for the ensemble of 100 seeds used in solving equation (12); $g = 0.02$, shown as the dashed line.

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.

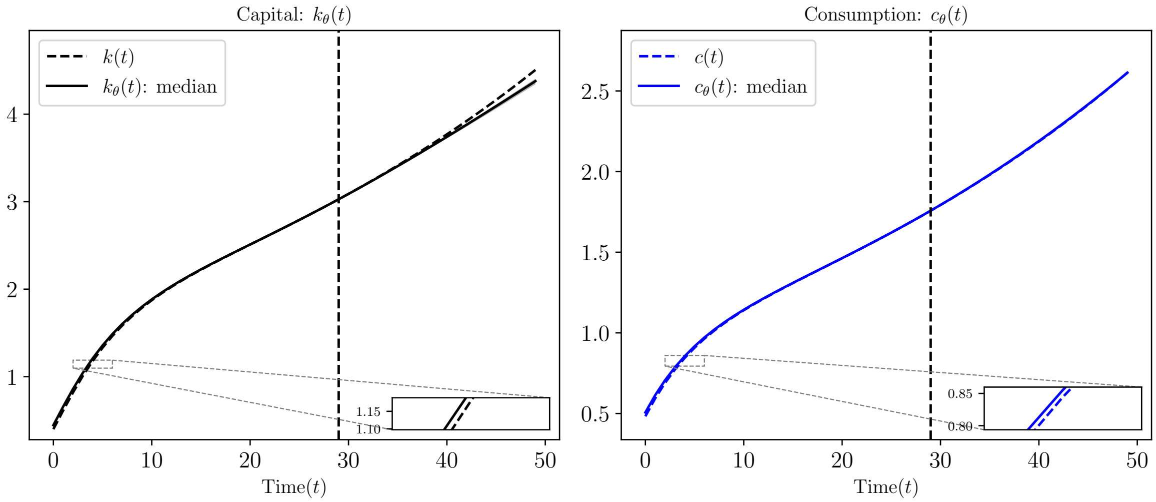

Figure 1.4: Solutions to equation (12) with the misspecified $k_{\theta}(t) = t \cdot \mathrm{NN}(t;\theta) + k_0$ and $g = 0.02$.

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.

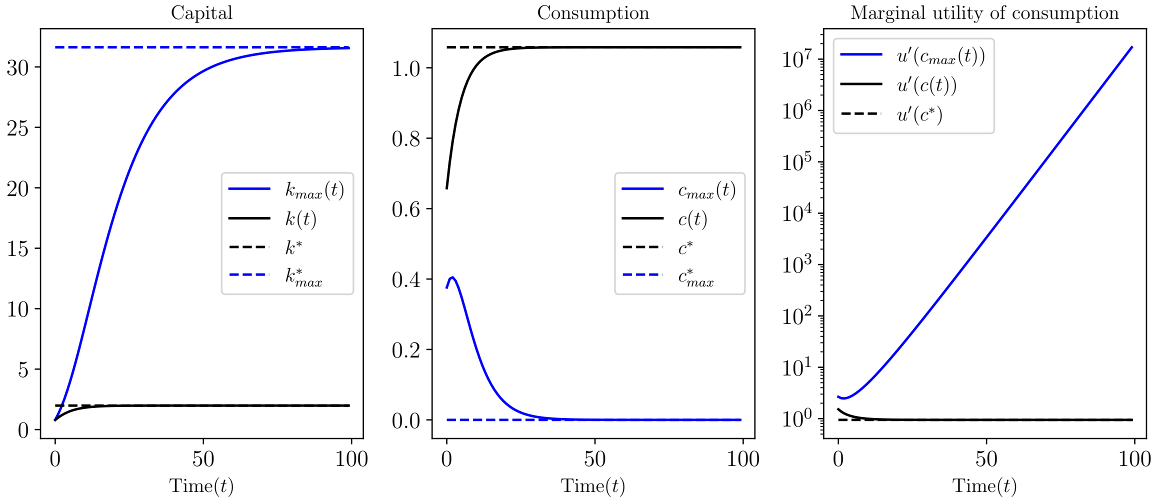

Figure 1.5: Comparison between the optimal solution and those violating the transversality condition.

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:

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

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

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.

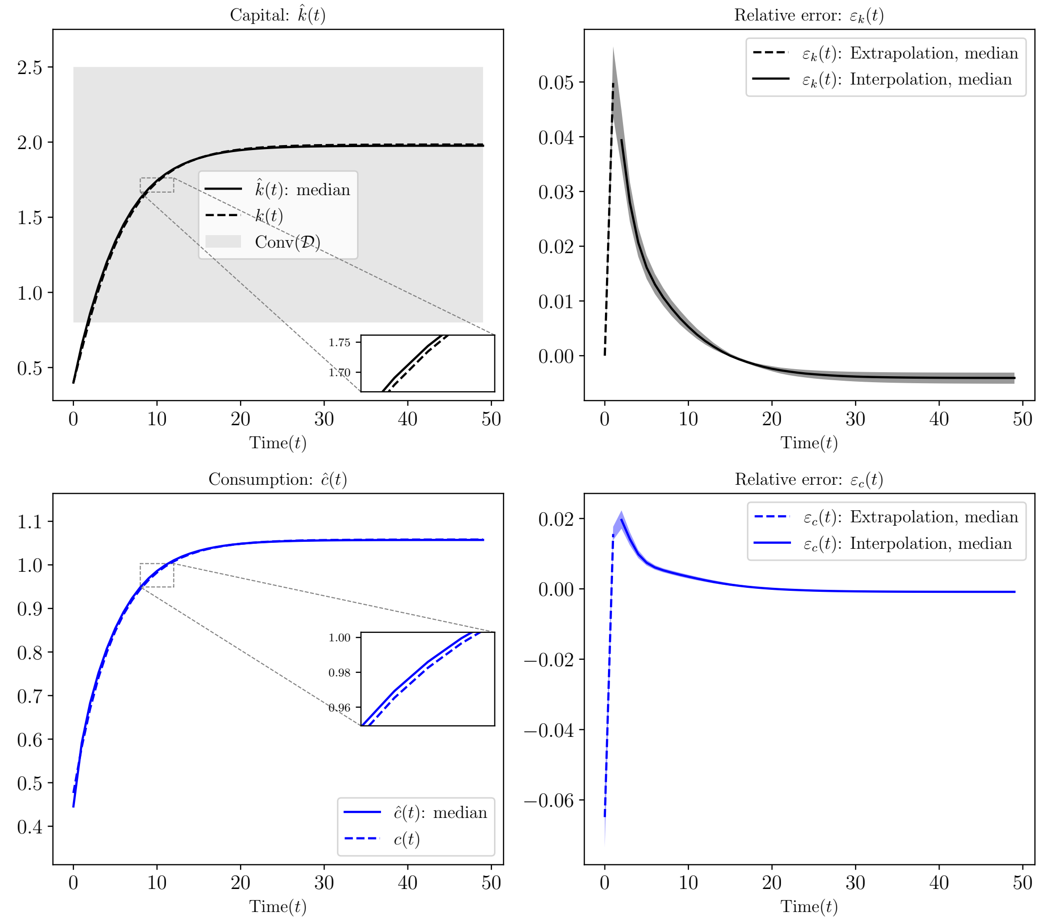

Figure 2.1: Solutions obtained by solving problem (4) for $g = 0$.

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 $\mathrm{Conv}(\mathcal{D})$.

BGP. Since the solution is homothetic when $g = 0.02$, we design our DNN as $k'_\theta(k, z) \equiv z \cdot \mathrm{NN}(k/z, z; \theta)$. 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 $\mathrm{NN}(k/z; \theta)$, but we retain the z parameter to handle nearly homothetic cases and to verify that the inductive bias prevents overfitting.

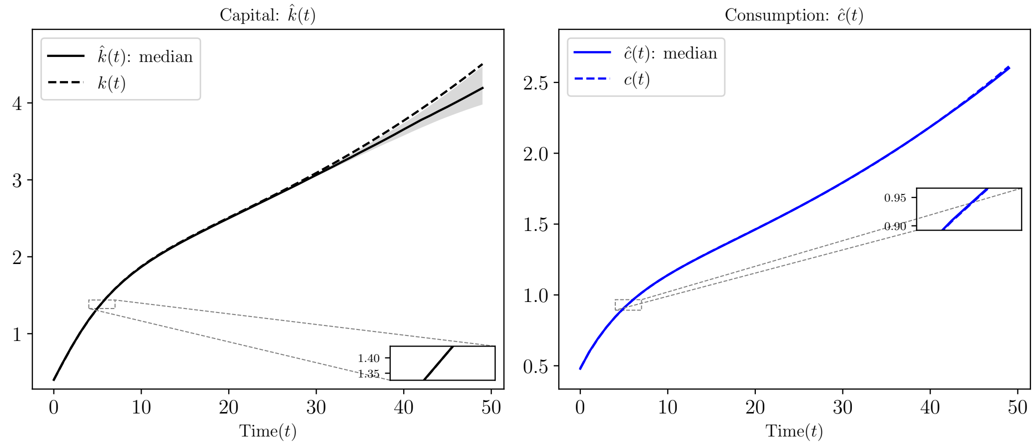

Figure 2.2: Solutions to problem 4 for $g = 0.02$.

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 $k^* \approx 2.0$.

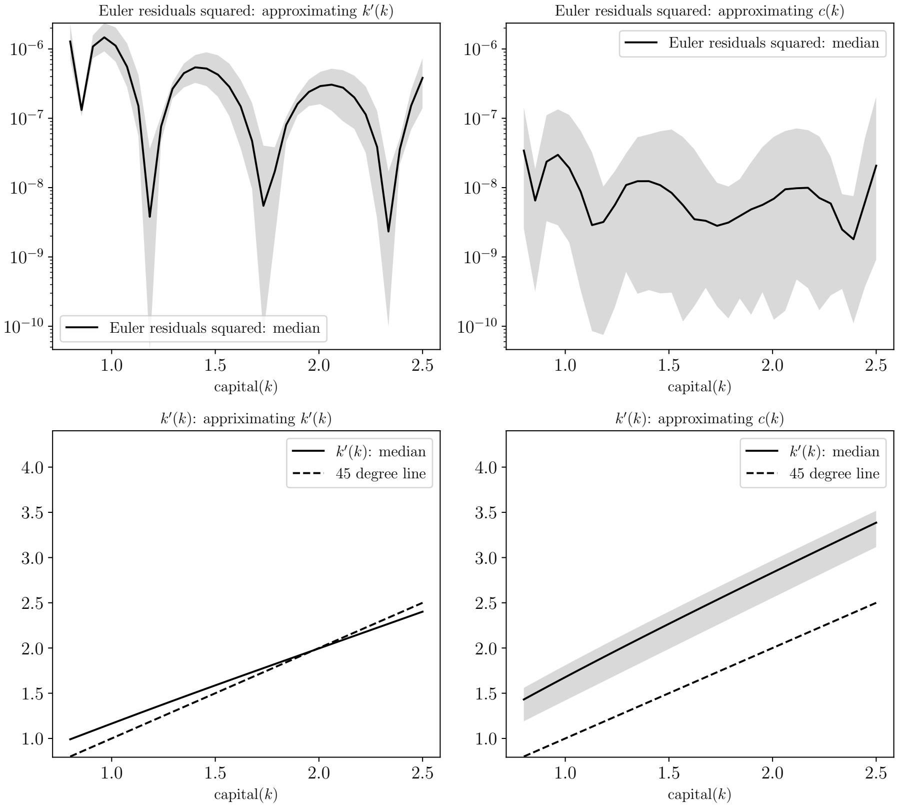

Figure 2.3: Comparison between approximating the policy function for capital $k'(k)$ vs. the consumption function $c(k)$ with a DNN.

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:

$$
\begin{aligned}
g_t^{(1)} &= \lambda_t \mathrm{mc}_t y_t + \beta\vartheta\mathbb{E}_t\left[\Pi_{t+1}^{\varepsilon} g_{t+1}^{(1)}\right], && (12)\\
g_t^{(2)} &= \lambda_t \Pi_t^* y_t + \beta\vartheta\mathbb{E}_t\left[\Pi_{t+1}^{\varepsilon-1}\left(\frac{\Pi_t^*}{\Pi_{t+1}^*}\right) g_{t+1}^{(2)}\right], && (13)\\
\varepsilon g_t^{(1)} &= (\varepsilon-1) g_t^{(2)}, && (14)\\
\frac{k_{t-1}}{l_t} &= \frac{\alpha}{1-\alpha}\frac{w_t}{r_t}, && (15)\\
\mathrm{mc}_t &= \left(\frac{1}{1-\alpha}\right)^{1-\alpha}\left(\frac{1}{\alpha}\right)^{\alpha}\frac{w_t^{1-\alpha} r_t^{\alpha}}{e^{z_t}}. && (16)
\end{aligned}
$$

- 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:

$$
\begin{aligned}
y_t &= \frac{1}{v_t} e^{z_t} k_{t-1}^{\alpha} l_t^{1-\alpha}, && (19)\\
c_t &= y_t + (1-\delta)k_{t-1} - k_t, && (20)\\
v_t &= \vartheta\Pi_t^{\epsilon} v_{t-1} + (1-\vartheta)(\Pi_t^*)^{-\epsilon}. && (21)
\end{aligned}
$$

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.