The Blessings of Overparameterization: Applications in Solving Economic Models

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

[1]Bowdoin College

[2]University of Pennsylvania

International Symposium on Nonparametric Statistics (ISNPS 2026)
Thessaloniki, Greece June 22–26, 2026

Motivation, Question, and Contribution

Motivation

"I remember my friend Johnny von Neumann used to say, with four parameters I can fit an elephant, and with five I can make him wiggle his trunk."—*Enrico Fermi*

- Deep learning has proved remarkably capable at solving **high-dimensional problems**.
- Economists are now using it to find accurate approximate solutions to dynamic models once thought intractable.
- These neural networks are **massively overparameterized**, with far more parameters than "data" points.
- Standard statistical wisdom says: more parameters than "data" points leads to **overfitting** and poor out-of-grid / out-of-distribution performance.

Question

When solving dynamic models with neural networks:

- Does overparameterization lead to poor approximation in out-of-grid / out-of-distribution performance?
- Is overparameterization a "necessary evil" that comes with modern machine learning methods, or does it give us something extra?

Contribution

1. **No overfitting:** as the number of network parameters grows, the solutions become more accurate.
2. **Algorithmic stability:** the approximate solutions of dynamic models in economics are unstable; overparameterization increases stability.

Implication

When in doubt, use a bigger network. Overparameterization is not a source of risk to be managed but a feature to be exploited.

Background: Economic Models, Approximate Solutions, and Overparameterization

An economic model is a system of functional equations

Many theoretical models can be written as functional equations:

- Economic object of interest: f, where $f : \mathcal{X} \to \mathcal{R} \subseteq \mathbb{R}^N$
  - e.g., entry/exit decision, investment choice, best-response, etc.
- Domain of f: $\mathcal{X}$
  - e.g., market conditions, space of capital, opponents' state, etc.
- The "Economics model": $\ell\big(x, f\big)$
  - e.g., Euler and Bellman residuals, equilibrium FOCs.

Then a **solution** is $f^* \in \mathcal{F}$ where $\ell(x, f^*) = \mathbf{0}$ for all $x \in \mathcal{X}$.

Approximate solution

1. Sample $\mathcal{X}$: $\mathcal{D} = \{x_1, \cdots, x_N\}$
2. Pick a family of parametric functions (e.g., polynomials, Chebyshev polynomials, neural networks) $f(\cdot; \theta)$:
   - $\theta = \{\theta_1, \cdots, \theta_M\}$: parameters for optimization (e.g., coefficients, or weights and biases).
3. To find an approximation for f solve:

$$\min_{\theta} \frac{1}{N} \sum_{x \in \mathcal{D}} \| \underbrace{\ell\left(x, f(\cdot; \theta)\right)}_{\text{Econ model error}} \|_2^2$$

Neural Networks and Overparameterization

Neural Networks form a **highly-overparameterized** ($M \gg N$) class of functions.

- Example: one layer neural network, $f(\cdot;\theta) : \mathbb{R}^Q \to \mathbb{R}$:

$$f(x;\theta) = W_2 \cdot \sigma\left(W_1 \cdot x + b_1\right) + b_2$$

- $W_1 \in \mathbb{R}^{P\times Q}$, $b_1 \in \mathbb{R}^{P\times 1}$, $W_2 \in \mathbb{R}^{1\times P}$, and $b_2 \in \mathbb{R}$.
- $\theta \equiv \{b_1, W_1, b_2, W_2\}$ are the coefficients, in this example $M = PQ + P + P + 1$.
- $\sigma(\cdot)$ is a nonlinear function applied element-wise (e.g., $\max\{\cdot, 0\}$).
- Making it "deeper" by adding another "layer": $f(x;\theta) \equiv W_3 \cdot \sigma(W_2 \cdot \sigma(W_1 \cdot x + b_1) + b_2) + b_3$.

Deep learning is a **non-convex** optimization

- Training a neural network is **non-convex** in θ on its own (even with linear neural networks).
- On top of that, most economic dynamic models are **nonlinear**.
- In a non-convex problem, the solution depends on the **initialization** of the weights and biases θ.
- So the "answer" can depend on the initialization.
- We run the algorithm for 50 different initializations (seeds) of the weights and biases.

Applications

Experimental design

- We demonstrate these properties across three canonical models with known benchmarks:

1. Linear-Quadratic

Closed-form solution. Sharpest test of approximation error.

2. McCall Job Search

1D state, kink at reservation wage. Non-smooth approximation.

3. Real Business Cycle

2D stochastic state. No closed form. Benchmarked vs. VFI.

- In all three: **accuracy** and **stability** improve with overparameterization.

Application 1: Linear-Quadratic regulator

The LQ problem

- The general deterministic LQ problem:

$$v(x) = \max_{u} \left\{ -x^\top R\, x - u^\top Q\, u + \beta\, v(x') \right\}$$
$$\text{s.t.} \quad x' = Ax + Bu, \quad x_0 \text{ given}$$

- State variable: $x \in \mathbb{R}^n$. Control variable: $u \in \mathbb{R}^m$.
- Closed-form solution:

$$v(x) = -x^\top P x, \qquad u(x) = -Fx$$

  where P and F are obtained from the discrete-time algebraic Riccati equation.
- Sharpest test: approximation error can be computed exactly at every point.

A competitive firm with adjustment costs

- A price-taking firm in a competitive industry.
- Own output y, aggregate output Y (exogenous). Inverse demand: $\alpha_0 - \alpha_1 Y$.
- The firm chooses investment u subject to quadratic adjustment cost $\frac{\gamma}{2}u^2$:

$$v(y, Y) = \max_u \left\{ (\alpha_0 - \alpha_1 Y) y - \tfrac{\gamma}{2} u^2 + \beta\, v(y', Y') \right\}$$
$$\text{s.t.} \quad Y' = h_0 + h_1 Y, \quad y' = y + u$$

- State is two-dimensional: (y, Y). With augmented state $x = (1, y, Y)^\top$, this maps into the general LQ framework.
- Euler equation:

$$\gamma\, u(y, Y) = \beta \left[\gamma\, u(y', Y') + (\alpha_0 - \alpha_1 Y') \right]$$

LQ: neural network implementation

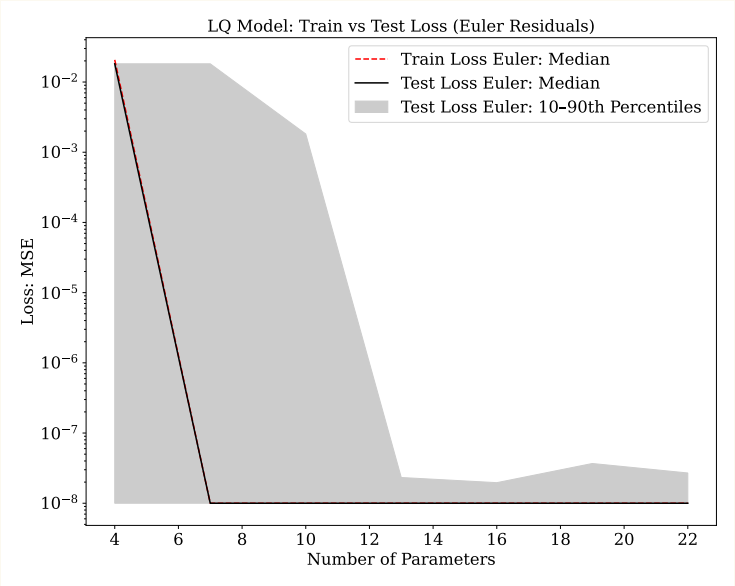
- **Architecture:** one-hidden-layer ReLU networks for value and policy functions. Hidden units $H = 1$ to $H = 7$ (4 to 22 parameters).
- **Training data:** the optimal u depends only on Y (y enters linearly and does not affect the marginal return to investment), so we minimize the mean squared Euler residual over $n_Y = 6$ grid points:

$$\min_{\theta_u} \frac{1}{n_Y} \sum_{i=1}^{n_Y} \left[\gamma\, u(Y_i;\theta_u) - \beta\big(\gamma\, u(Y_i';\theta_u) + \alpha_0 - \alpha_1 Y_i'\big)\right]^2$$

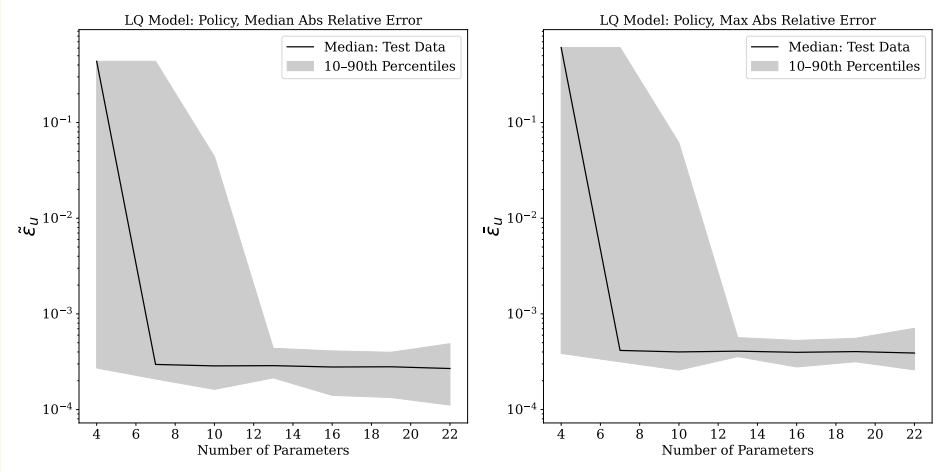
- Adam optimizer with early stopping at loss $< 10^{-8}$.
- **Test data:** $T = 29$ period path simulated from fixed initial conditions. Test points lie in the interior of the state space and never on a training node.

Parameterization

LQ: train and test Euler-residual loss

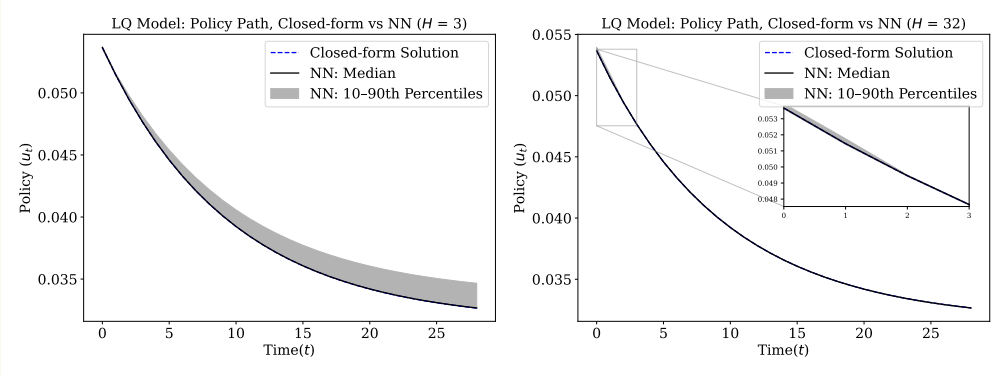


LQ: absolute relative error vs. closed-form solution



- Median (left) and maximum (right) absolute relative error of the NN policy. Both statistics and their cross-seed dispersion fall with network width.

LQ: policy paths, underparameterized vs. overparameterized



- Left: $H = 3$ (underparameterized). Median is roughly accurate, but the across-seed band is wide.
- Right: $H = 32$ (overparameterized). Median tracks the closed-form solution closely and the band collapses.

Application 2: McCall job-search model

From a linear to a non-smooth operator equation

- The LQ model is linear, which makes it the cleanest test but also the easiest.
- Does the blessing survive when the solution is nonlinear and non-smooth?
- The McCall model (McCall, 1970) provides exactly this test: the value function has a **kink** at the reservation wage, so the network must approximate a non-differentiable function.
- An unemployed worker receives i.i.d. wage offers $w \sim f$ on $[0, B]$ each period.
- Accept: permanent income stream $w/(1-\beta)$. Reject: unemployment compensation c plus continuation value.
- Bellman equation (a nonlinear integral equation with a max operator):

$$v(w) = \max\left\{\frac{w}{1-\beta},\; c + \beta \int_0^B v(w')f(w')\,dw'\right\}$$

McCall: the kink and why it matters

- Optimal policy: threshold form. There exists a reservation wage $\bar{w}$ such that accept iff $w > \bar{w}$.
- Closed-form value function:

$$v(w) = \begin{cases} \bar{w}/(1-\beta) & \text{if } w \le \bar{w}, \\ w/(1-\beta) & \text{if } w > \bar{w}. \end{cases}$$

- The reservation wage $\bar{w}$ solves:

$$\bar{w} - c = \frac{\beta}{1-\beta} \int_{\bar{w}}^{B} (w' - \bar{w}) f(w')\, dw'$$

  LHS: net gain from rejecting the marginal offer. RHS: expected gain from continued search.
- The kink at $\bar{w}$ is a demanding test: the network must learn a non-smooth function from only $N = 12$ training points.

McCall: neural network implementation

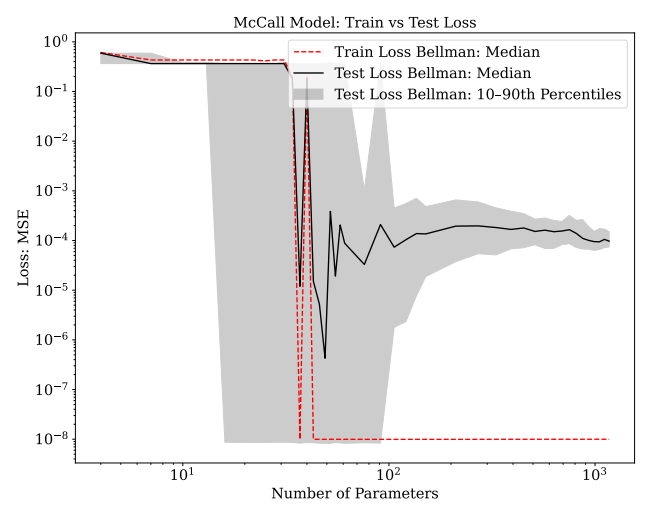
- **Architecture:** one-hidden-layer ReLU network $v(w;\theta_v)$, linear output. 50 random seeds per width.
- **Training:** $N = 12$ uniform grid on $[0, 1]$. Minimize mean squared Bellman residual:

$$\min_{\theta_v} \frac{1}{N} \sum_{w_i \in \mathcal{D}} \left[v(w_i;\theta_v) - \max\left\{ \frac{w_i}{1-\beta},\ c + \beta \int_0^B v(w';\theta_v) f(w')\, dw' \right\} \right]^2$$

  Integral by quadrature. Adam with early stopping at 10^{-8}.
- **Test:** finer $N = 100$ uniform grid, strictly outside the training set.
- **Accuracy metric:** absolute relative error $\varepsilon_v(w) = |v(w;\theta_v^*) - v(w)|/|v(w)|$. Summarized by max and median across test grid, then 10th/50th/90th percentiles across 50 seeds.

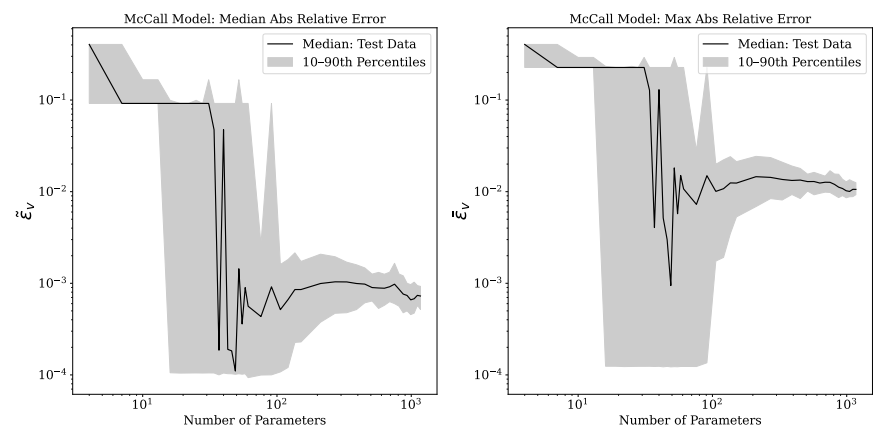
⏩ Parameterization

McCall: train and test Bellman-residual loss



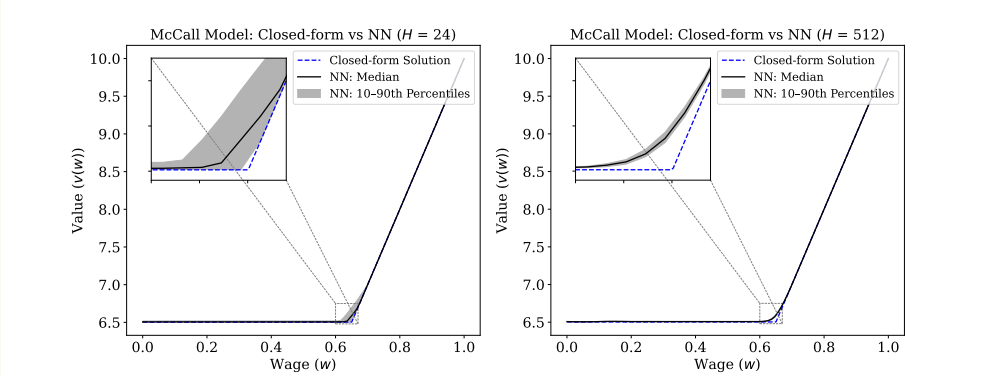
- Test loss descends again in the overparameterized regime: the **double-descent pattern**.
- The across-seed band collapses.

McCall: absolute relative error



- Median (left) and max (right) relative error fall from $\sim$30% to $\sim$2% as the network enters the overparameterized regime.
- The across-seed band narrows sharply: **stability blessing confirmed**.

McCall: value function, underparameterized vs. overparameterized



- Left: $H = 24$ (underparameterized). Median tracks the closed-form solution well, but the across-seed band is wide.
- Right: $H = 512$ (overparameterized). Band is nearly invisible: 50 random initializations converge to the same function.

Application 3: Real Business Cycle model

From deterministic to stochastic, from closed form to none

- The LQ model is linear with a closed-form solution. The McCall model is nonlinear with a kink but still one-dimensional and still admits a closed form.
- The Real Business Cycle model (Kydland and Prescott, 1982) is the most demanding test: a two-dimensional stochastic state space (k_t, z_t) and no analytical solution.
- Social planner, Cobb–Douglas production $e^{z_t} k_t^{\alpha}$, stochastic TFP:

$$\max_{\{c_t, k_{t+1}\}} \mathbb{E}_0 \sum_{t=0}^{\infty} \beta^t u(c_t) \quad \text{s.t.} \quad c_t + k_{t+1} = e^{z_t} k_t^{\alpha} + (1-\delta) k_t$$

$$z_t = \rho z_{t-1} + \sigma \varepsilon_t, \qquad \varepsilon_t \sim \mathcal{N}(0,1)$$

- Key object: policy function $k'(k, z)$. Benchmark: value function iteration (VFI) on a fine grid.

RBC: the Euler equation

- Euler equation:

$$u'(c_t) = \beta\, \mathbb{E}_t\left[u'(c_{t+1})\left(\alpha e^{z_{t+1}} k_{t+1}^{\alpha-1} + 1 - \delta\right)\right]$$

- **LHS**: marginal utility cost of saving one unit today.
- **RHS**: expected discounted marginal utility gain from extra capital tomorrow (marginal product plus undepreciated capital).
- Nonlinearity plus stochastic two-dimensional state means no analytical solution.

RBC: neural network implementation

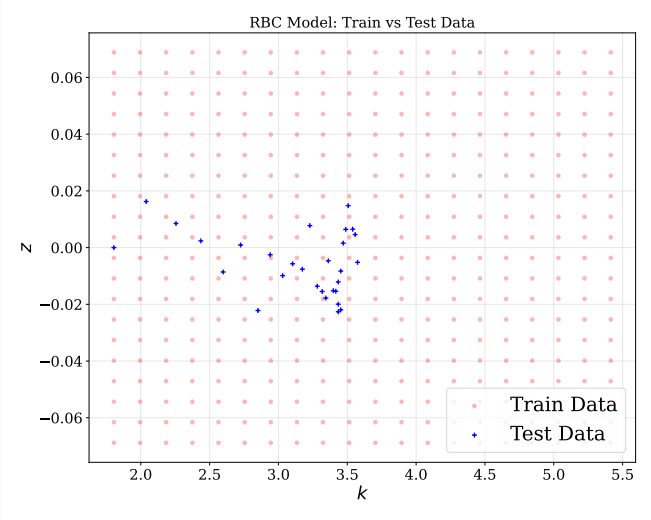
- **Architecture:** one-hidden-layer ReLU network $k'(k, z; \theta)$. Output layer: Softplus activation $\log(1 + e^x)$ to enforce $k' > 0$. 50 seeds per width.
- **Training:** 20×20 Cartesian grid ($N = 400$). Euler residual:

$$\ell_{\text{Euler}}(k, z; \theta) = \frac{1}{c(k, z; \theta)} - \beta \sum_{i=1}^{J} \frac{w_i}{\sqrt{\pi}} \left[\frac{\alpha e^{\rho z + \sqrt{2}\sigma\zeta_i} k'(k, z; \theta)^{\alpha - 1} + 1 - \delta}{c\big(k'(k, z; \theta),\ \rho z + \sqrt{2}\sigma\zeta_i;\ \theta\big)} \right]$$

  with $J = 15$ Gauss–Hermite nodes. Adam with StepLR scheduler.
- **Penalty term:** $\lambda\big(k'(k^*, 0; \theta) - k^*\big)^2$ with $\lambda = 0.01$, enforcing a necessary implication of the transversality condition at the steady state.

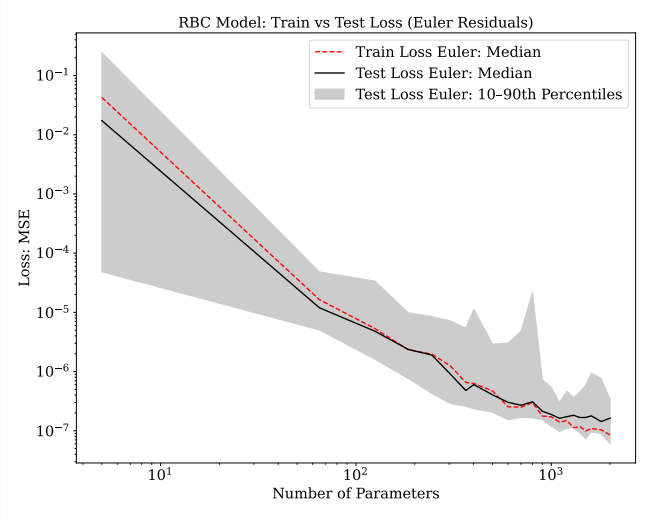
⏩ Parameterization

RBC: training and test data



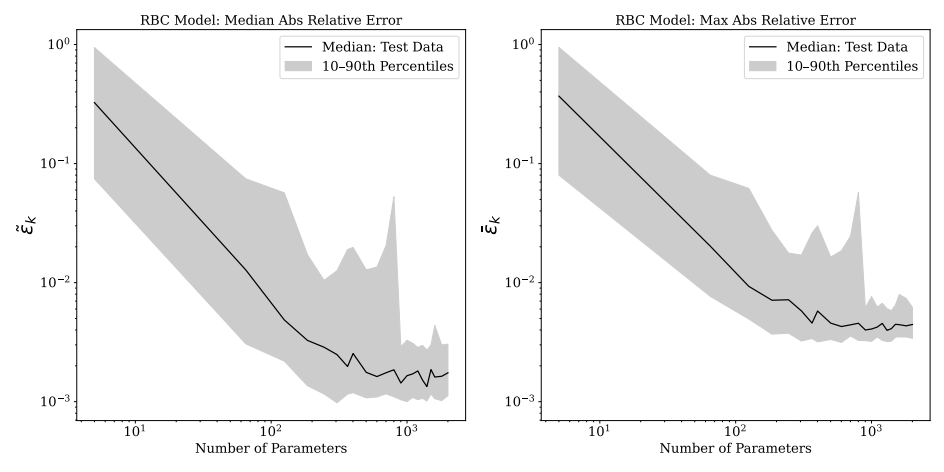
- Red dots: 20×20 training grid.
- Blue crosses: $T = 29$ path simulated under the VFI policy starting at $k_0 = 0.5\,k^*$.
- Test path visits the interior of the state space and is never on a training node.

RBC: train and test Euler-residual loss



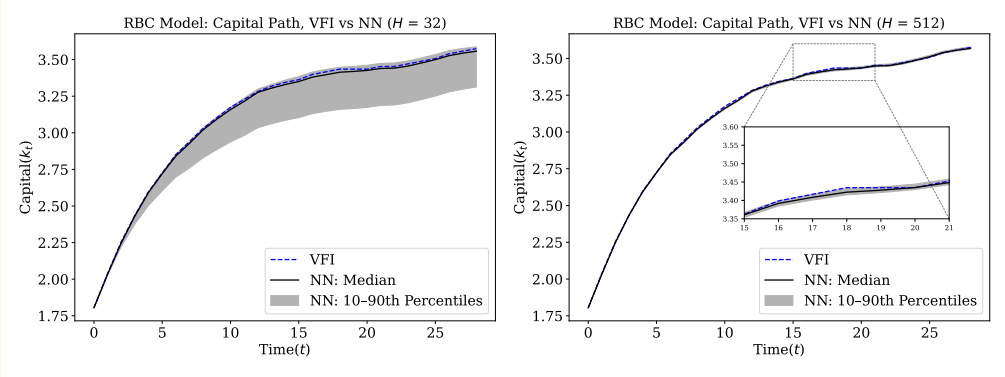
- Both losses decline as the network grows wider, with no **sign of overfitting**.
- The across-seed band narrows sharply: overparameterization simultaneously improves accuracy and reduces initialization sensitivity.

RBC: absolute relative error vs. VFI



- Median (left) and maximum (right) absolute relative error of the NN capital policy versus VFI.
- Both statistics decline and concentrate around the median seed as width increases.

RBC: capital paths, underparameterized vs. overparameterized



- Left: $H = 32$. Median path close to VFI, but across-seed band is wide. Accurate on average but unreliable.
- Right: $H = 512$. Median tracks VFI almost exactly over the full horizon and the band collapses to near zero.

Robustness

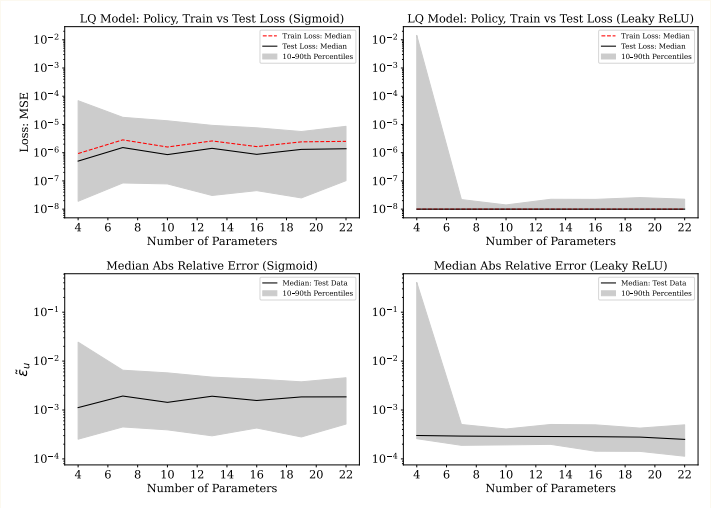
Are these results specific to ReLU and one hidden layer?

- A natural concern: the blessing might depend on the particular activation function or the single-layer architecture.
- We test robustness along two dimensions:
  - **Activation functions:** Sigmoid and Leaky ReLU in addition to the baseline ReLU.
  - **Network depth:** two-hidden-layer architectures in addition to the baseline one-hidden-layer.

Result

The blessing holds in every configuration we tried. The next slides document this for each model.

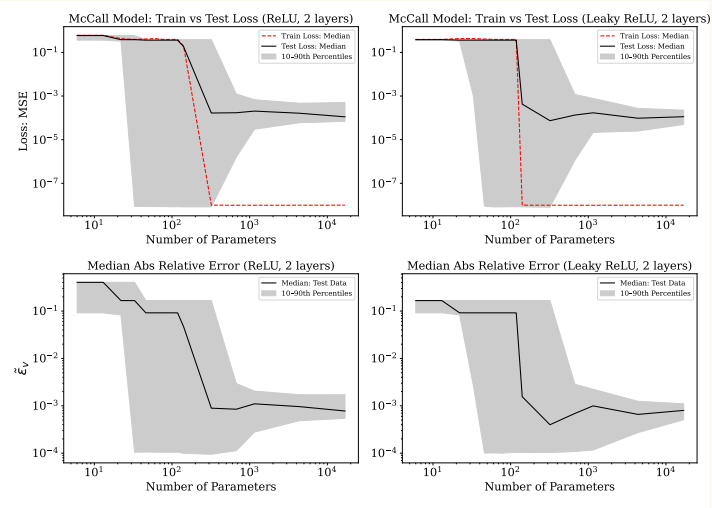
Robustness: alternative activations for the LQ model



- Sigmoid (left) and Leaky ReLU (right).
- Top row: train/test Euler MSE. Bottom row: median relative error.
- Both activations replicate the main finding: accuracy improves with width, across-seed dispersion collapses.

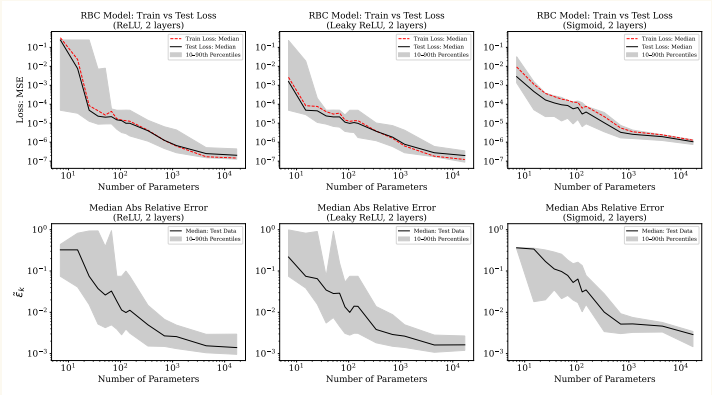
⏩ Activation functions

Robustness: deeper networks for the McCall model



- Two-hidden-layer architecture with ReLU (left) and Leaky ReLU (right).
- Quantitatively similar results: the blessing is not specific to one-hidden-layer networks.

Robustness: deeper networks for the RBC model



- Two-hidden-layer architecture with ReLU (left), Leaky ReLU (center), Sigmoid (right).
- All three activations replicate the main finding for a stochastic model with no closed-form solution.

Robustness: summary

- The blessing of overparameterization is robust across all dimensions most relevant to applied practice:
  - **Activation functions:** ReLU, Leaky ReLU, Sigmoid.
  - **Network depth:** one and two hidden layers.
  - **Operator type:** Euler residual (LQ, RBC), Bellman residual (McCall).
  - **Solution regularity:** smooth (LQ), non-smooth with kink (McCall), stochastic without closed form (RBC).
  - **State dimensionality:** 1D (McCall), 2D (LQ, RBC).

Conclusion

Conclusion

- **Finding 1:** no overfitting. As network width increases, out-of-sample Euler and Bellman residuals decline. Value and policy functions converge toward their benchmarks.
- **Finding 2:** algorithmic stability. Small networks produce solutions that depend on the **initialization**. Wide networks concentrate across seeds, making the algorithm reliable and **initialization-independent**.
- Documented consistently across three operator equations: linear with closed form (LQ), nonlinear with kink (McCall), stochastic with no analytical solution (RBC).
- Robust across activations (ReLU, Leaky ReLU, Sigmoid) and depths (one and two hidden layers).

Bottom line

When in doubt, use a wider network. Overparameterization is a blessing, not a curse.

Appendix

Parameterization: Linear-Quadratic model

Description	Symbol	Value	
Discount factor	β	0.9	
Demand intercept	α_0	1.0	inverse demand level
Demand slope	α_1	1.3	price sensitivity to aggregate
Adjustment cost	γ	100	convex investment cost
Aggregate drift	h_0	0.05	constant in $Y' = h_0 + h_1 Y$
Aggregate persistence	h_1	0.9	AR coefficient for Y

Back

Parameterization: McCall model

Description	Symbol	Value	
Discount factor	β	0.9	
Wage support	B	1.0	upper bound of $\mathrm{U}[0, B]$
Unemployment benefit	c	0.1	flow payoff while searching

Back

Parameterization: Real Business Cycle model

Description	Symbol	Value	
Discount factor	β	0.96	
Capital share	α	$1/3$	Cobb–Douglas exponent
Depreciation rate	δ	0.1	per-period capital depreciation
TFP persistence	ρ	0.9	AR(1) coefficient for z_t
TFP volatility	σ	0.01	std. dev. of ε_t
Utility function	$u(c)$	$\ln c$	log utility

Back

Activation functions

For a scalar pre-activation $z \in \mathbb{R}$, applied element-wise:

- **ReLU**:

$$\sigma(z) = \max\{0, z\} = \begin{cases} z & \text{if } z \geq 0, \\ 0 & \text{if } z < 0. \end{cases}$$

- **Leaky ReLU** (slope $\alpha = 0.01$):

$$\sigma(z) = \max\{\alpha z, z\} = \begin{cases} z & \text{if } z \geq 0, \\ \alpha z & \text{if } z < 0. \end{cases}$$

- **Sigmoid**:

$$\sigma(z) = \frac{1}{1 + e^{-z}}.$$