Global Estimation of Nonlinear DSGE Models with Neural Network Surrogates

A switching-fidelity estimator:
SEP offline, ROM1 online, posterior-guided refinement.

Mátyás Farkas
International Monetary Fund

Society for Computational Economics (CEF)

Views expressed are my own and do not necessarily represent the IMF.

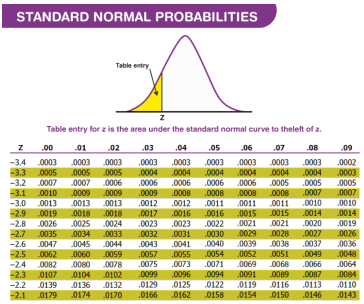
The bottleneck is the nonlinear solve inside the posterior loop

- Linear estimation is fast: the transition map is cheap.
- Nonlinear estimation is hard: the global solve sits *inside* the posterior sampler.
- What we need is not a new model. It is a **fast likelihood for the same model**.

$$s_{t+1} = g_{\theta, s_t, \varepsilon_{t+1}},$$
$$y_t = h(s_t; \theta) + \eta_t.$$

This paper:
Estimate **nonlinear DSGE models globally** with neural-network surrogates that move the nonlinear solve **offline**, use a switching-fidelity estimator **online**, and refine support where the posterior actually goes.

An old idea: precompute the hard function, then look it up



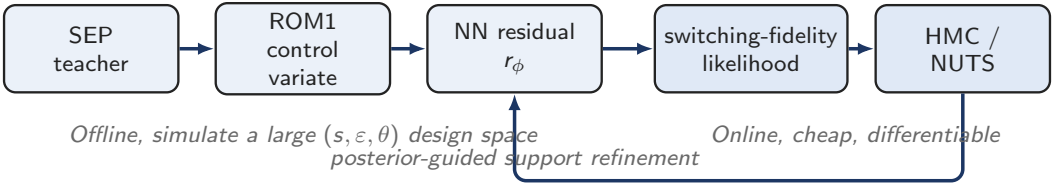
STANDARD NORMAL PROBABILITIES

Table entry for z is the area under the standard normal curve to theleft of z.

z	.00	.01	.02	.03	.04	.05	.06	.07	.08	.09
−3.4	.0003	.0003	.0003	.0003	.0003	.0003	.0003	.0003	.0003	.0002
−3.3	.0005	.0005	.0005	.0004	.0004	.0004	.0004	.0004	.0004	.0003
−3.2	.0007	.0007	.0006	.0006	.0006	.0006	.0006	.0005	.0005	.0005
−3.1	.0010	.0009	.0009	.0009	.0008	.0008	.0008	.0008	.0007	.0007
−3.0	.0013	.0013	.0013	.0012	.0012	.0011	.0011	.0011	.0010	.0010
−2.9	.0019	.0018	.0018	.0017	.0016	.0016	.0015	.0015	.0014	.0014
−2.8	.0026	.0025	.0024	.0023	.0023	.0022	.0021	.0021	.0020	.0019
−2.7	.0035	.0034	.0033	.0032	.0031	.0030	.0029	.0028	.0027	.0026
−2.6	.0047	.0045	.0044	.0043	.0041	.0040	.0039	.0038	.0037	.0036
−2.5	.0062	.0060	.0059	.0057	.0055	.0054	.0052	.0051	.0049	.0048
−2.4	.0082	.0080	.0078	.0075	.0073	.0071	.0069	.0068	.0066	.0064
−2.3	.0107	.0104	.0102	.0099	.0096	.0094	.0091	.0089	.0087	.0084
−2.2	.0139	.0136	.0132	.0129	.0125	.0122	.0119	.0116	.0113	.0110
−2.1	.0179	.0174	.0170	.0166	.0162	.0158	.0154	.0150	.0146	.0143

The surrogate is a differentiable lookup table:
The inverse normal Φ^{-1}, the error function, was tabulated once and read off fast. SEP plays the same role: solve the nonlinear model **offline**, and let the network be a smooth, differentiable lookup table for the transition map.

Move the nonlinear solve offline; let the posterior refine support



Closed loop:
The nonlinear cost is paid offline, but not blindly: a preliminary posterior run shows where the NN is extrapolating; those points go back to SEP, become new residual labels, and the promoted run uses the retrained surrogate.

Five literatures meet in one pipeline

Nonlinear solution	Fair–Taylor (1983); Judd (1992); Christiano–Fisher (2000); Guerrieri–Iacoviello (2015); Adjemian–Juillard (2025)
Nonlinear likelihood	Fernández-Villaverde–Rubio-Ramírez (2007); Andrieu–Doucet–Holenstein (2010); Herbst–Schorfheide (2016); Gust et al. (2017); Cuba-Borda et al. (2019)
Neural surrogates	Scheidegger–Bilionis (2019); Maliar–Maliar–Winant (2021); Azinovic–Gaegauf–Scheidegger (2022); Kase–Melosi–Rottner (2025); Naubert (2025)
Model reduction	Kennedy–O'Hagan (2001); Aruoba–Fernández-Villaverde–Rubio-Ramírez (2006); Benner–Gugercin–Willcox (2015); Quarteroni–Manzoni–Negri (2016); Peherstorfer–Willcox–Gunzburger (2018)
Bayesian / HMC	**Farkas–Tatár (2020)**; Neal (2011); Hoffman–Gelman (2014); Betancourt (2017); Childers et al. (2022); Vehtari et al. (2021)

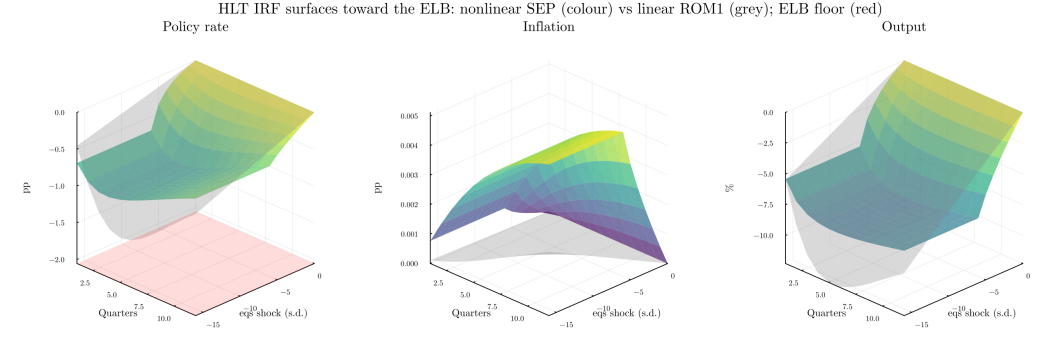
Supervised NN estimation:
The closest methods, Kase–Melosi–Rottner (2025) and Naubert (2025), train the network *unsupervised*, minimizing equilibrium residuals so the net itself becomes the solution. I train *supervised*: SEP solves the model offline to generate **labelled** $(s, \varepsilon, \theta) \to$ transition data, and the network regresses on the SEP–ROM1 residual.

Do not learn the solution, learn the gap to a cheap model

$$\underbrace{g^{\mathrm{SEP}}_{\theta,s_t,\varepsilon_{t+1}}}_{\text{full-order (truth)}} = \underbrace{g^{\mathrm{ROM1}}_{\theta,s_t,\varepsilon_{t+1}}}_{\text{reduced-order (cheap)}} + \underbrace{\left(g^{\mathrm{SEP}}_{\theta,s_t,\varepsilon_{t+1}} - g^{\mathrm{ROM1}}_{\theta,s_t,\varepsilon_{t+1}}\right)}_{\text{nonlinear residual}}.$$

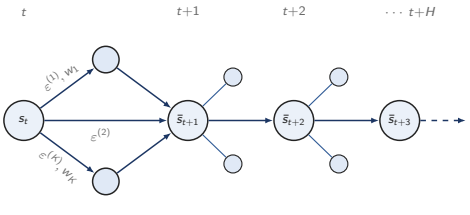
Control-variate logic:
hard object = known cheap object + small unknown object.
Where linear approximation, ROM1, is accurate use it, the network learns only on the nonlinear gap. (Multifidelity surrogates, FOM-ROM)

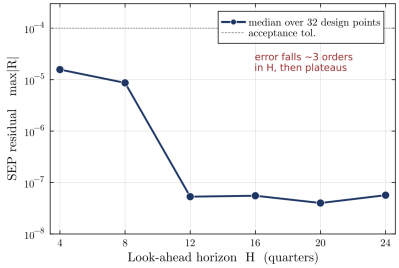
Linear ROM1 understates the contraction as the rate nears the ELB



Investment-specific (eqs) shock swept $0 \to -16$ s.d. at the HLT posterior mean. Colour = nonlinear SEP; grey = linear ROM1 (a tilted plane); red = ELB floor. The surfaces separate as the bound nears, that gap is what the residual learns.

SEP, the offline teacher: a Gauss–Hermite look-ahead tree





- Quadrature: at each step, K GH nodes branch *one period* to form $\mathbb{E}_t$. Problem grows linearly, $t \cdot K$.
- Look-ahead H: carry the expected path H periods, then truncate.
- Residual minimization: solve $R(\mathbf{Y})=0$ by damped Newton (AD Jacobian).

The DGP:
The global, ELB-capable solve is the **benchmark** every cheap approximation is judged against, computed *offline*, never in the posterior loop.

ROM1 is the cheap online model, and its error is the signal

$$s^{+}_{\mathrm{ROM1}} = g^{\mathrm{ROM1}}_{\theta,s,\varepsilon}, \qquad \widehat{\varepsilon} = I^{\mathrm{ROM1}}_{\theta}(y, s).$$

Inversion-filter:

- **Invert** observables to recover structural shocks.
- **Propagate** states (stable over the long recursion).

Why it is the right baseline

- Near the steady state, ROM1 is good.
- Farther away, its forecast error *is* the nonlinear object.

Estimator:
Inversion filter: Faster than the Kálmán filter, as we need the inversion once and do not need to invert, and AD the state covariance matrix each period.

The network learns one small object: the observable residual

$$r_\phi(s, \varepsilon, \theta) \approx y_\theta^{\mathrm{SEP}}(s, \varepsilon) - y_\theta^{\mathrm{ROM1}}(s, \varepsilon).$$

Neural Network Surrogate learns the SEP-ROM residual ($r_\phi(s, \varepsilon, \theta)$):

- **Inputs**, the design point (s, ε, θ), e.g. in SW/HLT it is $44 + 7 + 18 = 69$ dimensions:
  - current state s (44 model states/jumps);
  - structural shock ε (7: TFP, risk-premium, government, monetary, price markup, investment-specific, wage markup);
  - structural parameters θ (the 18 estimated parameters).
- **Output** the SEP$-$ROM1 residual, the nonlinear correction first order misses.

Why:
The network's error is floored at **ROM1**: it learns only the SEP$-$ROM1 *residual*, so ROM1 carries the bulk of the dynamics, NN only corrects it if it off.

It is simply a gate: the activation share is fixed, the threshold calibrated to it

Observable gate statistics

$$f_t = \left\| (y_t - y_t^{\mathrm{ROM1}}) \oslash \sigma_y \right\|, \quad e_t = \left\| \widehat{\varepsilon}_t \oslash \sigma_\varepsilon \right\|$$

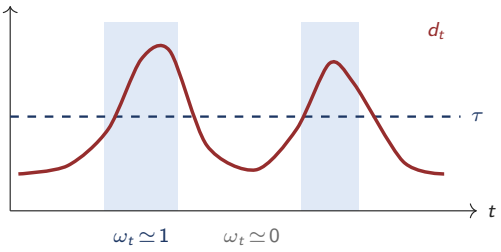
Gate (hard, or smoothed)

$$\omega_t = \mathbf{1}\{e_t > \tau_\varepsilon \text{ or } f_t > \tau_y\}$$

$$\omega_t = \Lambda\big(\beta_\varepsilon(e_t - \tau_\varepsilon) + \beta_y(f_t - \tau_y)\big)$$

Mixed likelihood increment

$$\ell_t = \omega_t\, \ell_t^{\mathrm{SEP}}(\theta) + (1 - \omega_t)\, \ell_t^{\mathrm{ROM1}}(\theta)$$



$d_t = (e_t, f_t)$ crosses $\tau \Rightarrow$ the correction switches on.

Calibration:
I fix the **activation share** and calibrate τ to deliver it, then hold it fixed in estimation. Cross-validation *separately* shows the surrogate is far more accurate than the linear solution far from the steady state. ω_t is data-driven, not a free parameter.

The estimator in four steps: residual, gate, support refinement, NUTS

Offline (once). On a design grid of (s, ε, θ):

$$(y^{\mathrm{SEP}}, s^{\mathrm{SEP}}) = \mathrm{SEP}_\theta(s, \varepsilon), \quad y^{\mathrm{ROM1}} = g^{\mathrm{ROM1}}_{\theta,s,\varepsilon}, \quad \min_\phi \sum \left\| r_\phi(s, \varepsilon, \theta) - (y^{\mathrm{SEP}} - y^{\mathrm{ROM1}}) \right\|^2 \to \widehat{\Sigma}_{\mathrm{sur}}.$$

Online (every NUTS draw θ). At each date t, invert and propagate with ROM1,

$$\widehat{\varepsilon}_{t+1} = I^{\mathrm{ROM1}}_\theta(y_{t+1}, s_t), \qquad s_{t+1} = g^{\mathrm{ROM1}}_{\theta,s_t,\widehat{\varepsilon}_{t+1}},$$

$$\ell_t = \underbrace{\omega_t}_{\text{fixed gate}} \mathcal{N}\big(y_{t+1};\, y^{\mathrm{ROM1}}_{t+1} + r_\phi,\, \Sigma\big) + (1 - \omega_t)\, \mathcal{N}\big(y_{t+1};\, y^{\mathrm{ROM1}}_{t+1},\, \Sigma_K\big).$$

Refine support. If a gate period leaves support, label it:

$$x_t = [s_{t-1}; \widehat{\varepsilon}_t; \theta], \qquad d_t = \max_j \left| \frac{x_{tj} - \mu_j}{\sigma_j} \right|, \qquad d_t \text{ large} \Rightarrow \mathrm{SEP}(x_t) \text{ label.}$$

Sampler. NUTS on $x = T(\theta)$, $\theta = T^{-1}(x)$:

$$\log \pi(x \mid y_{1:T}) = \textstyle\sum_t \ell_t(\theta) + \log p(\theta) + \log |J_T(x)|.$$

Posterior-guided support refinement:
A **differentiable** likelihood, the fixed gate path $\{\omega_t\}$, and a support score d_t that tells us where the surrogate needs new teacher labels.

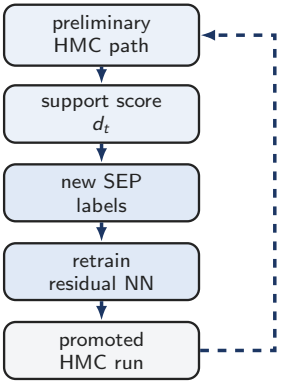
Posterior-guided support refinement: learn where the sampler goes

Problem: The offline SEP design is finite. A posterior run can visit gate-active states that were rare in the original training set.

Key mistake: Extrapolating silently.

1. Run a short gated HMC audit.
2. Score each active input $x_t = [s_{t-1}; \widehat{\varepsilon}_t; \theta]$.
3. Send high-d_t points back to SEP.
4. Append $g^{\mathrm{SEP}}(x_t) - g^{\mathrm{ROM1}}(x_t)$, retrain, rerun.

Solution: Posterior-guided support refinement. The true structural shock is unchanged; the NN support grows to cover it.



Verifications, step by step, closing on the one for Q&A

- ✓ **Map accuracy.** 10,000 supported SEP cells: surface RMSE 88.1 → **0.05**; residual RMSE 0.144 → **0.0007**.
- ✓ **Posterior recovery.** Galí hard-ELB bridge: direct-SEP and surrogate HMC posteriors agree.
- ✓ **Support refinement.** Current HLT pass: 113 hard-gate periods relabelled; all solve; retrained residual beats ROM1 by 96–98%.
- ✓ **End-to-end real data** (preliminary)**.** Full HLT stack runs with matched gate+NN and linear chains; **zero post-warmup divergences**.
- ✓ **Computational speed.** $\sim$**2,450**$\times$ cheaper per draw than direct SEP; bootstrap particle filter degenerates (ESS $= 1$).

Take-away::
Every layer is checked against direct SEP where feasible; the $\sim$2,450$\times$ speed-up is what makes the estimator usable.

The model I take to data: SW07 with the HLT lower bound

1. Effective lower bound

$$R_t = 1 + \tfrac{1}{\kappa_s}\log\!\left(1 + e^{\kappa_s(\tilde{R}_t - 1)}\right) \xrightarrow{\kappa_s \to \infty} \max\{1, \tilde{R}_t\}, \qquad \kappa_s = 100$$

2. Investment adjustment cost

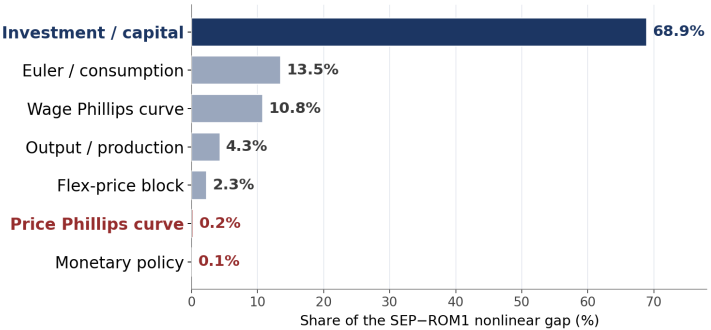
$$\bar{K}_t = (1-\delta)\bar{K}_{t-1} + q_t^s\left[1 - S\!\left(\tfrac{I_t}{I_{t-1}}\right)\right] I_t, \qquad S\!\left(\tfrac{I_t}{I_{t-1}}\right) = \tfrac{S''}{2}\left(\gamma \tfrac{I_t}{I_{t-1}} - \gamma\right)^2, \quad S'' = 6.01$$

$S(1) = S'(1) = 0$ at trend growth γ, a *pure second-order* term in the linearized capital law and Tobin-q FOC. (Capital utilization $a(z)$ adds further curvature inside this block.)

3. Kimball price & wage aggregator

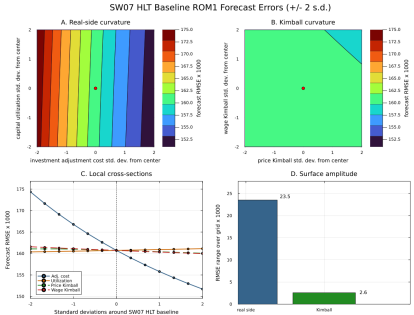
$$D_p \equiv \int_0^1 \left(\tfrac{P_t(i)}{P_t}\right)^{-\varepsilon_p/(\varepsilon_p - 1)} di$$

What linearization misses: 69% is the investment block



The mechanism is transparent:
The investment adjustment cost has $S(1) = S'(1) = 0$ at trend, so it is **invisible to first order**. Decomposition over 22,080 SEP solves; the price Phillips curve, where Kimball enters, is just 0.2%.

How the nonlinearity evolves: the SEP−ROM1 curvature surface



SEP−ROM1 one-step forecast-error curvature over a ± 2 s.d. grid *centred at the estimated posterior*: real-side (investment/utilization) curvature dominates, Kimball is flat, the surface behind the 68.9%/0.2% decomposition.

The error is a curvature, not noise:
The gap to ROM1 is **systematic**, concentrated in the real (investment) block and growing smoothly with distance from the steady state, where first order is weakest.

Which nonlinearity dominates is local to where HLT is estimated

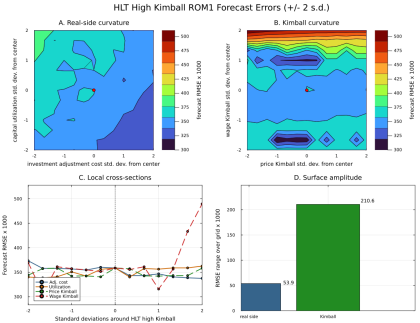
Local curvature range (1000×RMSE, ±2 s.d.)

Grid centre	Real-side	Kimball
Estimated HLT posterior	**23.5**	2.6
High-Kimball calibration	53.9	**210.6**

(real-side largest in 38–40 of 40 posterior draws)

How HLT is estimated: a **linear** Kalman likelihood on the *first-order* model, the ELB linearized away.

Why it is local: first order discards the investment cost and ELB kink, a local expansion that costs ∼**36 nats** against the global solve.



High-Kimball centre (Harding–Lindé–Trabandt 2022): Kimball curvature (210.6) towers over the real side (53.9).

Local, not global:
Near the estimated HLT posterior, **investment** dominates. At a high-Kimball calibration, Kimball can dominate. The ranking is local to the posterior region.

To explain 2020, the linear smoother needs impossible shocks

To fit the pandemic, the linear Kalman smoother attributes it to a 127σ **risk-premium shock** (2020Q1–2021Q2, at the linear estimate) a draw that should never occur. That is the first-order model signalling misspecification in the ELB/COVID regime. The switching surrogate needs smaller, 56.7σ risk-premium shock, and has lower forecasts.

One-step-ahead forecast RMSE, COVID window: Switching surrogate vs. ROM1:

	output	cons.	invest.	wage	aggregate
RMSE reduction	29%	24%	26%	8%	**16%**

Nonlinearity bites where the bound binds:
In the acute COVID window the switching surrogate cuts one-step forecast RMSE by **16%** aggregate and **26%** on investment.

Per draw, $\sim$2,450$\times$ cheaper than direct SEP, but the offline cost is real

Likelihood evaluation	per draw	1,000-draw run
Linear Kalman (ROM1)	0.003 s	100 min
Regime-switching surrogate	**0.06 s**	**2.5 h**
Bootstrap PF + SEP (N_p=100, 100 cores)	$\approx$147 s	$\approx$1.7 days

- Bootstrap particle filter **degenerates**: ESS $= 1$ every period, N_p up to 5,000.
- Single SEP solve 0.8 s, fatal in the leapfrog loop.
- Surrogate NUTS: 0 divergences, ESS $\geq$ 473.
- Total runtime: 14.4 h = 10 h data + 2 h training + 2.5h estimation

The offline price is not paid "once" for free:
Building the teacher set means simulating the model across the design space, and that grows with dimension. The win is *amortization*: one simulation campaign, reused across every posterior draw.

Takeaways: a feasible nonlinear estimator, and where the nonlinearity lives

1. SEP supplies the accurate transition **offline**; ROM1 carries the cheap **online** recursion.
2. The network learns only the SEP–ROM1 observable residual.
3. The gate switches the NN on only when nonlinear information is needed.
4. Posterior-guided support refinement sends extrapolation points back to SEP.
5. Validation is posterior-level: the 10,000-cell bridge matches direct SEP to $\sim$1650$\times$ tighter surface RMSE.

Source of non-linearity:
In the Smets–Wouters/HLT, **69% of what linearization misses is the investment block**, only 0.2% is the price Phillips curve.

Switching-fidelity inversion filter::
A good nonlinear estimator should know when *not* to be nonlinear.

Thank you for your attention!

Questions and comments welcome.

Backup: notation and one-step error

$$r_\theta^\star(s,\varepsilon) = z_\theta^{\mathrm{SEP}}(s,\varepsilon) - z_\theta^{\mathrm{ROM1}}(s,\varepsilon), \qquad r_\phi(s,\varepsilon,\theta) \approx r_\theta^\star(s,\varepsilon).$$

$$\widehat{z}_{\theta,\phi}(s,\varepsilon) = z_\theta^{\mathrm{ROM1}}(s,\varepsilon) + \omega\big(d(s,\varepsilon,\theta)\big)\, r_\phi(s,\varepsilon,\theta).$$

One-step surrogate error:

$$\|z_\theta^{\mathrm{SEP}} - \widehat{z}_{\theta,\phi}\| = \big\|(1-\omega)\,(z_\theta^{\mathrm{SEP}} - z_\theta^{\mathrm{ROM1}}) - \omega\,(r_\phi - r^\star)\big\|.$$

Set $\omega \simeq 0$ where ROM1 is accurate; $\omega \simeq 1$ where the residual is accurate on support; otherwise flag weak support and benchmark with direct SEP.

Backup: the gate in detail

$$f_t = \left\| \left(y_t - y_t^{\mathrm{ROM1}}\right) \oslash \sigma_y \right\|, \qquad e_t = \left\| \widehat{\varepsilon}_t \oslash \sigma_\varepsilon \right\|.$$

- f_t: normalized **linear forecast error**; e_t: normalized **recovered-shock norm**.
- Fix the target activation share; calibrate τ_ε, τ_y to it by bisection (`calibrate_tau`).
- Contiguous active regimes are padded to a minimum length; a soft gate mixes likelihood across dates.
- Where the gate does not fire, the estimator is exactly ROM1/Kalman, never a bad extrapolation.

Backup: relation to existing methods

Approach	Main computational object	Inference object
Particle filters	simulated latent-state paths	noisy likelihood estimate
Projection / OBC	nonlinear policy approximation	filter / MCMC around the policy
Neural solvers	learned nonlinear policy or equilibrium map	amortized likelihood / simulation
This deck	**SEP–ROM1 residual**	**smooth switched likelihood for HMC**

Backup: which equations drive the SEP–ROM1 gap?

Equation block	Baseline share	High-curvature share
Investment / capital	**68.9%**	**76.5%**
Price Phillips curve (Kimball)	0.2%	small

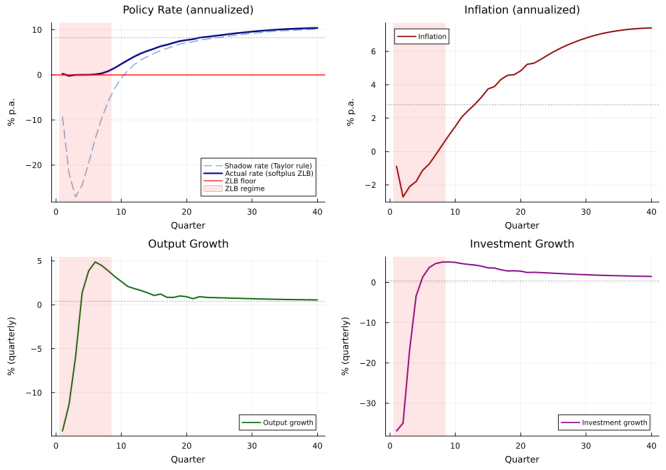
- Dominant drivers at the posterior: `csadjcost` (investment adjustment) and `czcap` (utilization).
- Kimball price/wage curvature (`curvp`, `curvw`) contributes little near the estimated region.

Backup: fixed-θ cross-evaluation

Evaluated at	full gate+NN LL	linear+gate LL
full gate+NN posterior mean	-1382.805	-1382.800
linear+gate posterior mean	-1382.882	-1382.877

Reading:
The neural correction is doing little *here* because the estimated posterior region is mostly linear. The residual delivers in the bridge and high-Kimball regions, exactly where the model is nonlinear.

Backup: a hard case where linear and nonlinear visibly part



Same shocks: nonlinear occasionally-binding model versus its linearization. The split the surrogate must capture.

Backup: one date of the switching-fidelity likelihood

$$\widehat{\varepsilon}_{t+1}(\theta) = I_\theta^{\mathrm{ROM1}}(y_{t+1}, s_t), \qquad \widehat{s}_{t+1} = g_{\theta, s_t, \widehat{\varepsilon}_{t+1}}^{\mathrm{ROM1}},$$
$$\widehat{y}_{t+1} = y_{t+1}^{\mathrm{ROM1}} + \omega_t \, r_\phi^{obs}(s_t, \widehat{\varepsilon}_{t+1}, \theta).$$

- ROM1 inversion recovers the shock, no particle filter.
- ROM1 propagation keeps the recursion stable.
- The gate ω_t decides whether the residual enters this date's likelihood.

Filter-free:
The shock is **recovered**, not integrated out, and the observable is corrected only on the dates the gate selects.

Backup: the HMC target

$$x = T(\theta), \qquad \log \pi(x \mid y_{1:T}) = \ell_{\text{profile}}\{T^{-1}(x)\} + \log p\{T^{-1}(x)\} + \log |J_T(x)|.$$

- AdvancedHMC / NUTS with logit transforms, HMC-based DSGE estimation as in **Farkas–Tatár (2020)**.
- Shocks recovered by ROM1 inversion, no latent-state sampling.
- Finite-difference gradients today; differentiating through the inversion is a clean next step.

No filter, no latent states:
Sample θ; the shocks are recovered inside the likelihood. The target is a smooth function of θ, exactly what NUTS needs.