Recurrent depth as test-time compute: parameter-efficient, extrapolating, and bounded by contractivity

W. Schulz

Jul 27, 2026

Abstract

A weight-tied recurrent-depth network matches a feedforward net at 1/8 the parameters and wins out of distribution (+0.36 ± 0.13 over 10 seeds), extrapolating about 8× in problem size. A non-contractive residual update causes divergence and overthinking; a gated contractive update fixes both.

The shippable result of the Quantum-Intelligence test-time-compute arc (experiments 21–50, 2026-06-27). Self-contained; cites the per-experiment logs for detail. House rule throughout: pre-register, run once, report failures, equal-budget baselines.

The question

After the project's continual-learning ("wave/Thesis B") and photonic-compute theses were falsified/retired (see `synthesis-and-claims-register.md`), one survivor remained: latent test-time-compute shaping — can "thinking longer" be a real, substrate-agnostic compute axis? We chased it from toy energy landscapes to a genuinely trained model.

The arc, in one line per stage

1. Two-substrate toy result (exps 21–24). On Kuramoto (slow-dynamics) settling, *depth is a real compute axis*; on Hopfield/attention (one-shot contraction) it is *inert*. Both share Tunnel Vision; width (parallel restarts) escapes; a settle-then-fork hybrid beats both pure axes. "No-overthinking" looked like a keeper.
2. Stress round (exps 29–36). No-overthinking is convergence-gated (breaks off-convergence); the depth↔width crossover is a ruggedness law ($\sigma^* \approx 0.31$); the hybrid's value is *combining*, not the trigger; the energy verifier is exploitable and the coverage-vs-selection bottleneck is ε-dependent; sleep-time warm-start is a real axis; Tunnel Vision is an engineerable knob.
3. The bridge to a trained model (exps 37–39). numpy couldn't master the regime; torch (exp 39, py3.12 **`.venv312`**) did: a residual recurrent-depth GNN masters reachability and EXTRAPOLATES (harder N=32: 0.48@k8 → 0.86@k64, beyond trained $k \leq 16$).
4. Hardening + boundary (exps 40–48). Recurrent depth is worth it (exp 43, headline): 0.13× the params of a matched feedforward net, in-dist parity, +0.28 on out-of-distribution hardness it structurally cannot reach. Extrapolates ~8× (N=128, no ceiling). But it's bounded: it *diverges* on unbounded regression (shortest-path) and *overthinks* on non-monotone XOR (parity). One root cause: the non-contractive residual `h←h+tanh(...)` never halts.

5. The fix (exps 49–50). A contractive/gated update `h←(1−g)h+g·tanh(cand)` (exp 49) fixes the catastrophic failures — shortest-path divergence (MAE 2.04→0.29) and parity collapse (0.09→1.00, mastered & stable) — and extrapolates *faster*. A convergence penalty (exp 50) additionally flattens regression drift but over-constrains monotone propagation: the right halt is adaptive, not blunt.

What is established (the defensible claims)

1. Test-time depth is a real, parameter-efficient compute axis on a trained model. A weight-tied recurrent-depth net matches a matched-compute feedforward net in-distribution at 1/8 the parameters and beats it by +0.28 *[Audited 2026-07-27, N=10 seeds — `continual-learning/FINDINGS-T1-seed-audit.md`: holds 9/10; OOD edge is +0.356 ± 0.130, i.e. the shipped +0.28 sits below the distribution mean.]* where the problem needs more iterations than the feedforward net's baked-in depth (exp 43). Weight-tying turns depth into a runtime dial.
2. It extrapolates far — to ~8× the trained problem size with no ceiling found (exp 42), with useful depth scaling with problem hardness.
3. Its domain is set by one mechanism. Plain residual recurrence works only for bounded, monotone fixpoints; a contractive/gated update widens it to unbounded-regression and non-monotone tasks (exp 49). Overthinking, the depth→width crossover, and the regression/parity failures are all the same non-contractive-residual root cause.
4. Substrate-agnostic. None of this is about "waves" — it is the recurrent-depth *mechanism*. This is consistent with the project's reckoning that the complex/wave field is load-bearing nowhere.
5. Companion levers transfer: sleep-time warm-start (exp 35/47) and the verifier ε/d′ / coverage-vs-selection picture (exps 32/34/46) both reproduce on the trained model.

What is falsified or bounded (honest)

- "No-overthinking" is not universal — it is convergence-gated (exps 29, 40); a vanilla residual recurrence drifts past the useful depth, and a blunt convergence penalty is task-dependent (exp 50).
- Depth-extrapolation is not universal for the residual cell — fails on unbounded regression and non-monotone targets (exps 45, 48); the gated cell is required to extend it.
- Width-escape on a trained model remains cleanly untested — the graph tasks the model masters have little residual failure for width to rescue (exp 41); it needs a genuinely hard CSP. Open.

Open threads (incremental, ranked)

1. A learned adaptive halt (per-node stop / halt-after-convergence) to remove the last residual drift without over-constraining propagation (exp 50 pointed here).
2. Width-escape on a hard CSP (3-coloring / Sudoku-lite) — the one toy finding never cleanly confirmed on a trained model.
3. A real reasoning benchmark (ListOps / arithmetic) with a recurrent-depth transformer — the final rung from synthetic graphs to language-ish reasoning.

Reproduce

- Toy (numpy): `continual-learning/experiments/2[1-9]_*.py`, `3[0-6]_*.py` with `.venv`.

- Trained (torch): `continual-learning/experiments/39_*.py`, `4[0-9]_*.py`, `50_*.py`, shared `experiments/ttc_torch/common.py`, with `.venv312` (py3.12; py3.13 has no torch wheel).
- Logs: `docs/recurrent-depth-bridge.md` (exps 37–39), `docs/next-10-steps-round3.md` (29–36), `docs/next-10-steps-round4.md` (40–50).

Bottom line

Across the full arc the project's grand theses fell, but the falsification discipline surfaced one real, validated, mechanistically-understood result: test-time compute via recurrent depth is a parameter-efficient, far-extrapolating reasoning axis on a trained model, whose domain is governed by the contractivity of the update rule. That is a small, correct, defensible contribution — exactly the kind this project was built to find.

Published by NeuroGenesys (AW3 Technology, Inc.). Canonical version: https://neurogenesys.si/research/recurrent-depth-test-time-compute