WHEN RECURSIVE MODELS FINISH COMPUTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Recursive models can continue updating their latent states beyond their nominal inference budget, so an incorrect output at that budget does not show whether computation is unfinished or has entered a persistently unsuccessful regime. We study the dynamics of completion in attention and MLP-based Tiny Recursive Models (TRMs) on 1,000 hard Sudoku puzzles. Extending recurrence from the nominal 16 steps to 512 steps increases cumulative exact-solve accuracy from 59.2% to 87.5% for the attention model and from 74.4% to 91.9% for the MLP model, solving more than two-thirds of the puzzles unsolved in the nominal budget. Across both architectures, latent-state motion drops sharply after the first exact solution. Completed states are typically locally contractive along the trajectory direction, even though the same local Jacobian retains strongly expanding directions. We characterize this phenomenon as *trajectory-conditioned anisotropic stability*. Perturbation experiments confirm this directional stability across both models. The multi-step fate of the maximally expanding direction differs: it is absorbed within 16 steps in the attention model but persists longer in the MLP model. The anisotropic-stability pattern also holds for a second attention checkpoint. Together, these results distinguish nominal-budget failure from completed computation and identify a common dynamical signature of completion across two recurrent architectures.

1 INTRODUCTION

Recursive models compute by iteratively updating a latent state, applying the same learned transition function at each step. This provides additional computational depth through weight sharing (Dehghani et al., 2019; Giannou et al., 2023) and enables computation to continue across multiple iterations (Graves, 2016; Schwarzschild et al., 2021; Bansal et al., 2022; Geiping et al., 2025). Hierarchical Reasoning Models (HRMs) and Tiny Recursive Models (TRMs) use this principle to build compact reasoning systems (Wang et al., 2025; Jolicoeur-Martineau, 2025). In such models, the number of recurrent steps determines how long the model can compute.

This creates a fundamental ambiguity. An incorrect answer at the end of the inference budget can mean two different things. The model may have failed, or it may simply need more computation (Anil et al., 2022). Conversely, a correct output does not tell us whether the underlying state has settled. The model may already have reached the answer, while its internal state continues to change (Knutson et al., 2026). Endpoint accuracy alone does not distinguish *failure*, *completion*, or *stability*.

Recurrent computation has long been studied via its state-space dynamics. Fixed points, attractors, and local Jacobians have been used to understand how recurrent models represent and transform information (Sussillo & Barak, 2013; Maheswaranathan et al., 2019). Equilibrium models make the connection between computation and fixed points explicit (Bai et al., 2019; Huang et al., 2026). Recent mechanistic studies of HRMs and TRMs have identified attractor structure, failure modes, spurious fixed points, and abrupt changes in solution quality (Efstathiou & Balwani, 2026; Ren & Liu, 2026). Complementary work on recurrent-depth reasoners has studied settling and the conditions under which additional test-time depth remains useful and stable (Viakhirev et al., 2026), while recent dynamical-systems analyses connect prolonged reasoning to transient chaos and fractal basin structure (Lai et al., 2026).

We ask a complementary question: *what changes in the recurrent state when a particular computation finishes?* Rather than comparing all trajectories at the same recurrent step, we align each problem

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

to the step at which it is first solved. We then study completion, latent motion, and local stability separately. Completion is the first step at which the exact solution is produced. Latent motion measures how much the recurrent latent state changes from one step to the next. Stability describes how perturbations of that state grow or shrink. We measure it using the local Jacobian and controlled finite perturbations. This lets us distinguish directional from uniform stability. The model may become stable along the direction its own computation follows, even though the local Jacobian contains an expanding direction.

We study this question in attention and MLP-based TRMs on the same 1,000 hard Sudoku puzzles. Both models have a nominal budget of 16 recurrent steps. We continue the unchanged recurrence to 512 steps. Cumulative exact solution rises from 59.2% to 87.5% for the attention model and from 74.4% to 91.9% for the MLP model. Among the puzzles that are unresolved at step 16, 69.4% and 68.4% are solved later, respectively. In both models, more than two-thirds of the apparent failures in the nominal budget are unfinished computations. This separates the puzzles into three completion groups. EARLY solvers are solved by step 16. LATE solvers are solved between steps 17 and 512. PERSISTENT puzzles remain unsolved through step 512. Once aligned with their own solution time, EARLY and LATE solvers show a similar completion pattern. Their latent motion decreases, and they enter a directionally stable regime. The PERSISTENT trajectories remain much more active and show a different stability profile.

Completion has a directional stability signature. Latent-state motion drops sharply after completion. Completed states are locally contractive along the direction in which the model is moving. However, the same local Jacobian retains strongly expanding directions. The recurrence, therefore, does not become uniformly contractive when the task is solved. Perturbation experiments confirm this directional contrast. Perturbations along the most expanding direction grow much more than perturbations of the same size along the model's natural direction of motion. We call this coexistence *trajectory-conditioned anisotropic stability*. The computation becomes stable along the direction of motion while the recurrent map retains locally expanding directions.

We repeat the full analysis in the attention and MLP models, using the same puzzles, recurrent schedule, probe protocol, and statistical procedures. The qualitative features replicate: unsolved trajectories remain active, completed trajectories settle, the natural direction is locally contractive after completion, and expanding directions remain in the Jacobian. But after completion, perturbations along the most expanding direction are absorbed within 16 steps in the attention model (§3.4) and persist in the MLP model (§3.5).

We also present supporting evidence beyond the two primary hard Sudoku runs. A second attention checkpoint, Attention-B, shows a similar anisotropic-stability pattern (Appendix G.7, Table 5). In Easy Sudoku, most puzzles are solved within the first few recurrent steps and fall into the EARLY-solver regime. Their latent dynamics closely resemble those of the EARLY solvers in Hard Sudoku (Appendix J). In Maze-Hard, a separate 30×30 grid task, most mazes are also solved within the first few recurrent steps and therefore behave mainly like EARLY solvers. Maze-Hard also shows that exact-match accuracy can be misleading. Some predictions differ from the ground-truth reference path, but are still valid solutions (Appendix K). The full extended-recurrence, Jacobian, and perturbation analyses are therefore centered on Hard Sudoku, where both early and late completion are well represented.

2 EXPERIMENTAL FRAMEWORK

2.1 MODELS, TASK, AND EXTENDED RECURRENCE

We use the TRM architecture of Jolicoeur-Martineau (2025) (see its Figures 1 and 3) and evaluate two trained checkpoints on the 1,000 hard Sudoku puzzles. **Attention-A** is a 6.83M-parameter recurrent transformer checkpoint (Gao, 2025a; Samsung SAIL Montreal, 2025). The **MLP** checkpoint replaces attention with sequence-axis MLP mixing and removes positional encoding (Gao, 2025c). Both use the same recurrent schedule, probe protocol, and statistics. The configuration details are shown in Appendices C and G.1.

The 1,000 puzzles come from the Sudoku-Extreme test split (Wang et al., 2025). All belong to its `puzzles4_forum_hardest_1905` source, the `forum_hardest_1905` list collected on the

Enjoy Sudoku players' forum and distributed with the `tdoku` benchmark suite (Dillon, 2019). Both checkpoints have a nominal 16-step outer budget. We reproduce the 16-step evaluation and continue the learned recurrence to a diagnostic horizon of 512 steps with the same weights, inputs, and pre-processing.

2.2 Recurrent state, motion, and completion groups

In the recurrent step t, the model carries two latent states, $Z_H^{(t)}$ and $Z_L^{(t)}$. We combine them into a single state,

$$s_t = (Z_H^{(t)}, Z_L^{(t)}), \qquad s_{t+1} = F_\theta(s_t; x). \tag{1}$$

All state-space analyses use this joint state, with dimension $2 \times 97 \times 512$. We measure how much each latent state component Z_H, Z_L changes between consecutive steps using the relative update

$$\delta_B(t) = \frac{\|B^{(t)} - B^{(t-1)}\|_2}{\|B^{(t-1)}\|_2}, \qquad B \in \{Z_H, Z_L\}. \tag{2}$$

A small value of $\delta_B(t)$ means that the latent state changes little from one step to the next. For each puzzle i, let τ_i denote the first recurrent step at which the decoded Sudoku is exactly correct. We divide the puzzles into three groups: Early solvers have $\tau_i \leq 16$, Late solvers have $16 < \tau_i \leq 512$, and Persistent puzzles remain unsolved through step 512. When comparing trajectories around completion, we measure time relative to the first solve: $r = t - \tau_i$. Here, $r = 0$ is the solving step; negative values precede completion, and positive values follow.

2.3 Local stability along the trajectory

To characterize the dynamics along the model's trajectory, we linearize the recurrent map at each visited state,

$$J_t = \left.\frac{\partial F_\theta}{\partial s}\right|_{s_t}. \tag{3}$$

We use it to distinguish stability along the direction followed by the model's trajectory from the largest amplification available in the surrounding state space. We define the unit trajectory direction

$$d_t^{\text{out}} = \frac{s_{t+1} - s_t}{\|s_{t+1} - s_t\|_2}, \tag{4}$$

whenever $\|s_{t+1} - s_t\|_2$ is numerically nonzero. Its one-step Jacobian gain is

$$\gamma_t^{\text{nat}} = \|J_t d_t^{\text{out}}\|_2. \tag{5}$$

Perturbations along the trajectory direction therefore contract to first order when $\gamma_t^{\text{nat}} < 1$ and grow when $\gamma_t^{\text{nat}} > 1$. Since $\Delta_{t+1} \approx J_t \Delta_t$ for $\Delta_t = s_{t+1} - s_t$, γ_t^{nat} approximates the ratio $\|\Delta_{t+1}\|_2 / \|\Delta_t\|_2$ of successive updates wherever the linearization is accurate. Thus, γ_t^{nat} provides a local dynamical interpretation of shrinking or growing latent updates. Its additional value comes from comparing the gain along the direction actually followed by the trajectory with the amplification available along other directions at the same state. In particular, $\gamma_t^{\text{nat}} < 1$ indicates contraction only along the local trajectory direction and does not imply contraction of the full surrounding state space.

To measure the strongest local amplification available at the same state, we compute the spectral norm of the Jacobian,

$$\sigma_{\max}(J_t) = \max_{\|v\|_2 = 1} \|J_t v\|_2 = \sigma_1(t). \tag{6}$$

Let $v_1(t)$ denote a corresponding leading right singular vector,

$$v_1(t) \in \arg\max_{\|v\|_2 = 1} \|J_t v\|_2. \tag{7}$$

Under one linearized recurrent step,

$$J_t v_1(t) = \sigma_1(t)\, u_1(t), \tag{8}$$

where $u_1(t)$ is the corresponding leading left singular vector. Here $v_1(t)$ is the most expanding local perturbation direction, $\sigma_1(t)$ is its one-step amplification factor, and $u_1(t)$ is its output direction.

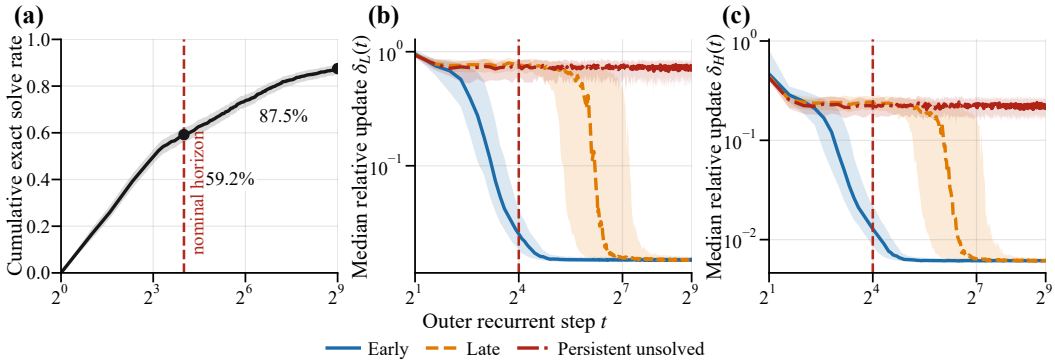

Figure 1: The nominal horizon truncates active computation. (a) Cumulative exact solve rate over 512 outer steps with Wilson bands; the dashed line is the nominal 16-step budget. (b,c) Median relative latent update with interquartile bands for the three completion groups. LATE solvers are as mobile as PERSISTENT ones at step 16 and settle only when they solve. PERSISTENT trajectories never settle.

The comparison between γ_t^{nat} and $\sigma_1(t)$ distinguishes stability along the trajectory from worst-case local stability. In particular,

$$\gamma_t^{\mathrm{nat}} < 1 \qquad \text{and} \qquad \sigma_1(t) > 1 \tag{9}$$

indicates anisotropic local dynamics: to first order, the trajectory-aligned direction is contracting even though expanding directions remain available in the surrounding state space.

We compute the leading singular triplet (σ_1, v_1, u_1) without explicitly constructing the Jacobian. We apply matrix-free power iteration to $J_t^\top J_t$ using forward-mode Jacobian-vector products and reverse-mode vector-Jacobian products. Implementation details, convergence checks, and numerical validation appear in Appendix E. We also compute the same directional gains for probes restricted to either the Z_H or Z_L block.

The squared alignment $A_1(t) = |\langle d_t^{\mathrm{out}}, v_1(t)\rangle|^2$ measures how closely the model's own trajectory aligns with the most expanding direction. We compare this $A_1(t)$ with an isotropic random-direction baseline evaluated at the same state.

Finally, we measure whether the amplified direction persists across recurrent steps. We define

$$T_1(t) = |\langle u_1(t), v_1(t+1)\rangle|^2. \tag{10}$$

A large T_1 means that the direction produced by maximal amplification at step t is aligned with the most expanding input direction at the next step. A small value means that this direction is largely rotated away. To probe stability beyond a single step, we measure finite-horizon growth by perturbing s_t along a unit direction v, with $\tilde{s}_t = s_t + \varepsilon v$. After evolving both trajectories for k recurrent steps, let

$$\delta s_{t+k}^{(v)} = \tilde{s}_{t+k} - s_{t+k}$$

denote their separation. We define

$$\lambda_k(v) = \frac{1}{k} \log \left(\frac{\|\delta s_{t+k}^{(v)}\|_2}{\varepsilon} \right). \tag{11}$$

Negative values indicate decay of the perturbation over k steps, while positive values indicate growth. This is a finite-horizon, trajectory-dependent measure of perturbation growth.

3 RESULTS

3.1 EXTENDED RECURRENCE REVEALS UNFINISHED COMPUTATION

Continuing the same recurrence beyond its nominal 16-step horizon changes the picture substantially. The exact solve rate rises from 59.2% at step 16 to 87.5% by step 512 (Figure 1a). Of the 408 puzzles that are still unsolved at step 16, 283 are solved later. About two-thirds of the apparent failures at the nominal horizon are not failures at all. They are computations that have not yet finished. LATE solvers

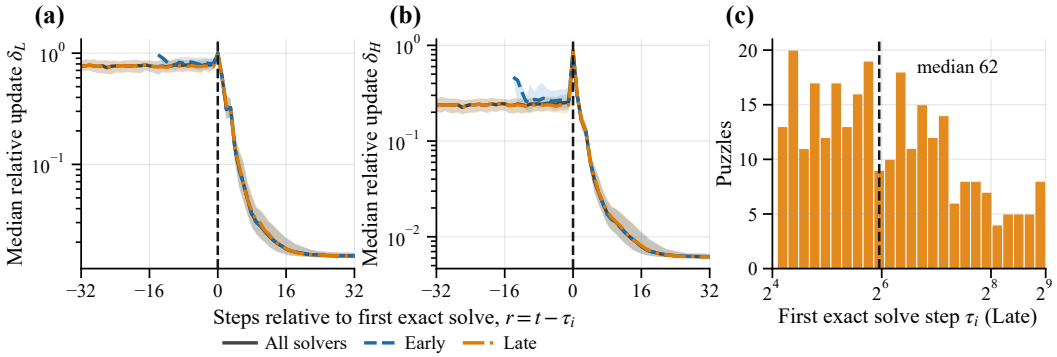

Figure 2: Settling is aligned to each puzzle's own completion. (a,b) Median relative update on the solve-aligned axis $r = t - \tau_i$, with interquartile bands. Motion is roughly constant before completion, spikes at $r = 0$, and then falls by more than an order of magnitude. The EARLY and LATE curves coincide once aligned, although their absolute solve times differ by an order of magnitude. (c) Distribution of τ_i for LATE solvers.

typically need many more steps: their median first solve occurs at step 62 (interquartile 33-130), although some solve much later.

The latent dynamics of these late solvers support the same interpretation. At step 16 (Figure 1b,c), late solvers are still moving strongly and look much more like persistently unsolved trajectories than early solvers. For example, their median δ_L is 0.763, compared with only 0.025 for early solvers. By step 512, however, late solvers have settled to the same low-motion regime as early solvers, while persistent trajectories remain highly active (Figure 1b,c). This yields three populations to analyze: EARLY solvers, LATE solvers, and PERSISTENT puzzles.

Puzzle difficulty does not explain this separation. Persistent puzzles are not systematically harder according to the dataset ratings. If anything, their median rating is slightly lower. Full difficulty statistics are reported in Appendix D.

3.2 LATENT SETTLING ALIGNS WITH COMPUTATIONAL COMPLETION

When trajectories are aligned to each puzzle's first solve time τ_i, the EARLY and LATE solvers follow a similar pattern (Figure 2). For LATE solvers, latent motion stays roughly constant before completion and then drops sharply after the solution is reached. In Z_L, the median motion after completion is less than half of the pre-solve level. After completion, LATE solvers settle to the same low-motion regime as EARLY solvers.

The solving step is a large reorganization. The strongest change occurs exactly at the solving step, $r = 0$ (Figure 2a,b). For LATE solvers, the median δ_H rises from 0.238 at $r = -1$ to 0.878 at $r = 0$. For 99.3% of LATE solvers, the Z_H update at the solving step is larger than at every earlier step in the trajectory. The effect is much weaker in Z_L, where the solving step is the largest update for only 35% of puzzles. This spike is not explained simply by a change in the decoded grid. Many earlier steps also change the prediction without solving the puzzle, but their Z_H updates are much smaller. The solving step, therefore, marks an unusually large reorganization of the high-level latent state. It is followed immediately by a sharp reduction in motion.

Solutions are usually stable. Solved puzzles stay in the recurrence, and almost all of them remain solved. Of the 875 puzzles that solve at least once, 859 never lose the solution again. The remaining 16 temporarily revert, but all recover by step 512. We therefore use the first exact solution as the completion time.

3.3 COMPLETED COMPUTATION IS DIRECTIONALLY STABLE, NOT UNIFORMLY CONTRACTIVE

Unlike our earlier analysis of latent settling, Jacobian analysis asks whether the perturbations along different state-space directions are amplified or contracted. The completed trajectories are typically locally contracting along their own direction of motion, while the same local Jacobian has strongly

Table 1: Local stability at the nominal horizon and at step 512, by completion group. γ^{nat} is the Jacobian gain along the model's own direction of travel. $\sigma_{\max}$ is the worst-case gain over all directions in the same state. $\Pr[\gamma^{\mathrm{nat}} < 1]$ represents the fraction of states in that group having $\gamma^{\mathrm{nat}} < 1$.

Step	Group	γ^{nat}	$\Pr[\gamma^{\mathrm{nat}} < 1]$	$\sigma_{\max}$
16	EARLY	0.83	0.74	25.7
	LATE	2.28	0.24	392.2
	PERSISTENT	3.44	0.12	529.4
512	EARLY	0.25	0.96	26.8
	LATE	0.31	0.92	26.9
	PERSISTENT	2.09	0.30	316.9

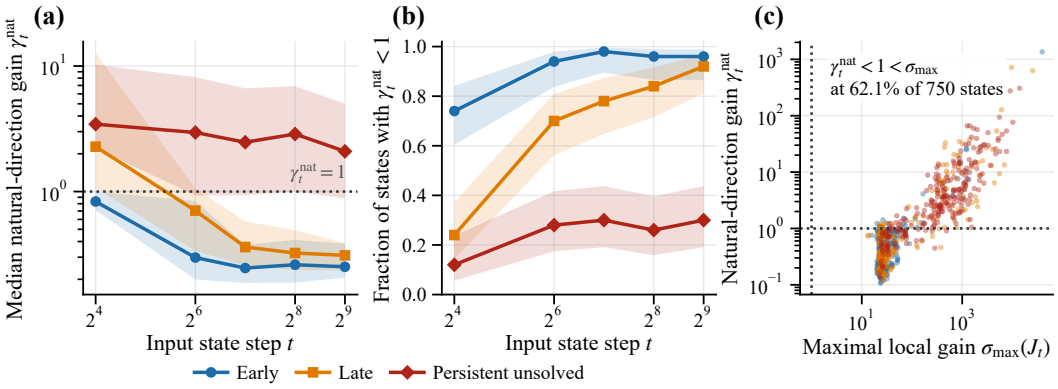

Figure 3: Stability is conditioned on the direction of travel. (a) Median natural-direction gain γ_t^{nat} by group and input-state step, with interquartile bands. The dotted line is $\gamma^{\mathrm{nat}} = 1$. (b) Fraction of states with $\gamma_t^{\mathrm{nat}} < 1$, with Wilson bands. (c) Every probed state: γ_t^{nat} against $\sigma_{\max}(J_t)$. No state has $\sigma_{\max} < 1$, yet most completed states sit below $\gamma^{\mathrm{nat}} = 1$.

expanding directions. We evaluate the Jacobian at 750 visited states, corresponding to 150 puzzles at five fixed recurrent steps each. The recovered singular vectors have unit norm up to numerical precision. Additional numerical checks and implementation details are in Appendix E.

Table 1 and Figure 3 show the main result. At step 16, the three completion groups have different natural-direction gains. EARLY solvers have a median γ^{nat} of 0.83, whereas LATE and PERSISTENT trajectories remain locally expansive, with medians of 2.28 and 3.44.

LATE solvers change once they finish computing. By step 512 (Table 1), their median γ^{nat} has fallen to 0.31, close to the EARLY value of 0.25 (Figure 3a). Most completed states then contract locally in the direction the model is moving. 96% of EARLY states and 92% of LATE states have $\gamma^{\mathrm{nat}} < 1$. At that point, every LATE solver has completed, so the two groups occupy the same directionally stable regime. Because Figure 3 compares fixed recurrent steps, it shows where the groups end up rather than when the change happens. The solve-aligned transition is in latent motion: aligned to their own solve step, LATE and EARLY solvers show the same drop in latent updates (Figure 2a,b). PERSISTENT trajectories remain different, with median $\gamma^{\mathrm{nat}} = 2.09$.

This stability is not a property of the full recurrent map. The worst-case Jacobian gain remains larger than one at every probed state. At step 512, $\sigma_{\max}$ is about 27 for both completed groups and above 300 for PERSISTENT trajectories. Contraction along the natural direction and expansion along some other direction coexist at 62.1% of probed states (Figure 3c).

Probes restricted to Z_H or Z_L show the same pattern. At step 512, the median Z_L-probe gain is 0.023 for EARLY solvers and 0.021 for LATE solvers, compared with 0.309 for PERSISTENT trajectories. Finite-horizon growth over the 16 steps after step 256 is also negative for both completed groups and positive for PERSISTENT trajectories. Full results are reported in Appendix F.

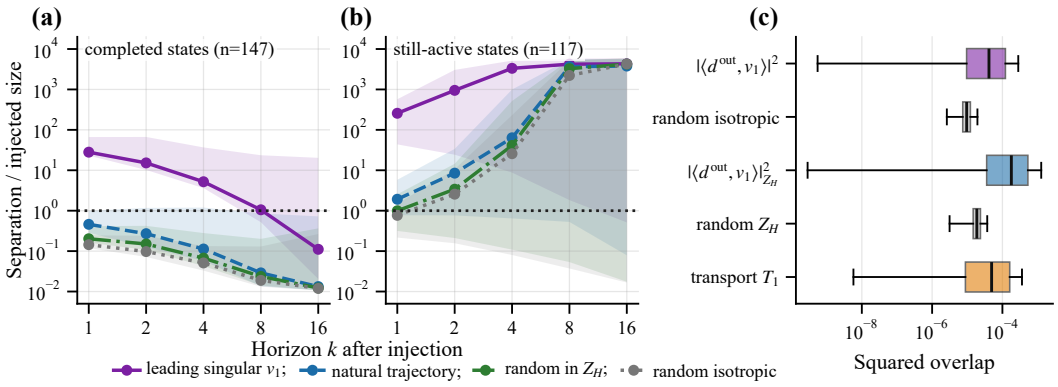

Figure 4: An expanding direction is present, but the trajectory is not orthogonal to it. Norm-matched perturbations (10^{-4} of the state norm) along four directions, followed for k further steps, at (a) states whose computation has completed and (b) states still computing. In (b) every direction, including a random one, reaches the scale of the state itself. These curves measure trajectory divergence. (c) Squared overlaps: the trajectory's overlap with v_1 is tiny but distinctly above the isotropic baseline. The transport of the leading direction across one step is of the same order. The "random isotropic" is the overlap of v_1 with a random unit direction drawn uniformly from the full hidden-state space, and "random Z_H" uses a random unit direction drawn from the Z_H subspace only.

3.4 Fate and geometry of the expanding direction

Although $\sigma_{\max} > 1$ at every probed state, this strong local expansion does not destabilize completed trajectories. We measure how the trajectory aligns with the most-expanding direction v_1 and track perturbations injected along that direction.

The trajectory does not avoid v_1. The alignment A_1 is small, with median 4.15×10^{-5} over 750 states. However, this is about four times larger than the isotropic random-direction baseline of 9.63×10^{-6}. The trajectory is therefore not orthogonal to the direction of expansion. The overlap is small because the state space has $\sim 10^5$ dimensions.

A v_1 perturbation and its later fate depend on computation completeness. A perturbation of size 10^{-4} of the state norm along v_1 grows by $44.4\times$ after one step, against $0.75\times$ along the natural direction (medians over the 180 fixed-step probe states). Paired state by state over all 264 probes, the median v_1-to-natural ratio is about $83\times$. The median ratio of one-step v_1 amplification to the estimated $\sigma_{\max}$ is 1.00. Over the 147 completed probe states (99 fixed-step and 48 solve-aligned), the median v_1 perturbation grows by about $28\times$ initially but falls to 0.11 of its injected size by $k = 16$. At states that are still computing, perturbations in all tested directions eventually produce large trajectory separation (Figure 4a,b).

The expanding direction is not carried forward. The amplified output direction $u_1(t)$ has very little overlap with the next step's most expanding input direction. The transport defined earlier is $T_1(t) \approx 4.9 \times 10^{-5}$. Thus, a direction that expands strongly in one step is largely rotated away before the next step. Strong one-step expansion, therefore, need not produce sustained multi-step growth. Non-normal recurrent dynamics can amplify a perturbation transiently without sustained growth Kerg et al. (2019), although we do not test whether this mechanism operates here. The leading input direction v_1 lies almost entirely in Z_H, while its amplified output lies mostly in Z_L. The strongest local amplification, therefore, acts mainly from Z_H into Z_L. This is also consistent with the latent supported probes in Appendix F.

3.5 Cross-architecture replication in an MLP recurrent model

We repeat the full analysis on the MLP model with the same task, data, completion definitions, diagnostic horizon, and statistics (§2.1). Only the model and checkpoints change.

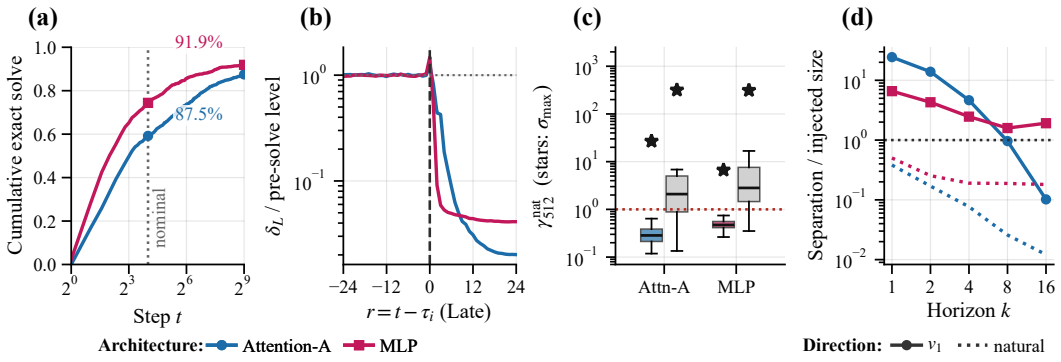

Figure 5: The completion geometry reproduces across architectures. Attention-A in blue, MLP in red. (a) Cumulative exact solve with extended recurrence, with the nominal horizon marked. (b) LATE-solver δ_L on the solve-aligned axis, each model normalized by its own pre-solve level. (c) Natural-direction gain at step 512 for completed (colored) and PERSISTENT (grey) states, with the red line at $\gamma^{\text{nat}} = 1$ and stars marking median $\sigma_{\max}$ at the same states ($\sigma_{\max} > 1$ at every probed state in both models). (d) Norm-matched perturbations at completed fixed-step states along v_1 (solid markers) and along the natural direction (dotted). Both models expand v_1 and contract the natural direction, but only Attention-A absorbs the v_1 perturbation within 16 steps.

The completion dynamics replicate. The MLP model is a stronger solver at the nominal horizon, with 74.4% exact solve at step 16 compared with 59.2% for Attention-A. Yet many of its remaining failures are also unfinished computations. By step 512, exact solve reaches 91.9%, and 175 of the 256 puzzles unresolved at step 16 are solved later (Figure 5a). The same solve-aligned settling pattern also appears. LATE solvers remain highly mobile at the nominal horizon. Their latent motion drops sharply around their own solution time, and they then enter the same low-motion regime as EARLY solvers (Figure 5b). The solving step is again the largest Z_H update of the preceding trajectory for almost every LATE solver. Full statistics are given in Appendix G.

Trajectory-conditioned stability also replicates. At step 512, the median natural-direction gain is about 0.47 for both completed MLP groups, but 2.82 for PERSISTENT trajectories. At the same time, $\sigma_{\max} > 1$ at every probed MLP state, just as in Attention-A. Completed computations are stable along their own direction of travel, while the recurrent map still contains expanding directions (Figure 5c). This coexistence occurs at 60.4% of MLP states, close to the 62.1% observed in Attention-A.

The scale of the worst-case expansion does differ. At completed states, $\sigma_{\max}$ is about 6.4-6.7 in the MLP model, compared with about 27 in Attention-A. Persistent trajectories, by contrast, have similar values in the two models. The qualitative geometry therefore replicates, while its magnitude does not.

The crucial difference lies in the expanding direction. In both models, a perturbation along v_1 expands much more strongly than one along the natural direction. The difference appears in later recurrent steps. For the 99 completed fixed-step Attention-A states, the v_1 perturbation is strongly amplified initially but falls to 0.10 of its injected magnitude by $k = 16$. In the MLP model, it remains at 1.91 times its injected size (Figure 5d). To determine whether this amplification ultimately decays or persists, the perturbation must be followed beyond the current horizon of $k = 16$. Transport shows similar differences: the overlap between the amplified output at one step and the most expanding input at the next is $T_1 = 1.83 \times 10^{-3}$ in the MLP model, compared with 4.93×10^{-5} in Attention-A. Attention-A rotates the expanding direction away after completion, whereas the MLP model carries more of it into the next step. Detailed singular-vector statistics and numerical checks are reported in Appendix G.

3.6 REPLICATION ACROSS TASKS AND CHECKPOINTS

Our main claims use only Attention-A and MLP, which were analyzed in full detail. As a checkpoint replication, we also run the complete 16-to-512 extended recurrence analysis on the Attention-B (PreetiMLresearcher, 2026) checkpoint. Attention-B shows the same trajectory-conditioned anisotropic-stability pattern: its exact-solve rate rises from 0.521 at step 16 to 0.726 at step 512,

natural-direction gain is below one in the completed groups, and expanding directions remain in the state space. All three extended-recurrence analyses are compared in Appendix G.7, Table 5.

We also tested Easy Sudoku and Maze-Hard (Appendices J and K). On Easy Sudoku, computation typically finishes within the first few steps, and the trajectories resemble EARLY solvers. The same pattern appears in Maze-Hard with 30×30 grids, where most trajectories settle within the first few steps and again resemble EARLY solvers. However, in Maze-Hard, many exact-match failures are valid alternative paths.

Finally, a constraint-aware decoder recovers many puzzles missed by simple argmax decoding (Appendix I). The TRM decoder independently fills each blank with the highest-scoring digit. To test whether useful alternatives remain in the logits, we evaluate the constraint-aware decoder on Attention-B. For each blank, it keeps the top-k predicted digits and uses backtracking to search only assignments that satisfy the Sudoku row, column, box, and clue constraints. The number of solved puzzles barely changes at $k = 1, 2, 3$, but rises by 22.3 percentage points at $k = 4$.

4 DISCUSSION

Our results show that failure at a fixed inference depth is not equivalent to computational failure. In both models, many puzzles that remain unsolved at step 16 are completed when the same recurrent computation is continued. The latent dynamics also change around completion: state updates decrease sharply, exact solutions typically persist, and the natural trajectory direction becomes locally contracting even though the full Jacobian has strongly expansive directions. Completion, therefore, does not correspond to uniform contraction of the recurrent map. Instead, contraction emerges along the direction followed by the ongoing computation, while unstable directions remain available in the state space. This qualitative picture is shared by the attention and MLP-based models. Unfinished computations remain dynamically active, whereas completed computations settle, and trajectory-aligned contraction coexists with strong off-trajectory expansion. The clearest difference between the two architectures lies in the fate of perturbations initialized along the locally most expansive direction. In the attention model, this perturbation is rapidly rotated away from the expanding direction and subsequently attenuated, whereas in the MLP model, a larger component persists over later recurrent steps. Thus, a similar trajectory-conditioned stability can coexist with different local perturbation geometries.

5 LIMITATIONS

Our results have three main limitations. First, step 512 is still finite, so we cannot determine the asymptotic fate of the PERSISTENT trajectories. Second, the full Hard-Sudoku analysis covers two attention checkpoints and one MLP checkpoint. The Maze-Hard provides a separate nominal-horizon task control, but broader claims require more checkpoints, architectures, and tasks. Finally, the Jacobian analysis is local to visited states and uses a subset of the full evaluation set. The most expensive perturbation measurements use smaller subsets.

6 CONCLUSION

A fixed inference budget can obscure the distinction between failed and unfinished computation. In the recursive models studied here, many nominal-horizon failures are solved when the same computation is allowed to continue, and completion is accompanied by a transition to a low-motion, trajectory-stable regime.

This suggests that completion is not the same as uniform contraction of the recurrent map. Instead, stability is tied to the particular path followed by the computation. Looking at these dynamics may help us understand when a recurrent model is still computing, when it has effectively settled, and how it uses its latent state during inference.

Reproducibility Statement

We provide the information needed to reproduce the main experimental results in Appendix A and in the supplementary code. These materials document the datasets, checkpoints, model, sampling, perturbation protocols, control points, and numerical validation used for the Attention-A, Attention-B, and MLP experiments. Statistical procedures are documented in Appendix B. All reported statistics and figures can be traced back to puzzle-level outputs. The supplementary code contains an analysis pipeline that regenerates them from those output archives without requiring model inference. The GPU procedure for regenerating the extended-recurrence results from the original checkpoints is also discussed in Appendix A.

AI Use Statement

Generative AI tools, including OpenAI Codex and Anthropic Claude, were used during several stages of this work. They assisted in implementing experimental and data-analysis code, generating plotting code, refining aspects of the experimental and statistical methodology (including the use and presentation of Wilson intervals, interquartile ranges, and confidence intervals), interpreting numerical results, and improving the clarity and presentation of the manuscript. AI tools also suggested additional diagnostic visualizations, including the analyses reported in Fig. 3c and Fig. 2c and the correlation between Sudoku rating and solve steps in Appendix D. The research direction, scientific questions, and final methodological and interpretive decisions were determined by the author. All AI-generated or AI-assisted code used for the reported experiments was reviewed and tested, and all reported numerical results and figures were obtained from executed computational experiments. All AI-assisted analyses were reviewed by the authors, who take full responsibility for the correctness of the results, claims, code, and final content of the paper.

References

Cem Anil, Ashwini Pokle, Kaiqu Liang, Johannes Treutlein, Yuhuai Wu, Shaojie Bai, J. Zico Kolter, and Roger B. Grosse. Path independent equilibrium models can better exploit test-time computation. In *Advances in Neural Information Processing Systems*, volume 35, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/331c41353b053683e17f7c88a797701d-Abstract-Conference.html.

Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. Deep equilibrium models, 2019. URL https://arxiv.org/abs/1909.01377.

Arpit Bansal, Avi Schwarzschild, Eitan Borgnia, Zeyad Emam, Furong Huang, Micah Goldblum, and Tom Goldstein. End-to-end algorithm synthesis with recurrent networks: Extrapolation without overthinking. In *Advances in Neural Information Processing Systems*, volume 35, 2022. doi: 10.52202/068431-1471. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/7f70331dbe58ad59d83941dfa7d975aa-Abstract-Conference.html.

Atilim Gunes Baydin, Barak A. Pearlmutter, Alexey Andreyevich Radul, and Jeffrey Mark Siskind. Automatic differentiation in machine learning: A survey. *Journal of Machine Learning Research*, 18(153):1–43, 2018. doi: 10.48550/arXiv.1502.05767.

Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Lukasz Kaiser. Universal transformers. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HyzdRiR9Y7.

Tom Dillon. tdoku: A fast Sudoku solver and benchmark suite. https://github.com/t-dillon/tdoku, 2019. Includes the `forum_hardest_1905` list collected from the Enjoy Sudoku players' forum.

B. Efron. Bootstrap methods: Another look at the jackknife. *The Annals of Statistics*, 7(1), 1979. doi: 10.1214/aos/1176344552.

Andreas Efstathiou and Aishwarya Balwani. Recursive reasoning as attractor landscape search: Mechanistic dynamics of the tiny recursive model. In *ICLR 2026 Workshop on Latent & Implicit Thinking – Going Beyond CoT Reasoning*, 2026. URL https://openreview.net/forum?id=kKps9W1K7n.

Rainer Engelken, Fred Wolf, and L. F. Abbott. Lyapunov spectra of chaotic recurrent neural networks. *Physical Review Research*, 5(4):043044, 2023. doi: 10.1103/PhysRevResearch.5.043044.

Xin Gao. TinyRecursiveModels-Sudoku-Extreme-att. Hugging Face model repository, 2025a. Accessed 2026-08-02.

Xin Gao. TinyRecursiveModel-Maze-Hard. Hugging Face model repository, 2025b. Maze-Hard checkpoint `step_9765`; accessed 2026-08-23.

Xin Gao. TinyRecursiveModels-Sudoku-Extreme-mlp. Hugging Face model repository, 2025c. MLP Sudoku checkpoint `step_16275`; accessed 2026-08-23.

Jonas Geiping, Sean McLeish, Neel Jain, John Kirchenbauer, Siddharth Singh, Brian R. Bartoldson, Bhavya Kailkhura, Abhinav Bhatele, and Tom Goldstein. Scaling up test-time compute with latent reasoning: A recurrent depth approach. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-1380. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/3b01972cf31e6fa0fe29e4b8b5c2a0a1-Abstract-Conference.html.

Angeliki Giannou, Shashank Rajput, Jy-Yong Sohn, Kangwook Lee, Jason D. Lee, and Dimitris Papailiopoulos. Looped transformers as programmable computers. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 11398–11442. PMLR, 2023. URL https://proceedings.mlr.press/v202/giannou23a.html.

Alex Graves. Adaptive computation time for recurrent neural networks, 2016. URL https://arxiv.org/abs/1603.08983.

Benhao Huang, Zhengyang Geng, and Zico Kolter. Equilibrium reasoners: Learning attractors enables scalable reasoning, 2026. URL https://arxiv.org/abs/2605.21488.

Alexia Jolicoeur-Martineau. Less is more: Recursive reasoning with tiny networks, 2025. URL https://arxiv.org/abs/2510.04871.

Giancarlo Kerg, Kyle Goyette, Maximilian Puelma Touzel, Gauthier Gidel, Eugene Vorontsov, Yoshua Bengio, and Guillaume Lajoie. Non-normal recurrent neural network (nnRNN): learning long time dependencies while improving expressivity with transient dynamics. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/hash/9d7099d87947faa8d07a272dd6954b80-Abstract.html.

Brandon Knutson, Amandin Chyba Rabeendran, Michael Ivanitskiy, Jordan Pettyjohn, Cecilia Diniz Behn, Samy Wu Fung, and Daniel McKenzie. On logical extrapolation for mazes with recurrent and implicit networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 22635–22643, 2026. doi: 10.1609/aaai.v40i27.39424. URL https://ojs.aaai.org/index.php/AAAI/article/view/39424.

Jeffrey Lai, Anthony Bao, John Quinn, and William Gilpin. Fractal basins trap latent reasoning, 2026. URL https://arxiv.org/abs/2609.04963.

Niru Maheswaranathan, Alex Williams, Matthew D. Golub, Surya Ganguli, and David Sussillo. Reverse engineering recurrent networks for sentiment classification reveals line attractor dynamics. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://arxiv.org/abs/1906.10720.

Rasmus Berg Palm, Ulrich Paquet, and Ole Winther. Recurrent relational networks. In *Advances in Neural Information Processing Systems*, 2018. URL https://arxiv.org/abs/1711.08028.

PreetiMLresearcher. TinyRecursiveModels-Sudoku. Hugging Face model repository, 2026. Twenty-five attention-model snapshots; accessed 2026-08-12.

Lucas Prieto, Melih Barsbey, Pedro A. M. Mediano, and Tolga Birdal. Grokking at the edge of numerical stability. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=TvfkSyHZRA. arXiv:2501.04697.

Zirui Ren and Ziming Liu. Are your reasoning models reasoning or guessing? a mechanistic analysis of hierarchical reasoning models, 2026. URL https://arxiv.org/abs/2601.10679.

Samsung SAIL Montreal. Tinyrecursivemodels. https://github.com/SamsungSAILMontreal/TinyRecursiveModels, 2025. Reference implementation; accessed 2026-08-02.

Avi Schwarzschild, Eitan Borgnia, Arjun Gupta, Furong Huang, Uzi Vishkin, Micah Goldblum, and Tom Goldstein. Can you learn an algorithm? Generalizing from easy to hard problems with recurrent networks. In *Advances in Neural Information Processing Systems*, volume 34, pp. 6695–6706, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/3501672ebc68a5524629080e3ef60aef-Abstract.html.

Jeffrey Seely, Yuki Imajuku, Tianyu Zhao, Edoardo Cetin, and Llion Jones. Sudoku-Bench: Evaluating creative reasoning with Sudoku variants. *arXiv preprint arXiv:2505.16135*, 2025. URL https://arxiv.org/abs/2505.16135.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. RoFormer: Enhanced transformer with rotary position embedding. *arXiv preprint arXiv:2104.09864*, 2021. URL https://arxiv.org/abs/2104.09864.

David Sussillo and Omri Barak. Opening the black box: Low-dimensional dynamics in high-dimensional recurrent neural networks. *Neural Computation*, 25(3):626–649, 2013. doi: 10.1162/NECO_a_00409.

Ivan Viakhirev, Kirill Borodin, Amirah Almutairi, Serguei Barannikov, Maxim Abramov, and Grach Mkrtchian. Think shallow, solve deep: Controlling recurrent dynamics for reliable test-time depth, 2026. URL https://arxiv.org/abs/2608.18222.

Ryan Vogt, Maximilian Puelma Touzel, Eli Shlizerman, and Guillaume Lajoie. On Lyapunov exponents for RNNs: Understanding information propagation using dynamical systems tools. *Frontiers in Applied Mathematics and Statistics*, 8:818799, 2022. doi: 10.3389/fams.2022.818799.

Guan Wang, Jin Li, Yuhao Sun, Xing Chen, Changling Liu, Yue Wu, Meng Lu, Sen Song, and Yasin Abbasi Yadkori. Hierarchical reasoning model, 2025. URL https://arxiv.org/abs/2506.21734.

Po-Wei Wang, Priya L. Donti, Bryan Wilder, and Zico Kolter. SATNet: Bridging deep learning and logical reasoning using a differentiable satisfiability solver. In *International Conference on Machine Learning*, 2019. URL https://arxiv.org/abs/1905.12149.

Edwin B. Wilson. Probable inference, the law of succession, and statistical inference. *Journal of the American Statistical Association*, 22(158):209–212, 1927. doi: 10.1080/01621459.1927.10502953.

A Reproducibility and implementation details

The extended-recurrence results use three complete analyses. **Attention-A** uses `step_21700` and took 91.5 minutes; **MLP** uses `step_16275` and took 82.1 minutes; and **Attention-B** uses `step_65100` and took 88.3 minutes. All three use the same `hard_sudoku.csv` data file and `trm.py` model source, PyTorch 2.11.0+cu128 on one CUDA device. The run uses global seed 0, the joint continuous carry $Z_H \oplus Z_L$, a nominal horizon of 16, and a diagnostic horizon of 512. Attention-A and Attention-B use PyTorch's mathematical scaled-dot-product attention backend, explicitly pinned as `SDPBackend.MATH` to prevent hardware-dependent backend selection. The forward recurrence uses bfloat16, while saved latent states and Jacobian, JVP, and VJP calculations

use float32. MLP contains no attention operation. The analysis configurations are identical apart from the model and checkpoint paths. For each analysis, we save the full experimental setup in its own `run_config.json` and `run_manifest.json`. This includes the group definitions, the 150-puzzle probe subset (50 per group), the 36-puzzle expensive subset (12 per group), and all perturbation settings. Jacobian probes use a relative perturbation of $\varepsilon = 10^{-4}$, while noise experiments use $\sigma \in \{0.02, 0.1\}$ with seeds $\{0, 1, 2\}$. We also record the use of 20,000 bootstrap replicates and five-fold cross-validation. Numerical checks, including the 16-step horizon and the singular-triplet (σ_1, v_1, u_1) identities, are in `manifests/numerical_checks.json` and `singular_geometry/table10b_numerical_checks.csv` within each archive. Every number in this paper can be traced back to a puzzle-level CSV, with the mapping given in `ICLR_RESULT_INVENTORY.md` for Attention-A and `MLP_RESULT_INVENTORY.md` for MLP. A single CPU, `python analysis/run_all.py`, recomputes every reported quantity from those puzzle-level files and regenerates all figures in about a minute, with no model inference. `REPRODUCE.md` gives the environment, the pinned package versions, and the expected outputs, and `requirements.txt` pins the analysis environment. Regenerating the results from the checkpoints needs a separate GPU path, which is also documented there. Attention-B follows a similar path. Its entries in Table 5 (Appendix G.7) come from the extended run with the math SDPA backend, which solves 521 puzzles at step 16. Appendices H and I use an earlier nominal-horizon evaluation of the same checkpoint, which solves 531. Easy Sudoku (Appendix J), Maze-Hard (Appendix K), and the constraint decoder (Appendix I) come from separate 16-step evaluations.

B Statistical procedures

Statistical procedures. We report binary proportions with Wilson 95% intervals (Wilson, 1927). Continuous per-puzzle quantities use a percentile bootstrap (Efron, 1979) over puzzles with 20,000 replicates. Every puzzle is an independent sampling unit. Repeated recurrent steps, probe directions, perturbation horizons, and noise seeds are treated as repeated measurements of the same puzzle rather than as independent samples. For solve-aligned analyses, we preserve the pairing between the pre- and post-solve windows by resampling whole puzzles. Jacobian analyses use a fixed subset of 150 puzzles, with 50 puzzles from each completion group. The more expensive direct-perturbation and transport analysis uses 12 puzzles per group. Perturbations injected at step 512 are followed for up to 16 further steps, beyond the 512-step run. Excluding these states moves the Attention-A median v_1 separation at $k = 16$ from 0.111 to 0.115 over all completed states and from 0.102 to 0.108 over fixed-step states, and leaves the MLP value of 1.91 unchanged.

Numerical precision. The standard forward pass uses bfloat16, matching the released model checkpoint. For the Jacobian analysis, we upcast the recurrent computation to float32, disable TF32, and use the mathematical SDPA attention backend. In a matched float32 control, the differentiable recurrence exactly reproduces the reference float32 forward pass, with a maximum absolute difference of 0.

Terminology. We use several dynamical terms in a limited sense. "Settling" means that the observed relative state updates become small. It does not imply that the map is contractive. "Directional gain" is the one-step Jacobian gain along a specified direction. "Finite-horizon growth" describes amplification over a fixed number of future recurrent steps. "Completion" means the first exact decode. A small fraction of trajectories later revert, as quantified in §3.2. A "block" probe is restricted to Z_L or to Z_H.

C TRM Setup

The attention-based TRM configuration is summarized in Table 2. Jolicoeur-Martineau (2025) gives the complete architecture and pseudocode (Figures 1 and 3 of that paper).

The loss in the above table is StableMax cross entropy loss (Prieto et al., 2025). The input sequence contains 81 grid-cell tokens prefixed by 16 puzzle-embedding tokens, giving 97 token positions. The recurrent carry contains two latent states, z_H and z_L, each of shape 97×512. Hence the joint continuous carry has dimension

$$\dim(z_H, z_L) = 2 \times 97 \times 512 = 99{,}328.$$

Table 2: Configuration of the attention-based TRM checkpoints.

Configuration	Value
`H_cycles`	3
`L_cycles`	6
`L_layers`	2
`H_layers`	0
Hidden size	512
Attention heads	8
MLP expansion factor	4
Position encoding	Rotary
`puzzle_emb_len`	16
`puzzle_emb_ndim`	512
`halt_max_steps`	16
`no_ACT_continue`	`true`
Forward dtype	`bfloat16`
Loss	StableMax cross-entropy via `ACTLossHead`
Number of parameters	6,829,570

In evaluation mode, ACT halts at the final permitted step and resets the carry on the subsequent call. Consequently, obtaining an uninterrupted N-step trajectory requires setting the halt cap to N. For the long-horizon evaluations in this work, we therefore set `halt_max_steps`=512; see §2.1.

D Extended-recurrence results

Solve rate

The cumulative solve rate is reported in §3.1. We distinguish it from the instantaneous solve rate. A puzzle enters the cumulative count as soon as it is solved exactly for the first time. The instantaneous count instead asks whether it is solved at that particular step. These two counts are identical at all reported checkpoints except step 32. By then, 660 puzzles had been solved at least once, while 659 were correct at step 32. One puzzle had briefly reverted after being solved. Since our completion groups are defined by the first exact solve, all group assignments use the cumulative count.

Puzzle difficulty by completion group

Datasets. Hard Sudoku: 1,000 puzzles from the test split of Sudoku-Extreme (Wang et al., 2025) (`https://huggingface.co/datasets/sapientinc/sudoku-extreme`), all from its `puzzles4_forum_hardest_1905` source, which is the `forum_hardest_1905` list of the `tdoku` benchmark suite (Dillon, 2019). Sudoku-Extreme permutes every puzzle by row, column, box, and digit and keeps its train and test splits mathematically inequivalent. The dataset carries a per-puzzle `rating` defined by Sudoku-Extreme as the number of backtracks the `tdoku` solver (Dillon, 2019) needs to solve the puzzle. Table 3 reports the ratings by completion group.

Table 3: Puzzle difficulty by completion group on the dataset's own rating scale. Difference rows report unpaired median differences relative to Early solvers, with 20,000 bootstrap replicates over puzzles in each group.

Group	n	min	Q1	median	Q3	max	mean	median 95% CI
Early solver	592	2	14.0	27	42	140	31.23	[25.0, 28.5]
Late solver	283	2	11.5	24	39	110	27.63	[20.0, 27.0]
Persistent unsolved	125	3	13.0	20	28	127	24.05	[18.0, 23.0]
All puzzles	1000	2	13.0	25	40	140	29.31	[23.0, 26.0]
Unpaired median difference against Early								
Late − Early	283			−3				[−7.0, 1.0]
Persistent − Early	125			−7				[−9.5, −3.0]

The puzzles the model never solves are rated *easier* than the ones it solves inside the nominal budget. Two rank statistics say the same thing. Among the 875 eventual solvers, Spearman correlation between rating and first solve time is $\rho = -0.097$ ($p = 0.004$): higher-rated puzzles are solved marginally *earlier*. Across all 1,000 puzzles, the correlation between rating and ever solving is $\rho = +0.095$ ($p = 0.003$). Binned by rating quintile, the PERSISTENT fraction runs 0.112, 0.200, 0.167, 0.066, 0.075 from the lowest to the highest quintile, with no monotone trend. Difficulty therefore does not explain the completion groups.

E JACOBIAN IMPLEMENTATION

Our Jacobian analysis requires differentiating through one outer recurrence of the model. We therefore use a differentiable version of the recurrence in which the `detach` and `no grad` boundaries are removed. We verified that this change does not alter the forward computation. After converting the relevant operations to float32, the differentiable and original implementations agree exactly at the level of the model state.

We estimate the leading singular value of the Jacobian, σ_1, using matrix-free power iteration on $J^\top J$. This avoids constructing the full Jacobian, which would be prohibitively large for our state space. Jacobian-vector and vector-Jacobian products are computed with `jvp` and `vjp`, respectively (Baydin et al., 2018). We use 10 power iterations with a convergence tolerance of 10^{-3}. Once the leading right singular vector v_1 is obtained, the corresponding left singular vector is defined as

$$u_1 = \frac{Jv_1}{\sigma_1}. \tag{12}$$

Because the quantities of interest are sensitive to small numerical errors, we perform these calculations in float32 and disable TF32. We also use the mathematical SDPA backend for attention rather than the fused Flash, memory-efficient, or cuDNN SDPA backends.

We carried out numerical checks on the estimated singular triplets. Across 1,100 probe entries, the singular vectors remain normalized to numerical precision, and 98.6% of the triplets satisfy the scalar convergence criterion based on relative changes in σ_1. The relation $Jv_1 = \sigma_1 u_1$ holds by construction. The adjoint norm ratio $\|J^\top u_1\|_2/\sigma_1$ remains within 1.3% of one across these entries.

Taken together, these checks suggest that the matrix-free procedure is sufficiently stable for the Jacobian analyses reported in the paper.

F LATENT BLOCK PERTURBATIONS AND FINITE-HORIZON GROWTH

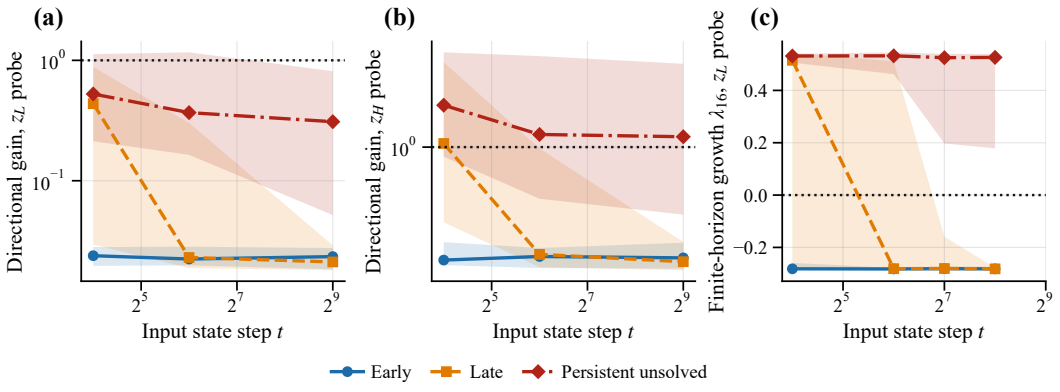

Figure 6: (a,b) Median one-step directional gain for probe directions supported on Z_L and Z_H, shown by group and input-state step. (c) Finite-horizon growth λ_{16} following a perturbation to Z_L. Shaded regions show interquartile ranges.

To probe stability beyond one step, we measure the finite-horizon separation of a clean and a perturbed trajectory. For each probed state s_t, we draw a isotropic unit direction v_L supported entirely in Z_L

and set

$$v = (0, v_L), \qquad \widetilde{s}_t = s_t + \varepsilon_t v, \qquad \varepsilon_t = 10^{-4}\|s_t\|_2.$$

After evolving both states under the same recurrent map for k steps, let

$$\Delta s^{(v)}_{t+k} = \widetilde{s}_{t+k} - s_{t+k}.$$

We define

$$\lambda_k(v) = \frac{1}{k} \log \left(\frac{\|\Delta s^{(v)}_{t+k}\|_2}{\varepsilon_t} \right).$$

Negative values mean that the separation at horizon k is smaller than the injected perturbation, while positive values mean that it is larger. This is a finite-horizon, trajectory-conditioned measure of perturbation growth, not an asymptotic Lyapunov exponent.

Latent block-supported probes show how the response differs between the two latent components (Figure 6 a,b). At step 16, the median gain for EARLY trajectories is below one in both blocks. The separation from PERSISTENT trajectories is more pronounced in Z_H, where the median gains are 0.190 for EARLY and 1.851 for PERSISTENT trajectories.

By step 512, the EARLY and LATE groups have similar responses in both latent blocks. PERSISTENT trajectories retain larger gains in Z_H with a median greater than one.

Finite-horizon growth shows the same distinction (Figure 6 c). For Z_L perturbations introduced at step 256, the median growth over the next 16 steps is $\lambda_{16} = -0.282$ in both the EARLY and LATE groups and 0.526 in the PERSISTENT group.

These quantities describe growth along the observed recurrent trajectory over a fixed number of future steps. They should therefore be interpreted as finite-horizon, trajectory-conditioned rates rather than asymptotic Lyapunov exponents (Vogt et al., 2022; Engelken et al., 2023).

G FULL MLP MODEL CROSS-ARCHITECTURE REPLICATION

In this appendix, we study the MLP experiments in full detail. Relevant quantities are defined in §2.1 and computed on the same task and data.

G.1 CHECKPOINT AND CONFIGURATION FOR MLP

The configuration of the MLP model used in our experiments is summarized in Table 4. Except for the token-mixing mechanism and positional encoding, its architectural and evaluation settings are identical to those of Attention-A. Both runs use the same puzzles file and the same model source.

The MLP replaces attention-based token mixing with a sequence-axis MLP and omits rotary positional encoding. The remaining architectural settings match Attention-A as shown in Table 4.

The analysis uses global seed 0 on a CUDA device. Over all 512,000 recorded step observations, no non-finite values occur, and both latent-state norms remain bounded. After conversion to float32, the differentiable recurrence also matches the original forward computation exactly.

G.2 EXTENDED RECURRENCE AND COMPLETION GROUPS

Extending the recurrence from 16 to 512 steps raises the number of puzzles solved at least once from 744 to 919 (Figure 7a). Moreover, blank-cell accuracy improves while StableMax loss decreases.

Of the 1,000 puzzles, 744 belong to the EARLY group $[71.6\%, 77.0\%]$, 175 to the LATE group $[15.3\%, 20.0\%]$, and 81 remain PERSISTENT$[6.6\%, 10.0\%]$. Among the puzzles that eventually solve, the median first solve time is 3 steps for EARLY trajectories (IQR 2-6) and 43 steps for LATE trajectories (IQR 28-100).

All 919 MLP puzzles that ever reach an exact solution keep it: $P(\text{stays solved}) = 1.000$ $[0.9958, 1.0]$, with zero reversions, against $859/875 = 0.982$ in Attention-A.

Table 4: Configuration of the MLP model used in the experiments.

Configuration	Value
Checkpoint	`step_16275`
Number of parameters	5,030,402
Attention-A parameters	6,829,570
Token mixing	Sequence-axis MLP
`mlp_t`	`true`
Position encoding	None
`H_cycles`	3
`L_cycles`	6
`L_layers`	2
Hidden size	512
MLP expansion factor	4
Forward dtype	`bfloat16`
Loss	StableMax cross-entropy
Nominal recurrence length	16 steps

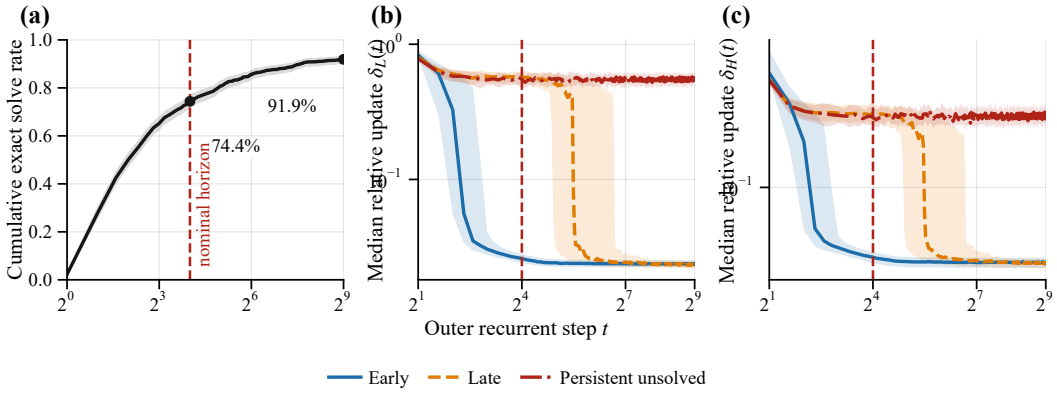

Figure 7: MLP extended recurrence. (a) Cumulative exact solve with Wilson bands. (b,c) Median relative latent update by completion group with interquartile bands.

The latent dynamics yield the same ordering as in Attention-A (Figure 7b,c). EARLY trajectories already have small updates at step 16 while LATE and PERSISTENT trajectories remain more active. At step 512, the LATE group reaches a similar low-motion regime to the EARLY group, whereas PERSISTENT trajectories continue to change substantially.

The difficulty rating is only weakly associated with these outcomes. Among the 919 eventual solvers, higher ratings are associated with slightly earlier first solutions $(\rho, p) = (-0.105, 0.0014)$. The association with whether a puzzle ever solves is small $(\rho = 0.025, p = 0.43)$. These results do not support a simple interpretation in which later or unsuccessful trajectories are harder puzzles on this rating scale.

G.3 SOLVE-ALIGNED SETTLING AND THE SOLVING STEP

The MLP shows a sharp reduction in latent motion around completion (Figure 8a,b). Among the 175 LATE solvers, the median paired change in window-mean δ_L between the pre-solve window $[-8, -1]$ and post-solve window $[0, 7]$ is -0.399 $[-0.408, -0.391]$. The corresponding change in δ_H is -0.138 $[-0.147, -0.132]$ compared with -0.029 in Attention-A.

The earlier comparison between $[-16, -9]$ and $[-8, -1]$ shows little change in δ_L: 0.006 $[-0.007, 0.014]$. The reduction is concentrated around completion rather than spread across the preceding windows. After solving, LATE trajectories approach a low-motion regime similar to EARLY trajectories.

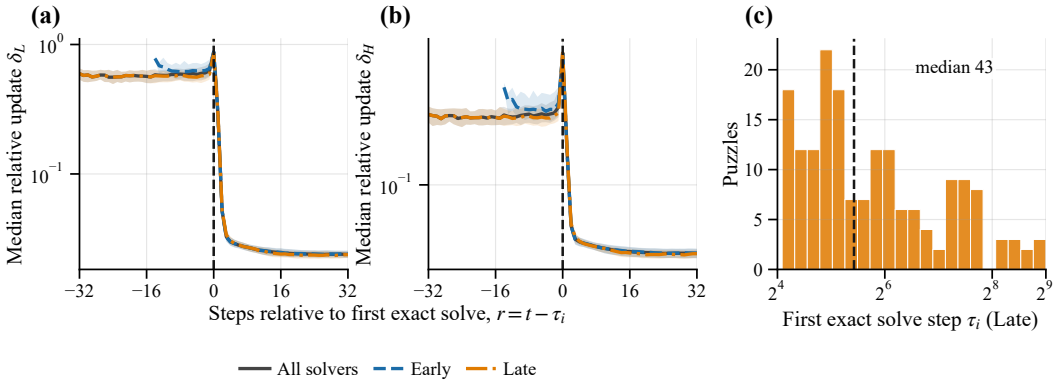

Figure 8: MLP solve-aligned dynamics. (a,b) Relative updates along the solve-aligned axis. (c) Distribution of first solve times for LATE solvers.

The solving step itself contains a pronounced state update. The paired increase in δ_H from $r = -1$ to $r = 0$ is 0.508 [0.498, 0.523]. For 99.4% of LATE solvers, the Z_H update at the solving step is larger than at every earlier step. The solving step Z_L update is the largest observed up to that step for 74.3% of LATE MLP trajectories, compared with 35.3% in Attention-A.

G.4 FIXED-TIME JACOBIAN, NATURAL-DIRECTION GAIN, AND $\sigma_{\max}$

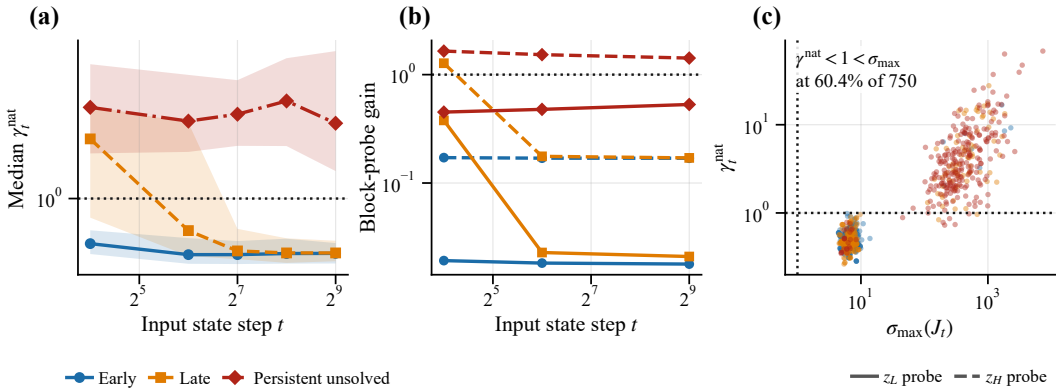

Figure 9: MLP local stability at fixed recurrent steps. (a) Median natural-direction gain by group. (b) Probe gains supported on Z_L (solid) and Z_H (dashed). (c) Natural-direction gain γ_t^{nat} versus $\sigma_{\max}(J_t)$ for all probed states.

The natural-direction gain already separates the three groups at the nominal 16-step horizon (Figure 9). Median γ^{nat} is 0.538 [0.494, 0.607] for EARLY, 2.271 [1.347, 2.858] for LATE, and 3.505 [2.450, 4.635] for PERSISTENT trajectories. Accordingly, 92% of EARLY states have gain below one, compared with 32% of LATE states and 12% of PERSISTENT states.

By step 512, median natural-direction gains are similar in the EARLY and LATE groups (0.470 and 0.473) while remaining above one in the PERSISTENT group (2.816). Nevertheless, $\sigma_{\max} > 1$ at all 750 probed states. Small gains along the natural direction coexist with expanding directions in the surrounding state space (Figure 9c).

Latent block-supported probes show the same group ordering. The EARLY and LATE groups have smaller median gains than the PERSISTENT group in both latent components (Figure 9b).

The finite-horizon measurements extend this comparison beyond one step. For Z_L perturbations introduced at step 256, growth over the following 16 steps is negative in 98% of EARLY states, 90% of LATE states and 12% of PERSISTENT states.

G.5 LEADING SINGULAR-VECTOR GEOMETRY

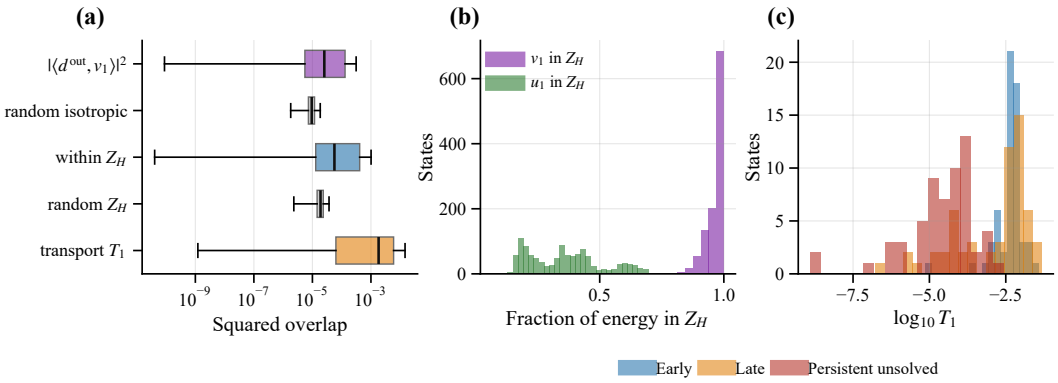

Figure 10: MLP leading singular-vector geometry. (a) Squared overlaps and one-step transport for the fixed-time states, together with random baselines. The "random isotropic" is the overlap of v_1 with a random unit direction drawn uniformly from the full hidden-state space, and "random Z_H" uses a random unit direction drawn from the Z_H subspace only. Transport is evaluated on the 180-state subset. (b) Fraction of the leading right and left singular vectors supported on Z_L and Z_H. (c) Distribution of $\log_{10} T_1$ across completion groups.

The natural update has a small overlap with v_1, but it exceeds the random isotropic baseline (Figure 10a). The median squared overlap is 2.55×10^{-5} compared with 9.46×10^{-6} for the random baseline. The paired difference is 1.61×10^{-5} $[1.33 \times 10^{-5}, 2.19 \times 10^{-5}]$. The Z_H-restricted comparison shows the same pattern.

In both architectures, the leading input direction v_1 lies almost entirely in Z_H, while its amplified output u_1 lies mainly in Z_L (Figure 10b). The Z_L component contains 65% of the output-direction energy in the MLP compared with 85.3% in Attention-A. The leading response is less concentrated in Z_L for the MLP.

The leading direction is transported more strongly between successive steps in the MLP (Figure 10c). Over the 180-state subset, median T_1 is 1.83×10^{-3} $[5.7 \times 10^{-4}, 4.6 \times 10^{-3}]$, compared with 4.93×10^{-5} in Attention-A.

G.6 DIRECT v_1 VERSUS NATURAL-DIRECTION PERTURBATIONS

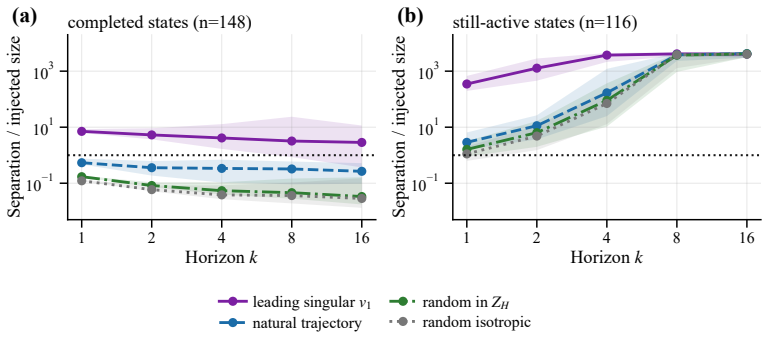

Figure 11: MLP response to norm-matched perturbations with initial size 10^{-4} of the state norm. Results are separated according to whether the unperturbed computation has already completed. Panel (a) pools the 100 fixed-step and 48 solve-aligned completed states (the text quotes fixed-step medians), and panel (b) shows the 116 still-active states. Shaded regions show interquartile ranges.

At the 100 completed fixed-step states, perturbations along v_1 initially expand, with a median one-step amplification of 6.58 [6.43, 6.96] (Figure 11a). After 16 steps, the median separation is 1.91 [0.87, 2.74] times the injected size. Although the point estimate remains greater than one, its confidence interval spans one, and 42% of states have separations less than the injected size. Recovery

also varies between completion groups: the corresponding medians are 0.91 for EARLY states and 3.10 for LATE states.

Natural-direction perturbations contract more consistently. After 16 steps, their median separation is 0.181 $[0.113, 0.234]$ times the injected size, with 93% of states below one. Both random controls contract more strongly still.

At the 116 still-active probe states, all four perturbation directions produce separations of approximately 4×10^3 times the injected size by step 16. Given the initial scale of 10^{-4} of the state norm, these separations are of the order of the state norm (Figure 11b).

These results distinguish initial amplification from subsequent recovery. The less consistent recovery of v_1 perturbations in the MLP is compatible with its larger measured direction transport.

G.7 CROSS-CHECKPOINT AND ARCHITECTURE SUMMARY

Table 5: Cross-checkpoint and cross-architecture comparison. Completion-group quantities use step 512. γ^{nat} and $\sigma_{\max}$ are medians over 50 probed states per group. Perturbation separations are medians over completed fixed-step probed states. A range in a completed entry gives the Early-Late group medians.

Quantity	Attention-A	MLP	Attention-B
Step-16 / step-512 exact solve	0.592 / 0.875	0.744 / 0.919	0.521 / 0.726
Step-16 unresolved with any later exact solve	283/408 = 0.694	175/256 = 0.684	206/479 = 0.430
Solutions never lost after first solve	859/875 = 0.982	919/919 = 1.000	719/727 = 0.989
γ^{nat}, completed / PERSISTENT	0.25-0.31 / 2.09	0.47 / 2.82	0.20-0.21 / 2.16
$\sigma_{\max}$, completed / PERSISTENT	26.8-26.9 / 316.9	6.4-6.7 / 312.4	17.6-19.0 / 283.9
$\Pr[\gamma^{\mathrm{nat}} < 1 < \sigma_{\max}]$	0.621	0.604	0.624
v_1 perturbation, k=1 / k=16	24.4 / 0.10	6.6 / 1.91	21.1 / 0.096
Natural perturbation, k=1 / k=16	0.38 / 0.012	0.50 / 0.18	0.23 / 0.0070
Transport T_1	4.9×10^{-5}	1.8×10^{-3}	1.4×10^{-4}

Table 5 summarizes the shared findings and differences between checkpoints. Extended recurrence resolves many nominal-budget failures. In Attention-A and the MLP, LATE trajectories settle after solving and approach the low-motion regime of EARLY trajectories. Completed groups typically have the natural-direction gains below one even though expanding directions remain in the local Jacobian.

The checkpoints differ in solve rates, the size of the completion transition, and the response to perturbations. In particular, the MLP has greater transport of the leading expanding direction and less consistent recovery of perturbations along that direction over 16 steps.

H ATTENTION-B TRAINING SNAPSHOTS

This section reports the separate nominal-horizon sweep over all 25 released Attention-B training snapshots (PreetiMLresearcher, 2026). The full extended analysis of the terminal checkpoint is reported in Appendix G.7, Table 5. The snapshots span training steps 2,604 to 65,100. Most are spaced by 2,604 training steps, except that the third snapshot is at step 6,510.

Nominal-horizon exact-solve performance improves overall across training, reaching 53.1% at the final checkpoint, with several temporary declines (Figure 12 a). Conflict severity among unsolved puzzles initially decreases and later rises slightly (Figure 12 b), although the set of unsolved puzzles changes between snapshots.

H.1 COMPARISON BETWEEN CHECKPOINTS AT THE NOMINAL HORIZON

The 25-snapshot sweep provides a training-time view of the outcome-associated trends seen in Attention-A. The terminal Attention-B checkpoint also provides an independent full-run replication of the same attention architecture, summarized in Appendix G.7.

The three Sudoku checkpoints are as follows. **Attention-A** (`step_21700`) is the checkpoint studied in the main text (Gao, 2025a). **Attention-B** (`step_65100`) comes from a separate public training run

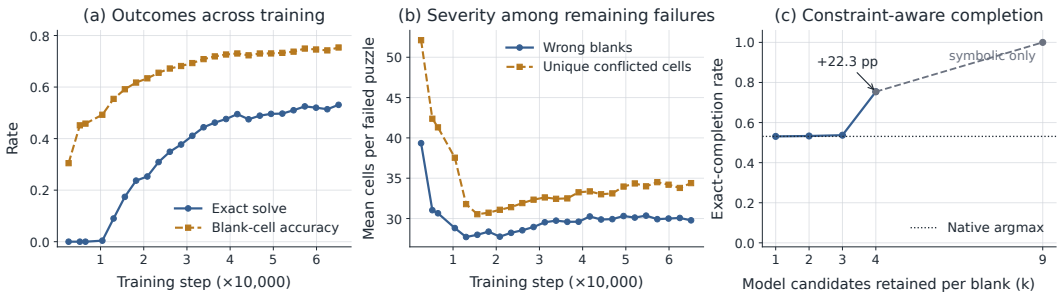

Figure 12: Attention-B training and decoder diagnostics. (a) Exact-solve and blank-cell accuracy across 25 correlated training snapshots. (b) Conflict severity. (c) Exact completion under model-guided Sudoku constraints; the dotted line is native argmax and $k = 9$ is a symbolic-only control.

of the same architecture (PreetiMLresearcher, 2026). **MLP** (`step_16275`) replaces attention-based token mixing with an MLP along the sequence axis and uses no positional encoding (Gao, 2025c).

All three models use hidden width 512, three H-cycles, six L-cycles, two layers in the repeated L-level module, and bfloat16 forward computation. The two attention models use eight attention heads and rotary positional encodings (Su et al., 2021).

Table 6: Nominal 16-step evaluation on 1,000 hard Sudoku puzzles. Final Z_L updates are reported as medians within the solved and failed groups. "Factor" is the ratio of the failed-group median to the solved-group median.

Checkpoint	Exact solve	Final Z_L update, solved / failed	Factor
Attention-A	0.592	0.025 / 0.761	∼30
Attention-B	0.531	0.069 / 0.708	10.2
MLP	0.744	0.018 / 0.536	∼30

The MLP achieves the highest nominal solve rate followed by Attention-A and Attention-B. All three checkpoints have larger median final Z_L updates on failed puzzles than on solved puzzles (Table 6).

I Constraint-aware decoding

The TRM decoder fills each blank with the highest-scoring digit independently. Related neural Sudoku methods incorporate constraint structure through recurrent message passing (Palm et al., 2018) or differentiable satisfiability layers (Wang et al., 2019). To test whether useful alternatives remain in the logits, we also evaluate a constrained decoder on Attention-B. For each blank, it keeps the top-k predicted digits and uses backtracking to search only assignments that satisfy the Sudoku row, column, box, and clue constraints. The search is capped at 200,000 nodes per puzzle (Table 7).

Table 7: Constraint decoding for Attention-B on 1,000 puzzles. "Supported" means that the correct digit for every blank appears in that blank's top-k predictions. The $k = 9$ setting removes model-based pruning and serves only as a symbolic-solver control.

k	Supported	Exact	Median nodes	95th pct. nodes
1	0.531	0.531	56	59
2	0.533	0.533	56	59
3	0.537	0.537	59	659
4	0.754	0.754	59	9,304
9	1.000	1.000	59	16,049

All 469 failures of the native decoder are complete boards that violate Sudoku constraints. Allowing two or three candidate digits per blank changes the solve rate slightly. With four candidates, the constrained decoder finds 754 exact solutions, recovering 223 cases missed by native argmax decoding (Table 7).

These recoveries show that useful alternatives remain in the logits. A correct completion can be found within the top-four candidate sets when Sudoku constraints are enforced by backtracking. The choice of $k = 4$ was made using the same 1,000 evaluation puzzles.

J EASY SUDOKU

The Easy Sudoku control uses the 100 puzzles in the `nikoli_100` set of Sudoku-Bench (Seely et al., 2025) (https://huggingface.co/datasets/SakanaAI/sudoku-bench-nikoli). These puzzles contain 46-58 blanks compared with 55-60 in the hard split.

Both Attention-A and the MLP solve all 100 puzzles within the nominal 16 steps. At the first step, Attention-A solves 21 puzzles and the MLP solves 66. The final latent updates are small in both models, although their magnitudes differ (Table 8). These updates are comparable to those of Hard Sudoku puzzles after they are solved, for both EARLY and LATE solvers.

Table 8: Easy Sudoku control, $n = 100$, at the nominal 16-step horizon.

Quantity	Attention-A	MLP
Exact-solve rate	1.000	1.000
Solved at step 1	21	66
Steps to 99% of final mean blank-cell accuracy	6	2
Z_L final update	0.017	0.024
Z_H final update	0.008	0.034

K MAZE-HARD

Maze-Hard provides a cross-task control using 1,000 30×30 mazes from the benchmark of Wang et al. (2025) (https://huggingface.co/datasets/sapientinc/maze-30x30-hard-1k). Each cell holds one of six symbols. We evaluate an attention checkpoint (`step_9765`; Gao, 2025b) with three H-cycles and four L-cycles.

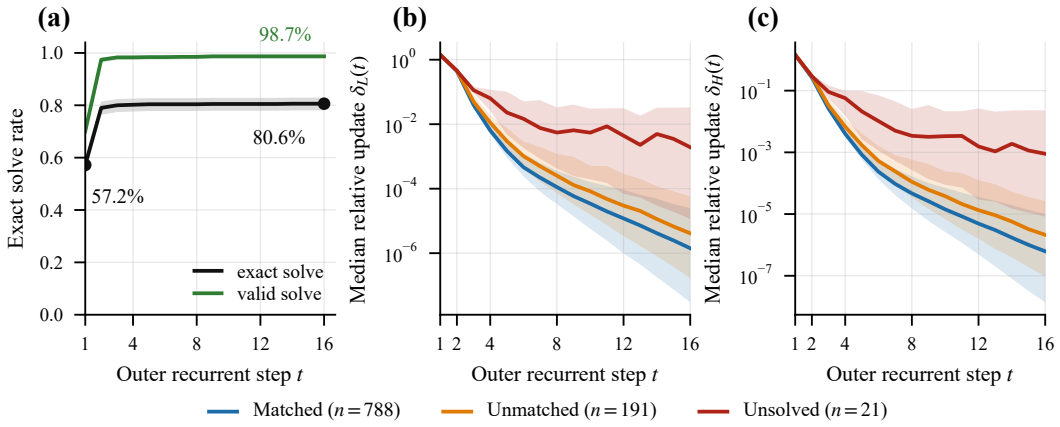

Figure 13: Maze-Hard outcomes and latent settling. (a) Cumulative exact solves from bfloat16 run, with Wilson bands. Figures (b,c) Median relative latent updates in float64. Shaded bands show interquartile ranges.

Most solutions that match the ground truth appear early, with 800 mazes matching the ground truth at step 3 (790 at step 2). Over the entire 16-step evaluation, 806 mazes match the ground truth at least once, but only 788 do so at the final step because 18 earlier solutions are subsequently lost. This 16-step evaluation does not determine whether the unresolved cases would solve if recurrence were extended beyond step 16.

Exact match with the ground truth also understates path validity. Of the 212 final non-matches, 191 are valid start-to-goal paths, 75 of them as short as the ground truth, and only 21 are structurally invalid.

Table 9: Maze-Hard results, $n = 1{,}000$, at the nominal 16-step horizon. Intervals are Wilson 95%.

Metric	Value
Exact-match rate	0.788 [0.762, 0.812]
Valid-path rate	0.979 [0.968, 0.986]
Valid but not exact	191
equal path length to reference	75
longer (mean +2.1 cells)	116
Structurally invalid paths	21
Exactly correct earlier, lost by step 16	18

Table 10: Maze outcomes and latent motion at step 16. Groups, mean all-cell accuracy, and mean path F1 come from the bfloat16 run. Median relative latent updates δ_L and δ_H come from the float64 evaluation.

Group	n	Acc.	Path F1	δ_L	δ_H
MATCHED	788	1.000	1.000	1.4×10^{-6}	6.2×10^{-7}
UNMATCHED	191	0.968	0.871	4.1×10^{-6}	2.1×10^{-6}
UNSOLVED	21	0.975	0.901	1.9×10^{-3}	9.0×10^{-4}

LATENT SETTLING

We categorize the puzzle in three groups: MATCHED, UNMATCHED, and UNSOLVED. MATCHED means the model solved the puzzle and it matches the ground truth ($n = 788$). UNMATCHED means the solution provided by the model is a valid path but doesn't match the ground truth ($n = 191$). UNSOLVED means the model failed to provide the structurally valid solution to the puzzle ($n = 21$).

At this scale, the measurements cannot reliably resolve differences in settling. We therefore replay the recurrence from the checkpoint's initial state in float64. We keep the groups and solve times fixed, so that the control does not also change the outcome definition.

Accuracy: Acc. is all-cell accuracy: the fraction of the 900 maze cells whose predicted label matches the reference. It includes walls, open cells, endpoints, and path cells. Many unchanged background cells can make this high even when the route is wrong. Path F1 measures overlap between predicted and reference path-marked cells:

$$\text{Path F1} = \frac{2|P \cap R|}{|P| + |R|}, \tag{13}$$

where P and R are the predicted and reference path-cell sets. Start and goal cells have separate labels and are excluded from these sets. The table (10) reports the mean per-puzzle score within each group at step 16, using the BF16 predictions. Importantly, Path F1 is not path validity. A valid alternative route can score below 1 because it differs from the reference. An invalid route can score highly because most of its cells overlap the reference. That explains why the Unsolved group can have higher accuracy and Path F1 than the Unmatched group. The change in latents Z_L and Z_H for these categories are also reported in table (10) and plotted in fig. (13 b, c).

Local stability: Small latent updates alone do not establish local stability. We investigate whether a small displacement along the trajectory is contracted by the next recurrent step. Let $s_t = (Z_{H,t}, Z_{L,t})$ be the joint latent state, with the puzzle held fixed

$$J_t = DF_\theta(s_t), \qquad d_t^{\text{out}} = \frac{s_{t+1} - s_t}{\|s_{t+1} - s_t\|_2}, \qquad \gamma_t^{\text{nat}} = \|J_t d_t^{\text{out}}\|_2. \tag{14}$$

A gain $\gamma_t^{\text{nat}} < 1$ indicates local contraction along that direction. We compare it with $\sigma_{\max}(J_t)$, the largest gain over all directions. The latter is estimated by power iteration. For contractions, see table 11 and fig. 14 a, b for more information.

A larger gain can accompany smaller motion. For MATCHED puzzles, median gain rises from 0.15 at $t = 2$ to 0.61 at $t = 15$ (Table 11). Over the same steps, the full-group median δ_L falls from 4.4×10^{-1} to 2.5×10^{-6}. There is no contradiction. Gain is a local amplification factor, whereas

Table 11: Local stability by input-state step. γ^{nat} measures gain along the trajectory. $\sigma_{\max}$ estimates the largest gain over all directions. Entries are medians with IQR ranges. n_γ counts valid directions. Each measured $\sigma_{\max}$ entry uses all 50, 50, or 21 states.

Step	Group	n_γ	γ^{nat}	$\Pr[\gamma^{\mathrm{nat}} < 1]$	$\sigma_{\max}$
2	MATCHED	50	0.15 [0.14, 0.21]	1.00	12.5 [9.2, 19.9]
	UNMATCHED	50	0.18 [0.14, 0.25]	0.96	21.5 [12.7, 28.9]
	UNSOLVED	21	0.51 [0.32, 3.12]	0.62	66.3 [38.5, 223.0]
7	MATCHED	50	0.53 [0.30, 0.76]	0.92	11.3 [8.8, 17.7]
	UNMATCHED	50	0.54 [0.36, 0.77]	0.82	16.1 [11.7, 25.1]
	UNSOLVED	21	0.79 [0.51, 1.18]	0.67	74.3 [27.0, 114.9]
15	MATCHED	49	0.61 [0.38, 0.77]	0.96	11.7 [8.9, 17.5]
	UNMATCHED	47	0.59 [0.43, 0.76]	0.91	15.5 [12.1, 23.1]
	UNSOLVED	21	0.75 [0.48, 1.27]	0.71	68.8 [27.0, 99.7]

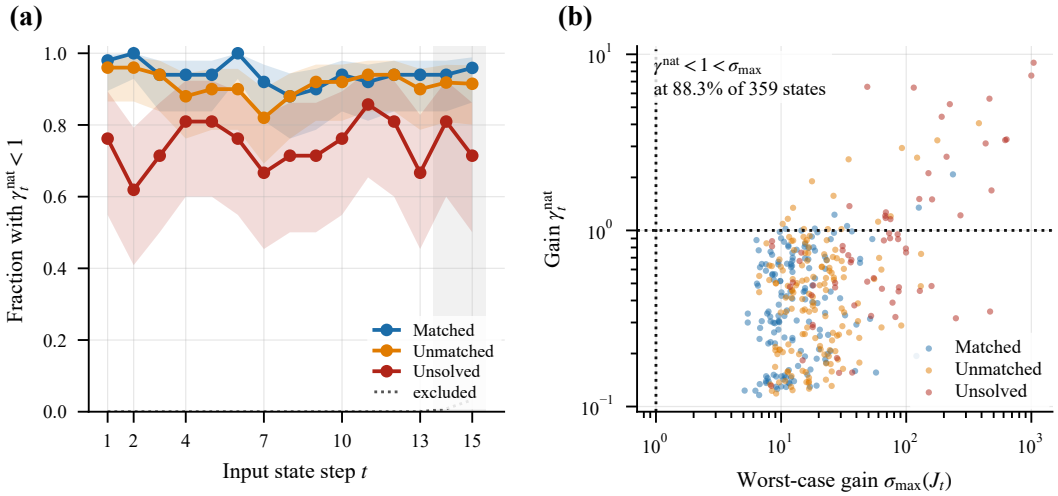

Figure 14: Directional contraction and worst-case expansion. (a) Fraction of valid directions with $\gamma_t^{\mathrm{nat}} < 1$, with Wilson bands. (b) States with both a valid directional gain and a singular-value estimate, at $t = 2, 7, 15$. The inequalities $\gamma_t^{\mathrm{nat}} < 1 < \sigma_{\max}(J_t)$ hold together at $317/359$ states (88.3%).

the update measures how far the state moves. These two also uses different populations. The update is a blockwise (either Z_L or Z_H) quantity over all 788 MATCHED puzzles. The gain uses selected joint-state probes.

Stability is directional, not global. The contraction along the trajectory coexists with expansion in another direction at $317/359$ states with both measurements (88.3%; Figure 14). The Wilson 95% interval is 84.6%–91.2%.