ROTATE THE LOOP: BLOCKWISE CONFORMALITY FOR LOOPED TRANSFORMERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Looped transformers reuse one shared block across many recurrent steps, so depth becomes a quantity that can be spent at runtime rather than a stack of distinct layers. Those extra steps help only if a write made on one recurrent step is still readable on later ones, and that depends on the linear map applied to the residual between recurrent steps. We show that whether the loop learns to track state is a geometric question about the form of this map, and we propose **blockwise conformal** injections: the residual splits into coordinate-aligned groups and rotates inside each one, so every group has its own timescale while the geometry stays exact within it. Our experiments show a performance increase on a demanding form of state tracking - last-token accuracy when the rest of the prefix is already correct - and on language modelling, where a signal written earlier has to be retrieved at the end of the sequence.

1 INTRODUCTION

Transformers compute in parallel across a sequence, and that parallelism costs them depth: with a fixed number of layers they cannot, under standard complexity assumptions, recognize regular languages or decide graph connectivity (Merrill et al., 2022; Merrill & Sabharwal, 2023; Chiang et al., 2023; Strobl et al., 2024). These are not exotic failures. They are the formal shadow of state tracking and of multi-step reasoning (Liu et al., 2023; Merrill et al., 2024; Wei et al., 2022), which is why depth - as opposed to width, or a short chain of thought - has become a resource worth studying on its own (Merrill & Sabharwal, 2025).

One way to buy depth without buying parameters is to loop a shared block (Dehghani et al., 2019; Yang et al., 2024; Geiping et al., 2025). A weight-tied transformer unrolled to $\Theta(\log n)$ depth can recognize any regular language and solve connectivity, and that is also, empirically, the depth training needs (Merrill & Sabharwal, 2025). Recurrence is the criterion that unrolling turns into: the same block can be applied at test time beyond the number of steps it was trained on (Geiping et al., 2025; Bansal et al., 2022; Schwethelm et al., 2026; Saunshi et al., 2025; Prairie et al., 2026; Movahedi et al., 2026; Deng et al., 2026).

The loop still has to converge. That is a trade-off between a contraction requirement and signal preservation, and weight tying is what makes the trade-off binding. The transition operator between recurrent steps has to contract, or the loop destabilizes: that is the reason for controlling the spectral radius (Prairie et al., 2026), and for residual scaling and Jacobian regularization (Wang et al., 2026; Yang et al., 2026). The same map has to keep a signal readable across many recurrent steps, or a partial answer written on one step is gone by the step that needs it. Signal preservation, from this view, is stability of the eigenbasis of the transition operator. States at different recurrent steps stay comparable when the angles between them are preserved, up to some tolerance, and then the signal is readable along the depth axis.

Angle preservation is what a conformal transformation is. The same idea is already how length generalization is obtained on the token axis, by a relative rotation (Su et al., 2024), how several timescales are kept in a Clockwork RNN (Koutník et al., 2014), and how a magnitude is paired with a phase once the recurrence is written in the complex plane, as in the LRU and in diagonal state-space models (Orvieto et al., 2023; Gu et al., 2022).

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.

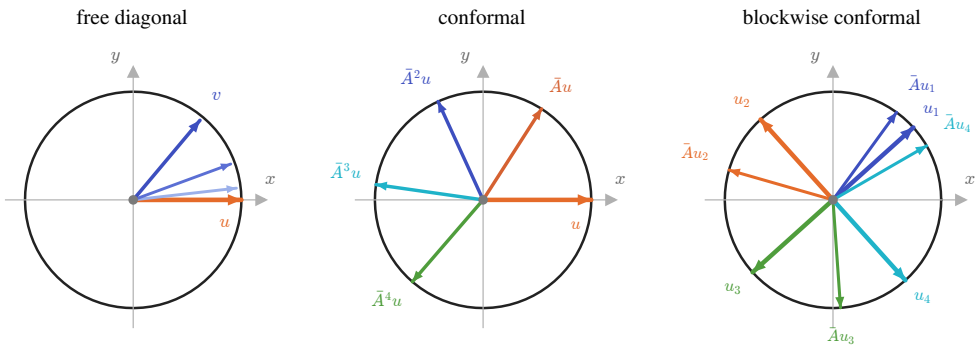


Figure 1: Depiction of how a write moves across recurrent steps, once the transition operator has been applied and the result renormalized. The free diagonal stretches each coordinate by its own factor, so two writes that started apart are dragged onto one axis. A conformal transition operator is a rotation: the write travels around the circle, and the angle between two writes stays the angle they were written with. Blockwise conformal splits the state into a few groups, each turning at its own rate, so the same rotation still moves the signal, and different groups can age at different speeds.

These properties are the reason a loop can contract and still be read. A single shared scale gives every direction the same lifetime, so the state is split into a few groups, each with its own rate and its own rotation: a complex diagonal whose magnitudes are tied inside a group. Existing methods do the opposite. Within the scope of our derivation, they rescale each coordinate by its own factor, one direction comes to dominate the eigenbasis, and relative angles are not preserved, thereby demonstrating that a nonnegative real spectrum is the regime that cannot express the state-tracking problems we study (Sarrof et al., 2024; Grazzi et al., 2025).

Overcoming this structural limitation requires enforcing angle preservation explicitly by constraining the geometry of the transition operator. In this paper, we bridge this gap by introducing a **blockwise conformal parameterization** that guarantees stability along the depth axis while preserving the integrity of the hidden state. By aligning the mathematical requirements of signal preservation with the architecture of looped transformers, we enable robust, long-horizon recurrence at test time.

Our contributions are

- a theoretical characterization of the destabilizing factor of leaving the eigenbasis of the transition operator unconstrained in the recurrent form of a looped transformer, together with a parameterization that constrains it and keeps the hidden state readable across any number of recurrent steps - a pathway toward test-time scaling that is not capped at the training trip count;
- improved last-token accuracy on state-tracking word problems, where the model must emit the correct final product given that the prefix is already correct;
- improved language-model quality when a signal has to reach the end of the sequence, with lower validation perplexity and higher LAMBADA, so the same linear transition operator helps on text as well as on the synthetics.

2 Related Work

Looped and recurrent-depth transformers. Weight tying with a variable trip count goes back to the Universal Transformer (Dehghani et al., 2019) and has become an axis of test-time compute: k layers looped L times match kL distinct layers on iterative reasoning and can simulate chain-of-thought steps in latent space (Saunshi et al., 2025); an adaptive trip count generalizes in length on copying, parity and addition (Fan et al., 2025); a 3.5B recurrent-depth model keeps improving on GSM8K out to tens of loops (Geiping et al., 2025); and looped models at 1.4B and 2.6B trained on trillions of tokens reach the quality of much larger fixed-depth models (Zhu et al., 2025). Depth allocated per token rather than per model is a parallel line (Bae et al., 2025; Bansal et al., 2022), and on the theory side $\Theta(\log n)$ looped depth suffices to recognize regular languages and is what training appears to need (Merrill & Sabharwal, 2025).

Parcae. Prairie et al. (2026) treat the recurrent state as a discretized linear recurrence and inject the prelude embedding at every pass through a diagonal transition operator with entries in $(0, 1)$, which bounds the spectral radius below one and makes long trip counts stable. Residual scaling, Jacobian regularization and related stabilizers pursue the same end by other means (Wang et al., 2026; Yang et al., 2026; Wang & Reid, 2026), and equilibrium treatments take the fixed point as the object of interest (Anil et al., 2022; Huang et al., 2026; Movahedi et al., 2026; Fein-Ashley & Rashidinejad, 2026). All of these constrain the *magnitude* of the transition operator. Prairie et al. (2026) note that many forms of the injection would work and choose a nonnegative diagonal because it is simple and cheap; that remark is the opening this paper takes up.

Test-time generalization across recurrent steps. The Universal Transformer adds a sinusoid of the recurrence index to the state (Dehghani et al., 2019), and Xu & Sato (2025) condition per-loop residual scales and normalization gains on an additive frequency embedding of the loop index. An additive index tells the model which recurrent step it is on, and at the same time it restricts what the model can do: the embedding is tied to the trip counts seen in training, so the loop cannot freely extrapolate past that range. On the token axis, rotary embeddings (Su et al., 2024) were the first to put a relative rotation into the residual geometry; later work generalizes the planes (Zhang et al., 2025) or accumulates Householder reflections along the sequence (Yang et al., 2025). Those maps are built per token and applied once.

Structured orthogonality and linear recurrences. The LRU and diagonal state-space models introduce a complex eigenbasis, with a magnitude and a phase per plane (Orvieto et al., 2023; Gu et al., 2022). We use that same notion, applied to the recurrence that buys the model test-time compute rather than to the sequence axis. Vorontsov et al. (2017) show that an orthogonal parameterization improves performance, and also that a hard constraint is not required: a neighbourhood of the orthogonal family trains even better. That softer neighbourhood is the same tendency as a Clockwork RNN (Koutník et al., 2014; Mikolov et al., 2015) - several timescales instead of one - and it is the tendency we keep when the state is split into groups. Jing et al. (2017) build a unitary from layers of 2×2 rotations and find that a small capacity is preferable on copying. Block-diagonal orthogonal factors appear in parameter-efficient finetuning (Liu et al., 2024; Gorbunov et al., 2024), on weight matrices rather than on a recurrent transition operator.

Normalization and privileged bases. Ashkboos et al. (2024) establish that an RMSNorm residual stream is exactly invariant to an orthogonal change of basis, so a rotation applied to the state is invisible to the normalizer that follows it. Elhage et al. (2023) document that the residual stream nonetheless develops an approximately axis-aligned privileged basis, attributable to per-coordinate optimizer normalizers. Together: rotating the residual stream is free as far as the norm is concerned, while the coordinate frame is not an arbitrary choice - two facts that matter when we place structure in that stream.

3 Motivation of Readability

The block of a transformer is nonlinear, and that is where its expressivity lives. In a loop the same block is applied at every recurrent step, and the cost of that expressivity is how much of a signal present at the start of the recurrence is still readable at the end. Between two steps the residual passes through one linear map, the transition operator $\bar{A}$, and the block that follows is the same map every time. Whatever that block computes, it computes as an inner product: attention logits, a linear readout, the pre-norm rescaling. The transition operator therefore decides whether those inner products still match the ones the block would have computed when the signal was written. Proofs are in Section C.

3.1 A Readable Operator is a Conformal One

Definition 3.1 (readable)**.** A matrix $M \in \mathbb{R}^{d\times d}$ is *readable* if there is a $c > 0$ with $\langle Mx, My\rangle = c^2\langle x, y\rangle$ for all $x, y \in \mathbb{R}^d$.

The block compares states by inner products, so a state stays readable when any two vectors can still be compared after the operator. The definition says that directly: for every pair x, y, the inner product of Mx and My is the inner product of x and y, up to one scale shared by the whole state. The comparison of x with y is the comparison it was. An operator that treats directions differently fails the test, because that same pair no longer presents the same comparison. That is what a later

recurrent step needs if it is to read a state written on an earlier one, and it is the property the transition operator has to stabilize.

Proposition 3.2 (readability is conformality). *M is readable with constant c if and only if $M = cQ$ with $Q^\top Q = I$.*

Readability, read strictly, is conformality: the transition operator preserves the angles between eigenvectors. We write $A_{\mathrm{rot}} = cQ$ for such a map.

Corollary 3.3 (what the block sees). *If $\bar{A} = cQ$ then for all x, y,*

$$\cos\angle(\bar{A}x, \bar{A}y) = \cos\angle(x, y), \qquad \|\bar{A}x - \bar{A}y\| = c\,\|x - y\|, \qquad \frac{\bar{A}x}{\|\bar{A}x\|} \;=\; Q\,\frac{x}{\|x\|}. \tag{1}$$

A pre-norm divides out the shared scale, so the normalized state that reaches the block is a rotated copy of what was written. An operator with a free diagonal rescales each coordinate by its own factor. One direction then dominates the eigenbasis, and a single global normalizer cannot undo that.

3.2 Weight Tying Is a Curse for Readability

Rotary embeddings (Su et al., 2024) put a rotation on the token axis and apply it once. On the depth axis the same kind of map is applied to its own output at every recurrent step, and repeated application of a linear map is power iteration.

Proposition 3.4 (iteration collapses a non-conformal operator). *Let $\bar{A}$ be normal with eigenvalues λ_j, and let $\mathcal{E}$ be the eigenspace of largest magnitude. If $\bar{A} \in \mathrm{CO}(d)$ then $\|\bar{A}^t u\| = c^t \|u\|$ for every u, and $\bar{A}^t u/\|\bar{A}^t u\| = Q^t(u/\|u\|)$ moves on the unit sphere without converging. If instead the largest magnitude is strictly dominant, then $\mathrm{dist}(\bar{A}^t u/\|\bar{A}^t u\|,\ \mathcal{E}) \to 0$ geometrically for every u with a nonzero component in $\mathcal{E}$.*

A transition operator that scales directions differently does not only fade a write. It turns every write toward one subspace, so writes from different recurrent steps end up pointing the same way. A conformal transition operator does not: the direction keeps moving, and the block can still tell the steps apart. Readability at one step propagates to every later step, and the distortion between those steps is estimated by a cosine, one for every pair of depths. For $u \neq 0$ let

$$G_u[t, s] \;=\; \frac{\langle \bar{A}^t u,\ \bar{A}^s u\rangle}{\|\bar{A}^t u\|\,\|\bar{A}^s u\|}, \qquad 0 \le s, t \le T, \tag{2}$$

the normalized Gram matrix of a single write. Each entry is how much the geometry of the operator has distorted the state between those two recurrent steps.

Proposition 3.5 (depth invariance). *If $\bar{A} = cQ$ then $\bar{A}^k$ is readable with constant c^k for every $k \ge 1$, and for every $u \neq 0$, $G_u[t, s] = \langle u, Q^{\,t-s} u\rangle/\|u\|^2$, which depends on t and s only through $t - s$.*

So G_u is constant along its diagonals: two states a fixed number of steps apart present the block with the same geometry wherever in the loop they sit. Figure 2 illustrates the differences between these operators.

3.3 Groups, and the Family They Generate

One shared scale gives every direction the same lifetime. The relaxation is to partition $\mathbb{R}^d = V_1 \oplus \cdots \oplus V_G$ into coordinate subspaces and give each subspace its own scale,

$$A_{\mathrm{rot}}^{(G)} \;=\; \mathrm{blockdiag}\left(c_1 Q_1,\ \ldots,\ c_G Q_G\right), \qquad c_g > 0, \quad Q_g^\top Q_g = I_{V_g}. \tag{3}$$

One scale and one orthogonal matrix per group.

Proposition 3.6 (blockwise readability). *For $\bar{A} = A_{\mathrm{rot}}^{(G)}$ as in (3) and any g, the restriction $\bar{A}|_{V_g}$ is readable with constant c_g: $\langle \bar{A}x, \bar{A}y\rangle = c_g^2\langle x, y\rangle$ for all $x, y \in V_g$. Consequently Corollary 3.3, Proposition 3.5 and the first half of Proposition 3.4 hold verbatim on V_g. For $x \in V_g$ and $y \in V_h$ with $g \neq h$ both sides vanish, and the relative norms of the two components evolve as $(c_g/c_h)^t$.*

The guarantee holds for any number of groups and any spread of rates, as long as a write lives inside one group. What changes across groups is only how long that write lasts: several timescales in one residual, the Clockwork construction (Koutník et al., 2014) on the depth axis. $G{=}1$ is one rotation for the whole state; letting every plane keep its own rate frees the spectrum.

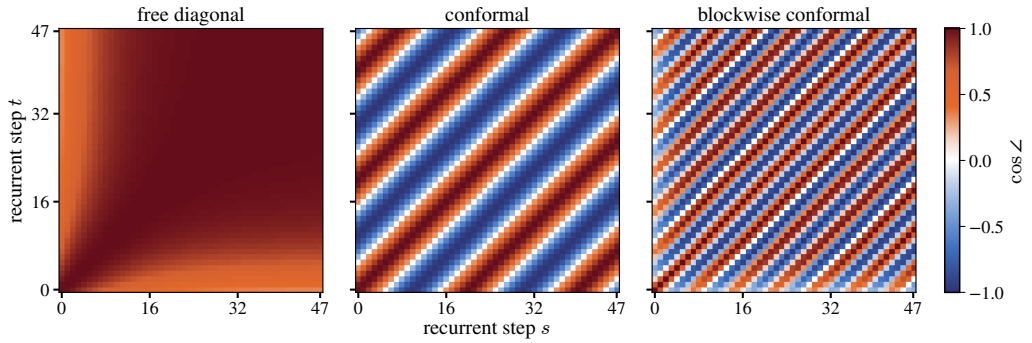


Figure 2: Each panel is the normalized Gram matrix of one write, the cosine of the angle between that write at recurrent step t and at recurrent step s. Free diagonal: the entries collapse toward one value, so the block cannot tell which recurrent step the signal came from. Conformal: the stripes depend only on how many steps apart the two appearances are, so the block can read the depth at which the signal was written. Blockwise conformal splits the state into coordinate-aligned groups. Inside a group the angles stay readable, as in the conformal panel, but the scale is now that group's own rate, so a write occupying several groups lays several timescales on top of one another. The finer texture is that granularity: the block can still read the depth of the signal, and each group keeps it readable for a different number of steps.

Proposition 3.7 (identification)**.** *Let* $Q_g = \mathrm{blockdiag}(R(\theta_i))_{i\in V_g}$ *with* $R(\theta)$ *the planar rotation by* θ*. Then* (3) *is unitarily equivalent to the complex diagonal*

$$A_{\mathrm{rot}}^{(G)} \;\cong\; \mathrm{diag}\left(c_{g(i)}\, e^{\mathrm{i}\theta_i}\right)_{i=1}^{d/2}, \tag{4}$$

whose eigenvalue phases *are free per plane and whose* magnitudes *are tied within a group. At* $G = 1$ *all magnitudes are tied; at* $G = d/2$ *they are free.*

Equation (4) is that family written in the complex eigenbasis: a phase per plane, a magnitude shared inside a group (Orvieto et al., 2023; Gu et al., 2022). Here t counts recurrent steps. Setting every phase to zero recovers a nonnegative diagonal, which cannot express the state-tracking problems we study (Sarrof et al., 2024; Grazzi et al., 2025).

4 Blockwise Conformal Injections

A looped transformer runs one weight-tied block $\mathcal{B}$ for T recurrent steps over a residual state $h_t \in \mathbb{R}^d$, re-injecting the prelude embedding e at every step:

$$h_{t+1} \;=\; \mathcal{B}\big(\bar{A}\, h_t + \Delta\, Be\big), \qquad t = 0, \ldots, T-1, \tag{5}$$

with a prelude producing e and a coda reading h_T. The published choice (Prairie et al., 2026) discretizes a real diagonal state-space recurrence,

$$A_{\mathrm{diag}} \;=\; \exp\big(-\Delta \odot \exp(a)\big) \;=\; \mathrm{diag}(\alpha_i), \qquad \alpha_i \in (0,1), \tag{6}$$

with $a \in \mathbb{R}^d$ and $\Delta = \mathrm{softplus}(\delta) \in \mathbb{R}^d_{>0}$ both learned per coordinate. The spectrum is real and nonnegative by construction, and the α_i are free of one another. This section replaces (6) with the transition operator (3). We partition the d coordinates into G contiguous groups of equal width and pair the coordinates inside each group, giving $P = d/2$ planes with P/G planes per group. Each group g has a rate pair (γ_g, δ_g) and each plane i has a frequency ω_i:

$$c_g \;=\; \exp\big(-\mathrm{softplus}(\gamma_g)\,\mathrm{softplus}(\delta_g)\big) \in (0,1), \qquad \theta_i \;=\; \omega_i\, \mathrm{softplus}(\delta_{g(i)}), \tag{7}$$

and $\bar{A} = \mathrm{blockdiag}\left(c_g R(\theta_i)\right)$ as in (3), with $R(\theta)$ the planar rotation. The coupling $\theta_i = \omega_i \Delta_g$ in (7) is the rule of discretization, read on a complex pole, $\exp\big(\Delta(-\gamma + \mathrm{i}\omega)\big) = e^{-\gamma\Delta}\big(\cos\Delta\omega + \mathrm{i}\sin\Delta\omega\big)$, so that decay and phase share one timescale inside one group, and (6) is the special case $\omega \equiv 0$. Swapping the two changes the structure of $\bar{A}$ and not the contraction rate training starts from.

$\bar{A}$ is never materialized. Because the transition operator is a product of planar rotations, we construct it with the kernel rotary position embeddings apply to queries and keys (Su et al., 2024), here applied

Table 1: **State tracking on A_5 and S_5.** Last-token accuracy on word problems of length n=128, width 512, trained with the recurrence count drawn from a truncated Poisson at $T_{\max}$=32. A_5 is on the left and S_5 on the right: the same injections, on a larger group that also contains odd permutations. Early prefixes resolve within a few steps, so the score is the full product, the one position that has to survive the whole loop. T=16 and T=32 are in-budget evaluations; T=48 is the same checkpoint at a recurrence budget it was never trained on. Cells are mean ± standard error over 10 seeds. G=256 is one group per rotation plane, the finest chart the width d=512 affords. Results within 95% of the best value are written in **bold**. Best values are highlighted in blue.

		A_5			S_5		
Injection	G	T=16	T=32	T=48	T=16	T=32	T=48
Diagonal	-	0.537 ±0.098	0.692 ±0.097	0.771 ±0.079	0.398 ±0.096	0.531 ±0.092	0.604 ±0.084
Blockwise conformal	1	0.628 ±0.066	0.746 ±0.061	0.795 ±0.050	0.511 ±0.071	0.624 ±0.065	0.681 ±0.055
	2	**0.812** ±0.058	**0.894** ±0.052	**0.926** ±0.040	**0.706** ±0.061	**0.808** ±0.054	**0.837** ±0.047
	4	0.481 ±0.076	0.573 ±0.074	0.632 ±0.070	0.364 ±0.074	0.448 ±0.072	0.502 ±0.068
	8	0.527 ±0.071	0.614 ±0.067	0.679 ±0.063	0.409 ±0.069	0.491 ±0.065	0.554 ±0.061
	16	0.661 ±0.057	0.766 ±0.053	0.819 ±0.045	0.542 ±0.061	0.647 ±0.057	0.693 ±0.049
	32	0.603 ±0.069	0.701 ±0.065	0.757 ±0.059	0.473 ±0.067	0.576 ±0.063	0.631 ±0.057
	64	0.637 ±0.063	0.724 ±0.057	0.784 ±0.052	0.518 ±0.061	0.601 ±0.056	0.664 ±0.051
	128	0.676 ±0.049	0.779 ±0.043	0.824 ±0.041	0.556 ±0.053	0.661 ±0.047	0.709 ±0.045
	256	0.654 ±0.059	0.741 ±0.054	0.791 ±0.049	0.533 ±0.057	0.612 ±0.053	0.676 ±0.049

to the recurrent state on the depth axis rather than to token positions, so we can still use the RoPE kernel. Further details are in the implementation of Section D.

There are no other changes in the architecture. The block, the prelude, the coda, the tokenizer, the objective, the optimizer, the schedule and the data stay as they are. The only edit is the body of the injection module, so every comparison in Section 5 differs in the $d \times d$ map applied between recurrent steps.

5 Experiments

5.1 Blockwise Conformality Improves State Tracking

We follow the synthetic state-tracking scenario of Merrill & Sabharwal (2025). The model reads generators $g_1, \ldots, g_n$ of a finite group and has to emit every prefix product $g_1 \cdots g_t$. Training uses the staged length curriculum that paper reports is needed to learn long A_5 words, on those sequences. We train Parcae looped transformers at width 512 with Adam at 10^{-4}, drawing the recurrence count from a truncated Poisson as in Prairie et al. (2026). The baseline is the published learned diagonal (6), trained identically. Last-token accuracy scores only the full product, the one position that has to survive the entire loop, and that is the number we report. Every group-word number is a mean over 10 seeds with its standard error. Full configurations are in Section E.

Table 1 is that comparison, with the recurrence count sampled and the decay left free. At the trained budget T=32 the published diagonal reaches 0.692 last-token accuracy and the blockwise conformal transition operator at G=2 reaches 0.894, an interval apart with no overlap. The separation is already there at T=16, and it is still there at T=48, a recurrence budget none of these checkpoints ever saw. Both methods improve when given extra recurrent steps at evaluation time, which is the behaviour looped models are prized for; the conformal method simply starts higher and stays higher.

The knob that matters is the group count, and its useful range is narrow. G=1, a single rotation and a single lifetime for the whole state, is already ahead of the diagonal. G=2 is the peak. Past that the curve turns around: G=4 and G=8 fall *below* the diagonal baseline, and the larger counts recover only partially, with G=256 - one group per rotation plane, the finest chart this width allows - ending near G=1. We read the shape as the two ingredients pulling in opposite directions. Phases are present at every G and help throughout, which is why the large-G end still lands near the top of the diagonal's range. Tied magnitudes are what a small G buys, and giving each of many groups its own rate hands back exactly the spread the family exists to remove, so the middle of the sweep is the worst of both.

Table 2: **Ablating the rotation.** Last-token accuracy on the A_5 protocol of Table 1 (n=128, width 512, $T_{\max}$=32). The grouped-diagonal rows tie the published baseline's rates inside G coordinate-aligned groups and leave the transition operator elementwise — the same lifetimes as Table 1, without a rotation. T=16 and T=32 are in-budget evaluations; T=48 is the same checkpoint at a recurrence budget it was never trained on. Cells are mean ± standard error over 10 seeds. Every grouped row is above the free diagonal and below the conformal transition operator at the same G=2. Results within 95% of the best value are written in **bold**. Best values are highlighted in blue.

		Last-token accuracy		
Injection	G	T=16	T=32	T=48
Diagonal	-	0.537 ±0.098	0.692 ±0.097	0.771 ±0.079
Grouped diagonal	2	0.721 ±0.051	0.810 ±0.036	0.858 ±0.033
	4	0.694 ±0.054	0.782 ±0.040	0.831 ±0.037
	8	0.668 ±0.057	0.756 ±0.044	0.808 ±0.040
	16	0.644 ±0.061	0.734 ±0.048	0.788 ±0.043
	32	0.619 ±0.066	0.718 ±0.052	0.786 ±0.047
Blockwise conformal	2	**0.812** ±0.058	**0.894** ±0.052	**0.926** ±0.040

Table 3: **Ablating the orthogonal parameterization.** Three nested writings of a conformal factor on A_5 at n=128, width 512, $T_{\max}$=16, scored against the matched slice of the diagonal baseline. All three reach $Q^\top Q = I$, so the setting isolates *which* orthogonal matrices the parameterization can reach - the pairwise chart of (7) covers a $d/2$-dimensional torus, Cayley fills $SO(d)$ up to a null set, and Householder QR reaches both components of $O(d)$. A general orthogonal factor already beats the diagonal; the most restricted writing beats it most. Cells are mean ± standard error over 10 seeds. Results within 95% of the best value are written in **bold**. Best values are highlighted in blue.

Injection	Chart	Last-token accuracy
Diagonal	Coordinate axes	0.408 ±0.061
Blockwise conformal	Pairwise	**0.721** ±0.042
Orthogonal	Householder QR, $O(d)$	0.552 ±0.054
Orthogonal	Cayley, $SO(d)$	0.274 ±0.059

Vorontsov et al. (2017) found an analogous shape on the time axis of an RNN - hard orthogonality is not optimal, an unconstrained spectrum is worse, a small relaxation beats both - and that finding is similar to the sweep above.

We also test the same comparison on S_5 state tracking, the right half of Table 1. The task is harder and the accuracies are lower, and the tendency is the same: G=2 is the peak, the middle of the sweep dips, and the gap remains at T=48. At the trained budget the diagonal reaches 0.531 and G=2 reaches 0.808.

5.2 A Neighbourhood of Conformality Accounts for the Gain

The same group-word protocol varies the two design choices separately: tie the rates inside coordinate-aligned groups (Proposition 3.6), and make the transition operator a rotation rather than a rescaling (Proposition 3.2).

Table 2 removes the rotation and keeps the grouping: the baseline's d free rates are replaced by G rates repeated across coordinate-aligned blocks, with the transition operator still elementwise. Tying rates alone helps, and helps substantially - 0.810 at G=2 and T=32 against the free diagonal's 0.692 - and it prefers the same small group count, with the gain shrinking back towards the baseline as G grows. Read together with Table 1 this gives a clean decomposition of the 0.692 → 0.894 gap at T=32. Tying magnitudes without rotating recovers 0.118 of it. Rotating without tying magnitudes, which is the G=256 row of Table 1, recovers 0.049. Doing both recovers all 0.202. Tying rates is the

Table 4: **Language-model quality improves with a conformal operator.** FineWeb validation perplexity (780k tokens) and LAMBADA exact match (5153 examples) against the number of recurrent steps, at the preferred group count G=2. The gap opens with T: at T=1 the two methods are close; from T=8 the conformal transition operator is ahead by about 15% on perplexity and about 20% on LAMBADA, and the margin holds through T=16, T=32 and T=48, in the same direction as the test-time depth sweep of Fein-Ashley & Rashidinejad (2026). The arrow at T=48 is the relative improvement over the parameter-matched diagonal. Results within 95% of the best value are written in **bold**. Best values are highlighted in blue.

		VALIDATION PPL ↓		LAMBADA ↑	
SIZE	T	OURS	DIAGONAL	OURS	DIAGONAL
60M	1	**28.40**	29.10	0.168	**0.174**
	8	**19.08**	22.45	**0.276**	0.230
	16	**18.82**	22.45	**0.279**	0.231
	32	**18.70**	22.45	**0.280**	0.232
	48	**18.62** ↓17.1%	22.45	**0.281** ↑21.1%	0.232
140M	1	**26.48**	27.04	**0.185**	0.177
	8	**16.56**	19.48	**0.320**	0.267
	16	**16.36**	19.48	**0.323**	0.269
	32	**16.24**	19.47	**0.324**	0.270
	48	**16.18** ↓16.9%	19.47	**0.325** ↑19.9%	0.271
370M	1	**18.72**	19.39	0.228	**0.233**
	8	**12.66**	14.90	**0.427**	0.356
	16	**12.51**	14.90	**0.431**	0.356
	32	**12.42**	14.90	**0.433**	0.357
	48	**12.37** ↓17.0%	14.90	**0.435** ↑21.8%	0.357

larger of the two effects and the rotation the smaller, the same ordering holds at T=16 and T=48, and neither ingredient reproduces the full gap alone. That is what the theory says should happen, since one choice fixes the geometry inside a group and the other decides how many lifetimes the state gets to have. The useful point is the small-G end: one shared conformal map is already ahead of a free spectrum, and a neighbourhood of it, a few groups rather than one lifetime or one rate per plane, is better than either extreme.

Table 3 varies the second choice in a different direction. Proposition 3.2 names a group, $\mathrm{CO}(d)$, and not a way of writing it down, and Section A explains why the theory cannot choose between writings: every claim is invariant to an orthogonal change of basis. So we compare three parameterizations that all reach $Q^\top Q = I$, at a matched spectrum, and let the optimizer decide. A general orthogonal factor does help - Householder QR reaches 0.552 against the diagonal's 0.408 - but the restricted pairwise chart of (7) reaches 0.721, well above it, while the Cayley writing of $SO(d)$ reaches 0.274 and finishes below the diagonal. Committing to a fixed eigenbasis and learning $d/2$ angles beats learning the whole rotation, and learning the whole rotation still beats the published diagonal. The theory is blind to the basis; the optimizer is not, and the direction of the effect is the one the privileged-basis account predicts (Elhage et al., 2023).

5.3 Blockwise Conformality Improves Language Modelling

The full comparison follows the Parcae training scenario (Prairie et al., 2026), about 80 tokens per parameter, and changes only the injection module. Before that run we chose the group count on a smaller model, 60M parameters at 20 tokens per parameter, the Chinchilla allocation (Hoffmann et al., 2022). Gradients there go through every recurrent step: full backpropagation through time improves perplexity over the truncated recipe and does not change which group count wins, and the sweep selects G=2 (Section F). Three sizes, 60M, 140M and 370M, are then trained once with each transition operator at that group count. Full configurations are in Section E.

Extra depth only pays when the transition operator keeps what the depth produced. Table 4 is the three-size family across recurrence budgets, and the clearest result is in the shape of its columns rather than in any single cell. At T=1 the two methods are close: perplexity favours the conformal transition operator by a few percent, and LAMBADA is a wash - in fact slightly behind at two of the three sizes. That is the control the theory predicts, since a transition operator applied once cannot compound and Proposition 3.4 has nothing to say. From T=8 the conformal transition operator is ahead everywhere, by about 15% on perplexity and about 20% on LAMBADA. Then the two methods stop behaving the

Table 5: **Toeplitz defect of the trained transition operator.** $D(T)$ as in Section B, averaged over 256 probe directions. Conformal values lie within precision error; diagonal cells are mean $\pm$ sd across probes. Results within 95% of the best value are written in **bold**. Best values are highlighted in blue.

		Recurrent steps T			
Size	Operator	8	16	32	48
60M	Conformal	$3.6{\cdot}10^{-16}$	$3.8{\cdot}10^{-16}$	$4.1{\cdot}10^{-16}$	$4.7{\cdot}10^{-16}$
	Diagonal	0.059 $\pm$0.011	0.174 $\pm$0.031	0.283 $\pm$0.023	0.281 $\pm$0.027
140M	Conformal	$3.8{\cdot}10^{-16}$	$4.0{\cdot}10^{-16}$	$4.4{\cdot}10^{-16}$	$5.0{\cdot}10^{-16}$
	Diagonal	0.139 $\pm$0.023	0.310 $\pm$0.012	0.308 $\pm$0.028	0.272 $\pm$0.029
370M	Conformal	$4.3{\cdot}10^{-16}$	$4.6{\cdot}10^{-16}$	$5.0{\cdot}10^{-16}$	$5.4{\cdot}10^{-16}$
	Diagonal	0.220 $\pm$0.047	0.324 $\pm$0.022	0.296 $\pm$0.029	0.270 $\pm$0.029

same way with depth. The diagonal's perplexity is flat from $T{=}8$ onwards at every size - 22.45 at 60M for all of $T \in \{8, 16, 32, 48\}$, and the same to within a hundredth at 140M and 370M - so past about eight recurrent steps the extra depth buys it nothing at all. The conformal transition operator keeps improving through $T{=}48$ at all three sizes, ending about 17% ahead on perplexity and about 20% ahead on LAMBADA. This is the practical form of the argument in Section 3.2: test-time depth is only worth spending if what the earlier steps wrote is still legible to the later ones, and a transition operator that collapses writes onto a common direction runs out of things to compose.

5.4 The Trained Map Has the Predicted Geometry

We further measure this geometry on the language models trained above. The Toeplitz defect asks whether two appearances of one state, a fixed number of recurrent steps apart, still present the block with the same comparison wherever in the loop they sit. That constancy is what depth invariance makes desirable: extra recurrent steps then do not rewrite the geometry the block reads. Section B defines the defect and states the proposition that a vanishing value certifies conformality, and that a nonzero value certifies the operator is not conformal. Table 5 is the measurement. The conformal method stays within precision error at every width and every depth. The diagonal method does not: the defect grows with the number of recurrent steps, and it is already larger at larger width, from 0.06 at 60M and $T{=}8$ toward 0.3 by $T{=}32$. That growth is the collapse of Proposition 3.4, and it meets Table 4 where the diagonal perplexity stops responding to further recurrent steps.

6 Conclusion

We presented a blockwise conformal map between recurrent steps of a weight-tied transformer. The need it answers is state tracking on the depth axis, with a potentially very large number of virtual layers obtained by repeating the same block across more recurrent steps. Those layers count only while a signal written early is still readable late. The map between steps has to contract, or the loop does not settle, and it has to keep the angles between states, or the tied block can no longer compare what it wrote. Left free, one direction comes to dominate as the same map is applied again and again, and the comparisons collapse. Conformality, or a neighbourhood of it, is the practical way through: the state shrinks together, the angles stay, and a few groups are allowed their own pace, so part of the signal outlasts the rest. That is what turns a long unroll into computation rather than a fade. The gains are on an extreme form of state tracking, the word problem in a finite group, and the same map also improves language modelling. The direction worth working on is to spend that depth. Larger models, run for many more recurrent steps than training currently affords, are where a readable loop should keep using layers it was never shown.

Reproducibility Statement

Every claim rests on artifacts we describe. The construction is a self-contained replacement for one module. Nothing outside that module differs between the methods of a comparison. We report ten seeds on every group-word table, retain every seed, and build on publicly available data and a publicly

available codebase. Full configurations, evaluation protocols and the complete implementation of the transition operator are given in Section D and Section E. Source code and configurations will be released.

Ethics Statement

This work studies the linear map between passes of a weight-tied transformer. The experiments use publicly available corpora and synthetic word problems over finite groups generated by the authors; no human subjects, no personally identifying information and no new data collection are involved.

AI Use Statement

We used generative AI tools, in particular a large language model (LLM), as a writing and coding assistant. Concretely, we used them to improve the clarity and grammar of the exposition and to produce the plotting and partly implementation scripts. We have reviewed all AI-assisted text and code and take full responsibility for the final content of this work.

References

Cem Anil, Ashwini Pokle, Kaiqu Liang, Johannes Treutlein, Yuhuai Wu, Shaojie Bai, J. Zico Kolter, and Roger Grosse. Path independent equilibrium models can better exploit test-time computation. *Advances in Neural Information Processing Systems*, 2022.

Saleh Ashkboos, Maximilian L. Croci, Marcelo Gennari do Nascimento, Torsten Hoefler, and James Hensman. SliceGPT: Compress large language models by deleting rows and columns. In *International Conference on Learning Representations (ICLR)*, 2024.

Sangmin Bae, Yujin Kim, Reza Bayat, Sungnyun Kim, Jiyoun Ha, Tal Schuster, Adam Fisch, Hrayr Harutyunyan, Ziwei Ji, Aaron Courville, and Se-Young Yun. Mixture-of-recursions: Learning dynamic recursive depths for adaptive token-level computation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

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. *Advances in Neural Information Processing Systems*, 35:20232–20242, 2022.

David Chiang, Peter Cholak, and Anand Pillay. Tighter bounds on the expressivity of transformer encoders. In *International Conference on Machine Learning (ICML)*, 2023.

Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Lukasz Kaiser. Universal transformers. In *International Conference on Learning Representations (ICLR)*, 2019.

Chunyuan Deng, Yizhe Zhang, Rui-jie Zhu, Yuanyuan Xu, Jiarui Liu, T. S. Eugene Ng, and Hanjie Chen. LT2: Linear-time looped transformers. *arXiv preprint arXiv:2605.20670*, 2026.

Nelson Elhage, Robert Lasenby, and Christopher Olah. Privileged bases in the transformer residual stream. *Transformer Circuits Thread*, 2023.

Ying Fan, Yilun Du, Kannan Ramchandran, and Kangwook Lee. Looped transformers for length generalization. In *International Conference on Learning Representations (ICLR)*, 2025.

Jacob Fein-Ashley and Paria Rashidinejad. Solve the loop: Attractor models for language and reasoning. *arXiv preprint arXiv:2605.12466*, 2026.

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. *arXiv preprint arXiv:2502.05171*, 2025.

Mikhail Gorbunov, Nikolay Yudin, Vera Soboleva, Aibek Alanov, Alexey Naumov, and Maxim Rakhuba. Group and shuffle: Efficient structured orthogonal parametrization. *arXiv preprint arXiv:2406.10019*, 2024.

Riccardo Grazzi, Julien Siems, Jörg K. H. Franke, Arber Zela, Frank Hutter, and Massimiliano Pontil. Unlocking state-tracking in linear RNNs through negative eigenvalues. In *International Conference on Learning Representations (ICLR)*, 2025.

Albert Gu, Karan Goel, Ankit Gupta, and Christopher Ré. On the parameterization and initialization of diagonal state space models. In *Advances in Neural Information Processing Systems*, 2022.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. Training compute-optimal large language models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 30016–30030, 2022.

Benhao Huang, Zhengyang Geng, and Zico Kolter. Equilibrium reasoners: Learning attractors enables scalable reasoning. *arXiv preprint arXiv:2605.21488*, 2026.

Li Jing, Yichen Shen, Tena Dubček, John Peurifoy, Scott Skirlo, Yann LeCun, Max Tegmark, and Marin Soljačić. Tunable efficient unitary neural networks (eunn) and their application to rnns. In *International Conference on Machine Learning (ICML)*, 2017.

Jan Koutník, Klaus Greff, Faustino Gomez, and Jürgen Schmidhuber. A clockwork RNN. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *PMLR*, pp. 1863–1871, 2014.

Bingbin Liu, Jordan T. Ash, Surbhi Goel, Akshay Krishnamurthy, and Cyril Zhang. Transformers learn shortcuts to automata. In *International Conference on Learning Representations (ICLR)*, 2023.

Weiyang Liu, Zeju Qiu, Yao Feng, Yuliang Xiu, Yuxuan Xue, Longhui Yu, Haiwen Feng, Zhen Liu, Juyeon Heo, Songyou Peng, Yandong Wen, Michael J. Black, Adrian Weller, and Bernhard Schölkopf. Parameter-efficient orthogonal finetuning via butterfly factorization. In *International Conference on Learning Representations (ICLR)*, 2024.

William Merrill and Ashish Sabharwal. The parallelism tradeoff: Limitations of log-precision transformers. *Transactions of the Association for Computational Linguistics*, 11, 2023.

William Merrill and Ashish Sabharwal. A little depth goes a long way: The expressive power of log-depth transformers. *arXiv preprint arXiv:2503.03961*, 2025.

William Merrill, Ashish Sabharwal, and Noah A. Smith. Saturated transformers are constant-depth threshold circuits. *Transactions of the Association for Computational Linguistics*, 10:843–856, 2022.

William Merrill, Jackson Petty, and Ashish Sabharwal. The illusion of state in state-space models. In *International Conference on Machine Learning (ICML)*, pp. 35492–35506, 2024.

Tomáš Mikolov, Armand Joulin, Sumit Chopra, Michaël Mathieu, and Marc'Aurelio Ranzato. Learning longer memory in recurrent neural networks. In *International Conference on Learning Representations, Workshop Track*, 2015.

Sajad Movahedi, Vera Milovanović, Shlomo Libo Feigin, Alexander Theus, Thomas Hofmann, Valentina Boeva, T. Konstantin Rusch, and Antonio Orvieto. Fixed-point reasoners: Stable and adaptive deep looped transformers. *arXiv preprint arXiv:2606.18206*, 2026.

Antonio Orvieto, Samuel L. Smith, Albert Gu, Anushan Fernando, Caglar Gulcehre, Razvan Pascanu, and Soham De. Resurrecting recurrent neural networks for long sequences. In *International Conference on Machine Learning (ICML)*, 2023.

Hayden Prairie, Zachary Novack, Taylor Berg-Kirkpatrick, and Daniel Y. Fu. Parcae: Scaling laws for stable looped language models. *arXiv preprint arXiv:2604.12946*, 2026.

Yash Sarrof, Yana Veitsman, and Michael Hahn. The expressive capacity of state space models: A formal language perspective. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Nikunj Saunshi, Nishanth Dikkala, Zhiyuan Li, Sanjiv Kumar, and Sashank J. Reddi. Reasoning with latent thoughts: On the power of looped transformers. In *International Conference on Learning Representations (ICLR)*, 2025.

Kristian Schwethelm, Daniel Rückert, and Georgios Kaissis. How much is one recurrence worth? iso-depth scaling laws for looped language models. *arXiv preprint arXiv:2604.21106*, 2026.

Julien Siems, Timur Carstensen, Arber Zela, Frank Hutter, Massimiliano Pontil, and Riccardo Grazzi. Deltaproduct: Improving state-tracking in linear RNNs via householder products. *arXiv preprint*, 2025.

Lena Strobl, William Merrill, Gail Weiss, David Chiang, and Dana Angluin. What formal languages can transformers express? A survey. *Transactions of the Association for Computational Linguistics*, 12, 2024.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Eugene Vorontsov, Chiheb Trabelsi, Samuel Kadoury, and Chris Pal. On orthogonality and learning recurrent networks with long term dependencies. In *International Conference on Machine Learning (ICML)*, 2017.

Shaowen Wang, Bingrui Li, Ge Zhang, Wenhao Huang, Shen Yan, and Jian Li. On the residual scaling of looped transformers: Stability and transferability. *arXiv preprint arXiv:2606.18524*, 2026.

Wenlong Wang and Fergal Reid. Looped transformers under the jacobian lens: Does the global workspace survive recurrence? arXiv preprint, 2026.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, 2022.

Kevin Xu and Issei Sato. On expressive power of looped transformers: Theoretical analysis and enhancement via timestep encoding. In *International Conference on Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, pp. 69613–69646, 2025.

Liu Yang, Kangwook Lee, Robert D. Nowak, and Dimitris Papailiopoulos. Looped transformers are better at learning learning algorithms. In *International Conference on Learning Representations (ICLR)*, 2024.

Songlin Yang, Yikang Shen, Kaiyue Wen, Shawn Tan, Mayank Mishra, Liliang Ren, Rameswar Panda, and Yoon Kim. PaTH attention: Position encoding via accumulating householder transformations. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Xiao-Wen Yang, Ziyu Han, Xi-Hua Zhang, Wen-Da Wei, Jie-Jing Shao, Lan-Zhe Guo, and Yu-Feng Li. Stabilizing recurrent dynamics for test-time scalable latent reasoning in looped language models. *arXiv preprint arXiv:2605.26733*, 2026.

Yifan Zhang, Andrew Chi-Chih Yao, and Quanquan Gu. GRAPE: Group rotary position embedding. *arXiv preprint arXiv:2512.07805*, 2025.

Rui-Jie Zhu, Zixuan Wang, Kai Hua, Tianyu Zhang, Ziniu Li, Haoran Que, Boyi Wei, Zixin Wen, Fan Yin, He Xing, et al. Scaling latent reasoning via looped language models. *arXiv preprint arXiv:2510.25741*, 2025.

A THEORY

A.1 THE DESIGN SPACE IN ONE PICTURE

Figure 3 draws the family of Section 3.3 as a set of eigenvalues in the unit disk, and it makes the two design choices of Section 5.2 visible side by side.

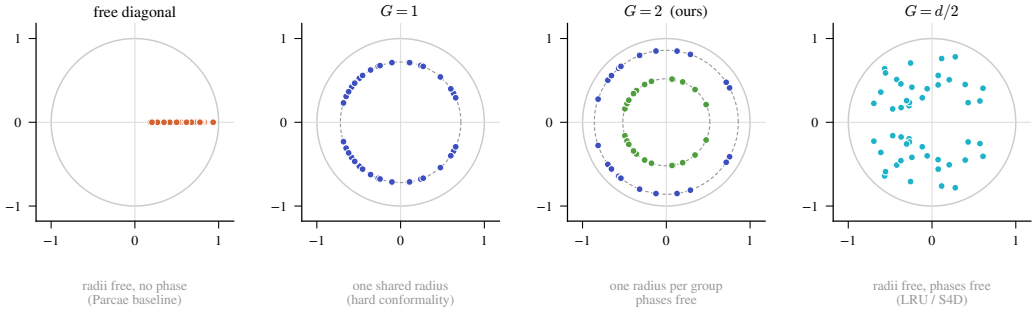


Figure 3: Eigenvalues of the transition operator in the unit disk. Free diagonal: free radii and no phase. One shared radius is a single conformal map; one radius per group is the blockwise family; freeing every radius is a complex diagonal.

Moving from the baseline to our family frees the *phase*, lifting the spectrum off the nonnegative real axis. Moving along the family from $G{=}1$ to $G{=}d/2$ frees the *magnitudes*, one group at a time, until every plane has its own radius and the construction has become an ordinary complex diagonal. The empirical sweeps in Table 1 and Table 8 traverse this picture from left to right, and both stop near its left end.

A.2 WHAT THE THEORY DOES NOT DECIDE

Proposition A.1 (basis blindness)**.** *Let U be orthogonal and $\bar{A}' = U A^{(G)}_{\mathrm{rot}} U^\top$. Then $\bar{A}'$ and $A^{(G)}_{\mathrm{rot}}$ have the same singular values, and G_u computed for $\bar{A}'$ equals $G_{U^\top u}$ computed for $A^{(G)}_{\mathrm{rot}}$. In particular the two transition operators realize the same set of Gram matrices* (2) *and satisfy Proposition 3.2-Proposition 3.4 identically.*

The coordinate alignment of Section 4 is therefore not a geometric claim but a claim about which basis the block's parameters can use. The residual stream of a trained transformer develops an approximately axis-aligned privileged basis, with basis-aligned outlier dimensions attributable to the per-coordinate normalizers of adaptive optimizers (Elhage et al., 2023). A dense U spreads each written item over all d coordinates, so the tied readout no longer finds it where it was put. Proposition A.1 makes this an empirical question rather than a theoretical one, and Table 3 treats it as one by comparing three writings of a conformal factor at matched spectra.

A second freedom is left open in the same way. Proposition 3.2 pins down the scale and leaves Q entirely unconstrained, and how to parameterize Q is an axis with its own literature. Jing et al. (2017) write a unitary as layers of 2×2 rotations with a tunable capacity L, spanning a small subspace at $L = 1$ and the full group at $L = N$, and report that the useful capacity depends on the task. We take $L = 1$, the most restricted writing available, and Table 3 reports what the larger ones do: on this task the restricted chart wins, and the least restricted writing we tried finishes below the baseline.

A.3 WHAT CONFORMALITY TRADES AWAY

Nothing in Section 3 says the loop should fail to settle, and it is worth being precise about what the construction does and does not change. A free diagonal with $a_i < 1$ and a conformal transition operator with $c < 1$ both contract in *norm*; neither prevents the recurrence from converging. By Proposition 3.4 they differ in *direction*. A non-conformal transition operator additionally converges in direction, onto its dominant eigenspace, and a conformal one never does.

A.4 On Importing Spectral Obstructions

The spectral results we invoke in Section 3.3 are theorems about the *token*-axis recurrence of a linear RNN, and we are explicit about what they license. The negative direction is what rules out the baseline: a nonnegative real spectrum cannot recognize PARITY at arbitrary length (Sarrof et al., 2024, Thm. 2), and Grazzi et al. (2025, Thm. 1) show an eigenvalue off the nonnegative reals is necessary, with a non-real one required for counting modulo m (Grazzi et al., 2025, Thm. 2). Freeing the phase in (4) removes that obstruction.

The positive direction does not transfer as cleanly, and we do not claim it does. What those results certify as *sufficient* for composing a non-commutative transition monoid is a transition that depends on the input, realized as a product of generalized Householders (Grazzi et al., 2025; Siems et al., 2025, Thms. 3-4), and a fixed transition operator iterated T times is a statement about powers of one matrix. Our claim on A_5 and S_5 is empirical: the shape of the transition operator changes what gradient descent finds, at matched spectra and matched everything else. Making θ a function of the token would bring the construction inside the sufficiency results, costs nothing in the implementation (Section D), and is the extension we would run next.

A.5 Normalization

Pre-normalization is what makes Corollary 3.3 operational. Ashkboos et al. (2024) show that an RMSNorm residual stream is exactly invariant to an orthogonal change of basis, which is the property (1) needs: a global normalizer removes the shared scale and commutes with Q. No global normalizer can undo a per-coordinate change of angle, and the angular part is what Section 3 constrains.

B The Toeplitz Defect

For a unit probe u and a trip count T, the Toeplitz defect is the root-mean-square distance from the Gram matrix (2) to the nearest matrix that is constant along diagonals:

$$D(T) \;=\; \min_{\tau \in \mathbb{R}^{T+1}} \left(\tfrac{2}{T(T+1)} \sum_{0 \le s < t \le T} \big(G_u[t,s] - \tau_{t-s} \big)^2 \right)^{1/2}. \tag{8}$$

In Section 5.4 the minimum is averaged over 256 probes drawn uniformly on the sphere.

Proposition B.1 (the Toeplitz defect is a certificate)**.** *Let $\bar{A}$ be normal and invertible with eigenvalue magnitudes $r_1 \ge \cdots \ge r_d > 0$. If $r_1 > r_d$ then there is a u for which G_u is not constant along its diagonals. Hence for normal $\bar{A}$, G_u is Toeplitz for every u if and only if $\bar{A} \in \mathrm{CO}(d)$.*

By Proposition 3.5 the defect vanishes for a conformal transition operator. Proposition B.1 is the converse: for a normal operator, a nonzero value certifies that it is not conformal. The proof is in Section C.

C Proofs

Proposition 3.2 (readability is conformality)

If $M = cQ$ with $Q^\top Q = I$, then $\langle Mx, My \rangle = x^\top M^\top M y = c^2 x^\top Q^\top Q y = c^2 \langle x, y \rangle$. Conversely, (3.1) states that $x^\top (M^\top M - c^2 I) y = 0$ for all $x, y \in \mathbb{R}^d$. A bilinear form vanishing on all pairs has zero matrix, so $M^\top M = c^2 I$ and $(M/c)^\top (M/c) = I$. □

Corollary 3.3 (what the block sees)

Setting $y = x$ in (3.1) gives $\|Mx\|^2 = c^2 \|x\|^2$, so $\|Mx\| = c\|x\|$. Then $\cos \angle(Mx, My) = \langle Mx, My \rangle / (\|Mx\| \|My\|) = c^2 \langle x, y \rangle / (c^2 \|x\| \|y\|) = \cos \angle(x, y)$. Bilinearity gives $\|Mx - My\|^2 = \|M(x-y)\|^2 = c^2 \|x - y\|^2$. Finally $Mx/\|Mx\| = cQx/(c\|x\|) = Q(x/\|x\|)$. □

PROPOSITION 3.5 (DEPTH INVARIANCE)

$(cQ)^k = c^k Q^k$ and Q^k is orthogonal, so Proposition 3.2 applies to $\bar{A}^k$ with constant c^k. For the Gram matrix,

$$\langle \bar{A}^t u, \bar{A}^s u\rangle = c^{t+s}\langle Q^t u, Q^s u\rangle = c^{t+s}\langle u, Q^{s-t}u\rangle, \qquad \|\bar{A}^t u\| = c^t\|u\|,$$

using $Q^\top = Q^{-1}$ in the middle step. Substituting into (2), the factor c^{t+s} cancels against $c^t \cdot c^s$, leaving $\langle u, Q^{s-t}u\rangle/\|u\|^2$. Symmetry in $\pm(t-s)$ holds because $\langle u, Q^{-k}u\rangle = \langle Q^k u, u\rangle = \langle u, Q^k u\rangle$. □

PROPOSITION B.1 (THE TOEPLITZ DEFECT IS A CERTIFICATE)

Diagonalize the normal matrix $\bar{A}$ unitarily, with eigenvalues $\lambda_j = r_j e^{\mathrm{i}\theta_j}$ and orthonormal eigenvectors v_j. Take $u = v_1 + v_d$, so that $\bar{A}^t u = r_1^t e^{\mathrm{i}t\theta_1} v_1 + r_d^t e^{\mathrm{i}t\theta_d} v_d$ and $\|\bar{A}^t u\|^2 = r_1^{2t} + r_d^{2t}$. At lag one,

$$G_u[s+1, s] \;=\; \frac{r_1^{2s+1} e^{\mathrm{i}\theta_1} + r_d^{2s+1} e^{\mathrm{i}\theta_d}}{\sqrt{(r_1^{2s+2} + r_d^{2s+2})(r_1^{2s} + r_d^{2s})}}.$$

Divide numerator and denominator by r_1^{2s+1} and write $\beta = r_d/r_1 < 1$:

$$G_u[s+1, s] \;=\; \frac{e^{\mathrm{i}\theta_1} + \beta^{2s+1} e^{\mathrm{i}\theta_d}}{\sqrt{(1+\beta^{2s+2})(1+\beta^{2s})}} \;\longrightarrow\; e^{\mathrm{i}\theta_1} \qquad (s \to \infty),$$

a limit of modulus one. At $s = 0$ the modulus is $|r_1 e^{\mathrm{i}\theta_1} + r_d e^{\mathrm{i}\theta_d}|/\sqrt{2(r_1^2 + r_d^2)} \le (r_1 + r_d)/\sqrt{2(r_1^2 + r_d^2)} < 1$, where the last inequality is strict precisely because $r_1 \neq r_d$ (equality in $(a+b)^2 \le 2(a^2+b^2)$ requires $a = b$). So the lag-one diagonal of G_u is not constant. The reverse implication is Proposition 3.5, and a normal $\bar{A}$ with $r_1 = r_d = c$ satisfies $\bar{A}^\top \bar{A} = c^2 I$, hence lies in $\mathrm{CO}(d)$. □

PROPOSITION 3.4 (ITERATION COLLAPSES A NON-CONFORMAL OPERATOR)

The conformal case is Corollary 3.3 applied to $\bar{A}^t = c^t Q^t$. For the second claim, expand $u = \sum_j a_j v_j$ in the eigenbasis of $\bar{A}$ and factor out λ_1^t:

$$\bar{A}^t u \;=\; \lambda_1^t \Big(\sum_{\lambda_j \in \mathcal{E}} a_j v_j \;+\; \sum_{\lambda_j \notin \mathcal{E}} a_j (\lambda_j/\lambda_1)^t v_j \Big).$$

Every ratio in the second sum has modulus $\beta_j = |\lambda_j|/r_1 < 1$, so that sum vanishes geometrically at rate $\max_{j \notin \mathcal{E}} \beta_j$, while the first sum is a fixed nonzero vector of $\mathcal{E}$ by assumption. Normalizing removes the prefactor λ_1^t up to a unit scalar. □

PROPOSITION 3.6 (BLOCKWISE READABILITY)

$\bar{A}$ preserves each coordinate subspace V_g and acts on it as $c_g Q_g$ with Q_g orthogonal on V_g, so Proposition 3.2 applies verbatim on V_g and gives (3.6). Powers act blockwise, $(c_g Q_g)^t = c_g^t Q_g^t$, so Proposition 3.5 and the norm identity of Proposition 3.4 transfer to V_g unchanged. Distinct V_g are orthogonal as coordinate subspaces and each is $\bar{A}$-invariant, so for $x \in V_g$, $y \in V_h$ with $g \neq h$ we have $\bar{A}x \in V_g$, $\bar{A}y \in V_h$ and both $\langle x, y\rangle$ and $\langle \bar{A}x, \bar{A}y\rangle$ vanish. The stated ratio follows from $\|\bar{A}^t x\| = c_g^t \|x\|$ and $\|\bar{A}^t y\| = c_h^t \|y\|$. □

PROPOSITION 3.7 (IDENTIFICATION)

The planar rotation $R(\theta)$ has eigenvalues $e^{\pm \mathrm{i}\theta}$ with eigenvectors $(1, \mp\mathrm{i})^\top/\sqrt{2}$, so the 2×2 change of basis $W = \frac{1}{\sqrt{2}}\left(\begin{smallmatrix} 1 & 1 \\ -\mathrm{i} & \mathrm{i} \end{smallmatrix}\right)$ is unitary and diagonalizes it. Applying W blockwise on each coordinate pair diagonalizes every Q_g simultaneously, and multiplying block g by the scalar c_g multiplies both of its eigenvalues by c_g, which gives (4). At $G = 1$ there is a single c, so all magnitudes coincide; at $G = d/2$ each plane is its own group and the magnitudes are unconstrained. □

Algorithm 1 The transition operator, as run.

```
# --- inside the loop: x <- Block(A_bar x + dt * B e), one tied block per pass ---
def core_block_forward(self, x, input_embeds, freqs_cis, mask, step, total_steps, **kw):
    x = self.transformer.adapter(x, input_embeds)      # the transition operator, below
    for block in self.active_blocks:                   # the tied recurrent block
        x = block(x, freqs_cis, mask, **kw)
    return x

def forward(self, x_t, e):                             # the injection module
    return self._apply_A(x_t, self._decay()) + self._dt() * (e @ self.B.T)

def _apply_A(self, x, decay):
    return self._rotate(x) * decay                     # ours: rotate, then group scale

# --- rate and angle: c_g = exp(-softplus(gamma_g) softplus(dt_g)), theta_i = omega_i dt_g ---
def _group_expand(self, v):                            # (G,) -> (d,), one value per group
    return v.repeat_interleave(2 * self.pairs_per_group)

def _decay(self):
    gam, dtg = F.softplus(self.gamma_log), F.softplus(self.rot_dt_log)          # (G,), (G,)
    return self._group_expand(torch.exp(-gam * dtg))   # one modulus per group

def _theta(self):
    dt_g = F.softplus(self.rot_dt_log)                 # decay and phase share one timescale
    return self.rot_omega * dt_g.repeat_interleave(self.pairs_per_group)

# --- the transition operator itself: blockdiag(R(theta_1), ..., R(theta_P)) as a pairwise
     complex multiply ---
def _rotate(self, x):
    theta = self._theta().to(x.dtype)
    cos, sin = torch.cos(theta), torch.sin(theta)
    xr = x.reshape(*x.shape[:-1], self.rot_pairs, 2)
    x0, x1 = xr[..., 0], xr[..., 1]
    return torch.stack([x0 * cos - x1 * sin,
                        x0 * sin + x1 * cos], dim=-1).reshape(x.shape)
```

PROPOSITION A.1 (BASIS BLINDNESS)

$\bar{A}'^{\top}\bar{A}' = U(A_{\mathrm{rot}}^{(G)})^{\top}A_{\mathrm{rot}}^{(G)}U^{\top}$ is a similarity by an orthogonal matrix, so it has the same eigenvalues as $(A_{\mathrm{rot}}^{(G)})^{\top}A_{\mathrm{rot}}^{(G)}$ and $\bar{A}'$ has the same singular values as $A_{\mathrm{rot}}^{(G)}$. Since $(\bar{A}')^t = U(A_{\mathrm{rot}}^{(G)})^t U^{\top}$ and U preserves inner products and norms,

$$\frac{\langle(\bar{A}')^t u, (\bar{A}')^s u\rangle}{\|(\bar{A}')^t u\|\|(\bar{A}')^s u\|} = \frac{\langle(A_{\mathrm{rot}}^{(G)})^t U^{\top}u, (A_{\mathrm{rot}}^{(G)})^s U^{\top}u\rangle}{\|(A_{\mathrm{rot}}^{(G)})^t U^{\top}u\|\|(A_{\mathrm{rot}}^{(G)})^s U^{\top}u\|},$$

which is $G_{U^{\top}u}$ for $A_{\mathrm{rot}}^{(G)}$. As u ranges over $\mathbb{R}^d \setminus \{0\}$ so does $U^{\top}u$, so the two sets of Gram matrices coincide. □

D IMPLEMENTATION

Algorithm 1 is the transition operator as run.

Parameter counts. Implemented on its own, the transition operator uses $d/2 + 2G$ scalars, against $2d$ for the published diagonal. The input map B is shared and does not change. That is about four times fewer scalars: 258 against 1024 at width 512 with one group, and 324 against 1280 at the 60M width with two groups.

E EXPERIMENTAL DETAILS

E.1 COMPUTE RESOURCES

All runs were performed on NVIDIA H100 GPUs. A single group-word run at $T_{\max}$=32 with the length curriculum takes roughly 2.5 GPU-hours, and the group-word study is the bulk of the cost because every cell in Table 1, Table 1, Table 2 and Table 3 is ten seeds. Including the language models, the evaluation sweeps, and failed or exploratory runs, the project cost approximately 4000 H100 GPU-hours in total. The evaluations in Sections 5.3 and 5.4 load existing checkpoints and train nothing; the Toeplitz measurement is negligible, since it operates on a single $d \times d$ matrix on CPU.

E.2 Group Word Problems

The model reads a sequence of generators and emits every prefix product. Sequences are sampled rather than read from a fixed split, so train and evaluation words do not overlap by construction. The recurrence count is drawn from a truncated Poisson with the support top $T_{\max}$ stated in each caption, following Prairie et al. (2026). The length curriculum presents lengths $2, 4, 8, \ldots, n$ in equal slices, which Merrill & Sabharwal (2025) report is required to learn long A_5 words at all. Every method within a table shares this configuration exactly; the injection module is the only difference.

Field	Value	Field	Value
Optimizer	Adam	Width	512
Learning rate	10^{-4}	Recurrent block	2 layers, 4 heads
Schedule	constant	Recurrence	$T \sim \Lambda$, truncated Poisson
Warmup steps	300	Word length n	as stated per table
Batch size	128	Curriculum	$2, 4, 8, \ldots, n$ in equal slices
Grad clip	1.0	Precision	float32, TF32 off
Seeds	10	Seed handling	all retained, none dropped

Table 6: Group-word training configuration, identical across methods within a comparison.

Scoring. Mean token accuracy averages over prefixes and is high and nearly identical for every injection, because early prefixes resolve within the first few passes. Last-token accuracy scores only the full product. The T=16 and T=32 columns of Table 1, Table 1 and Table 2 evaluate the $T_{\max}$=32 checkpoints in budget; T=48 re-evaluates the same checkpoints without retraining.

Grouped-diagonal control. The control in Table 2 keeps (6) elementwise and replaces its d free rates by G rates repeated across coordinate-aligned blocks of equal width, matching the partition used by the conformal method at the same G. No rotation is applied. Initialization, optimizer and schedule are those of the baseline.

Charts. The three writings compared in Table 3 all produce an exactly orthogonal factor and differ only in which orthogonal matrices they can reach. The pairwise chart is (7), a product of $d/2$ planar rotations in fixed coordinate planes. The Cayley map sends a learned skew-symmetric S to $(I - S)(I + S)^{-1}$ and covers $SO(d)$ up to a null set. The Householder writing takes the Q factor of a QR decomposition of a learned dense matrix and reaches both components of $O(d)$. The spectrum is matched across the four rows, so the only thing that varies is the reachable set.

E.3 Language Models

We adopt the training scenario of the Parcae codebase (Prairie et al., 2026) without modification - the tokenizer, the data pipeline, the optimizer, the learning-rate schedule, the prelude and coda, and the recurrence sampler - which trains at approximately 80 tokens per parameter. The only edit is the body of the injection module, so the two methods of every comparison differ in the $d \times d$ map applied between passes.

Group-count sweep (Table 8). 60M parameters on FineWeb-Edu at sequence length 2048 and global batch 524k tokens, with the recurrence count drawn from a truncated Poisson at mean 8 and validation perplexity measured at T=8. Group counts must divide the 320 rotation planes at this width, which is what fixes the values of G that appear in the table. Truncated BPTT at 4 is the published recipe; full BPTT at 8 backpropagates through every pass. We report both for the diagonal and for G=1 to show that the choice is worth a small constant and does not reorder anything, and compare the remaining rows at full BPTT.

Geometry and language-model family (Table 5, Table 4). Three widths, each trained once with each transition operator, giving the six checkpoints on which the Toeplitz defect is measured. The recurrent state width equals the model width. The conformal method uses G=2 and the coupled angle $\theta_i = \omega_i \,\mathrm{softplus}(\delta)$ of (7). Recurrence counts are sampled during training around a mean of 8, and evaluation overrides T at load time, so the $T \in \{1, 8, 16, 32, 48\}$ columns of Table 4 are a test-time depth sweep in which $T \geq 16$ is beyond the training distribution.

Field	60M	140M	370M
Width d	640	768	1024
State width	640	768	1024
Mean recurrence (train)	8	8	8
Injection groups G	2	2	2
Angle mode	coupled ($\theta = \omega\Delta$)	coupled	coupled

Table 7: Configurations for the three-size family. Everything not listed is the baseline's (Prairie et al., 2026) and is identical across the two methods at a given width.

Toeplitz defect. $\bar{A}$ is assembled in float64 on CPU, its powers are taken directly, and (8) is solved in closed form by averaging each diagonal of G_u. Probes are drawn uniformly on the unit sphere with a fixed seed shared across methods, so the two methods of a comparison see the same 256 directions.

F Preliminary Group Count

The sweep that selects G for the language-model comparison is a 60M model trained at 20 tokens per parameter, the Chinchilla allocation (Hoffmann et al., 2022). Full backpropagation through the eight recurrent steps improves perplexity over truncation at four steps, on both the diagonal (22.71 to 22.45) and G=1 (19.64 to 19.42), and it does not change the ordering. At full backpropagation the diagonal reaches 22.45, G=2 reaches 19.08, G=1 reaches 19.42, and every other count falls between 19.31 and 19.66. Two groups is the value used in Table 4.

Table 8: **Preliminary group count on a small language model.** Validation perplexity at T=8 for a 60M looped model trained at 20 tokens per parameter on FineWeb-Edu, sequence length 2048 and global batch 524k tokens, with the recurrence count drawn from a truncated Poisson at mean 8. The injection is the only difference between rows. The G=2 entry is the 60M, T=8 cell of Table 4. G=320 is one group per rotation plane, the finest chart this width affords. BPTT is how far gradients travel through the loop: truncated (4) is the published recipe and full (8) backpropagates through every pass, the comparison of Schwethelm et al. (2026); it is a small consistent gain on every matched pair and does not change the ordering, so the comparison we quote is the pair at full BPTT. The arrow is the relative improvement of G=2 over the diagonal at full BPTT. Results within 95% of the best value are written in **bold**. Best values are highlighted in blue.

Injection	BPTT	G	Validation ppl ↓
Diagonal	Truncated (4)	-	22.71
	Full (8)	-	22.45
Blockwise conformal	Truncated (4)	1	19.64
	Full (8)	1	19.42
	Full (8)	2	**19.08** ↓15.0%
	Full (8)	4	19.31
	Full (8)	8	19.48
	Full (8)	64	19.62
	Full (8)	160	19.66
	Full (8)	320	19.52