COMPLEX KDA: UNDERSTANDING AND ENHANCING THE EXPRESSIVITY OF KIMI DELTA ATTENTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Linear RNNs based on the delta-rule enable efficient sequence modeling, but their linear updates with a low-rank correction constrain their expressivity. Prior work has shown that composing two delta-rule transitions in a single recurrent update can model a 2D rotation, but this increases the rank and the cost of the updates compared to a single transition. We show that Kimi Delta Attention (KDA) can realize 2D rotations by combining a single delta-rule transformation with a second reflection supplied by its channel-wise gate. This requires extending the parameter ranges of KDA by combining two existing range extensions: allowing gates in $[-1, 1]$ and the delta-rule coefficient β in $[0, 2]$. We call the resulting model Complex KDA (CKDA). It preserves KDA's stability and efficiency, with transitions that remain diagonal-plus-rank-one and non-expansive, while reaching the known state-tracking expressivity of DeltaProduct$_2$. We characterize the expressivity of CKDA and prove that every orthogonal diagonal-plus-rank-one matrix is exactly a CKDA transition matrix. A single CKDA layer can track every finite group isomorphic to a subgroup of $\mathrm{SO}(3)$, and many state-tracking results use one fewer layer for CKDA compared to other diagonal-plus-rank-one Linear RNNs. Empirically, combining both extensions yields the strongest length extrapolation among tested KDA range settings on S_3, S_4, and periodic audio continuation. In language modeling, CKDA (and its attention hybrid) is competitive to KDA, outperforms Transformers and other linear RNNs, and shows promising scaling behavior.

1 INTRODUCTION

Linear recurrent neural networks (RNNs) offer efficient sequence modeling with linear scaling in sequence length and a fixed-size recurrent state. Their efficiency and expressivity depend on the structure of their state-transition matrices: diagonal transitions, as used in Mamba-1/2 (Gu & Dao, 2024; Dao & Gu, 2024), GLA (Yang et al., 2024a), and mLSTM (Beck et al., 2024), support fast computation, while non-diagonal transitions using the delta-rule (Schlag et al., 2021) introduce a rank-one correction that mixes information across state coordinates (Yang et al., 2024b; Peng et al., 2025; Hatamizadeh et al., 2026). Gated delta-rule models have become recurrent backbones of large language models, with two variants differing in how they parameterize the gate in the state-transition $\boldsymbol{A}_t = (\boldsymbol{I} - \beta_t \boldsymbol{k}_t \boldsymbol{k}_t^\top)\,\mathrm{Diag}(\boldsymbol{\alpha}_t)$. Gated DeltaNet (GDN) (Yang et al., 2025) uses a scalar gate, $\boldsymbol{\alpha}_t = \alpha_t \mathbf{1}$, and is adopted in several recent language models (Qwen Team, 2026; Qiu et al., 2026; Merrill et al., 2026b). Kimi Delta Attention (KDA) (Kimi Team, 2025) allows a separate gate value per channel and has likewise been adopted in recent models (Upstage Solar Team, 2026; Z.ai, 2026; inclusionAI, 2026; Kimi Team et al., 2026). Both retain diagonal-plus-rank-one transitions, motivating us to understand what the shift from a scalar to a full diagonal adds to their expressivity.

We study this expressivity question through state tracking, which requires composing input-dependent updates over time, as in parity, modular addition or permutation composition. Equipped with increasingly complex orthogonal transitions, Linear RNNs can solve harder state-tracking problems with one layer: 1D reflections for parity, 2D rotations for modular addition and higher-dimensional orthogonal representations for permutation composition. DeltaProduct$_k$ (Siems et al., 2025) controls this complexity by composing k delta-rule transitions per token, each an identity-plus-rank-one matrix. In particular, one layer can solve modular addition via 2D rotations when $k = 2$, using two delta-rule updates per token but increasing the computational cost relative to DeltaProduct$_1$.

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.

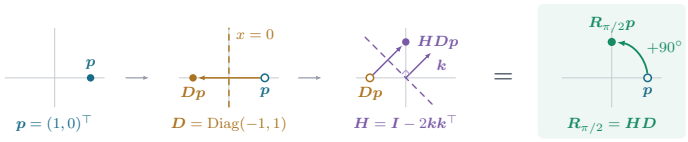

Figure 1: Visualization of CKDA applying a rotation to a vector $\boldsymbol{p}$ within a single recurrent update $(\boldsymbol{I} - \beta \boldsymbol{k}\boldsymbol{k}^\top)\,\mathrm{Diag}(-1, 1)$ by combining a signed diagonal gate with a Householder reflection ($\beta = 2$). $\boldsymbol{k} = (1, 1)^\top/\sqrt{2}$ allows a 90° rotation, while varying $\boldsymbol{k}$ allows any planar rotation angle.

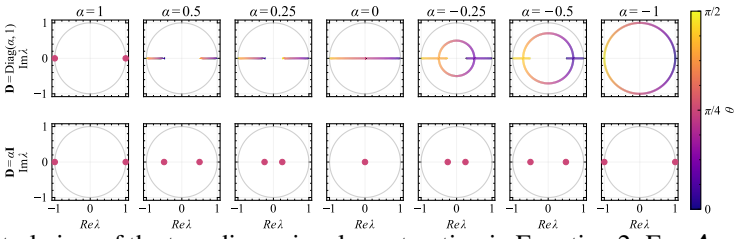

Figure 2: Spectral view of the two-dimensional construction in Equation 2. For $\boldsymbol{A} = (\boldsymbol{I} - 2\boldsymbol{k}\boldsymbol{k}^\top)\boldsymbol{D}$ with $\boldsymbol{k} = (\cos\theta, \sin\theta)^\top$ and $\theta \in [0, \pi/2]$. For the channel-wise gate $\boldsymbol{D} = \mathrm{Diag}(\alpha, 1)$, negative α allows a complex-conjugate pair, which moves toward the unit circle as α approaches -1 and recovers a rotation. For the scalar gate $\boldsymbol{D} = \alpha\boldsymbol{I}$, the spectrum remains real (see also Figure 8).

We show that KDA's channel-wise gate provides the structure needed to break the symmetry of the Householder–diagonal transition, making complex eigenvalues possible, whereas GDN's scalar gate cannot break this symmetry. However, KDA's standard nonnegative gate still restricts the transition to a real spectrum. We introduce *Complex KDA (CKDA)* by combining two existing range extensions: gate entries in $[-1, 1]$ (Sarrof et al., 2024) and $\beta_t \in [0, 2]$ (Grazzi et al., 2025). As a consequence, CKDA can realize any 2D rotation while maintaining non-expansiveness and efficient recurrent computation. CKDA realizes planar rotations using gate entries of opposite sign to supply coordinate-aligned reflections, which combine with the freely oriented Householder reflection at $\beta_t = 2$ (Figures 1 and 2). Our main contributions are:

- We characterize the spectrum of CKDA's transitions and prove that every orthogonal diagonal-plus-rank-one matrix is exactly a CKDA transition. We also study the expressivity of products of CKDA matrices (Section 4).
- One CKDA layer tracks every finite subgroup of $\mathrm{SO}(3)$ (3D rotations), including S_3, S_4, and A_5 (Theorem 3). Three layers solve arbitrary finite-group word problems and, with $\beta > 2$, simulate weighted finite automata in polynomial precision (Theorem 5). These bounds match $\mathrm{DeltaProduct}_2$ and use one layer fewer than (Gated) DeltaNet.
- We rule out one-layer S_5 tracking by CKDA and $\mathrm{DeltaProduct}_k$ ($k \leq 3$) when states are bounded and states mapping to different group elements are uniformly separated (Theorem 4). A general reduction transfers spectral impossibility results for a finite number of reachable states with orthogonal transitions and no additive terms to these broader assumptions.
- Combining both extensions yields the strongest length extrapolation among the tested KDA range settings on S_3, S_4, and periodic waveform continuation. At 1.3B parameters and 100B tokens in language modeling, CKDA (and its attention hybrid) performs on par with KDA and outperforms Transformers and other linear RNNs. Our implementation is a minor modification of KDA recurrence kernels from FLA (Yang & Zhang, 2024) and retains 96–97% of standard KDA's throughput.

2 Background & Related Work

Linear RNNs. Linear recurrent neural networks process sequences through stacked layers with affine state updates. For one head, following Grazzi et al. (2025), we write

$$\boldsymbol{H}_i = \boldsymbol{A}(\boldsymbol{x}_i)\boldsymbol{H}_{i-1} + \boldsymbol{B}(\boldsymbol{x}_i), \qquad \hat{\boldsymbol{y}}_i = \mathrm{dec}(\boldsymbol{H}_i, \boldsymbol{x}_i), \qquad i = 1, \ldots, t. \tag{1}$$

Here $\boldsymbol{H}_i \in \mathbb{R}^{n \times d_v}$ is the recurrent state; the learned maps $\boldsymbol{A}$, $\boldsymbol{B}$, and dec specify the transition applied to its columns, additive update, and output. Architectures differ mainly in the structure imposed on $\boldsymbol{A}$. GLA (Yang et al., 2024a) uses diagonal transitions, while Mamba-2 (Dao & Gu, 2024) and mLSTM (Beck et al., 2024) use scalar-times-identity transitions within each head.

Diagonal-plus-low-rank (DPLR) transitions permit channel mixing. DeltaNet (Schlag et al., 2021; Yang et al., 2024b) uses $\boldsymbol{I} - \beta_i \boldsymbol{k}_i \boldsymbol{k}_i^\top$: for unit keys, $\beta_i = 0, 1, 2$ give identity, projection, and Householder reflection (Householder, 1958; Grazzi et al., 2025), respectively. Gated DeltaNet (Yang et al., 2025) adds scalar decay, whereas KDA (Kimi Team, 2025) uses a channel-wise positive gate, $\boldsymbol{A}_i = (\boldsymbol{I} - \beta_i \boldsymbol{k}_i \boldsymbol{k}_i^\top) \operatorname{Diag}(\boldsymbol{\alpha}_i)$; both retain diagonal-plus-rank-one transitions and efficient WY-based chunk-wise implementations. Related delta architectures introduce channel-wise learning rates and separate erase/write gates (GDN-2 by Hatamizadeh et al. (2026)), or asymmetric rank-one corrections (RWKV-7 by Peng et al. (2025)).

Complex and rotation-based recurrences. S4 (Gu et al., 2022) and LRU (Orvieto et al., 2023) use complex DPLR and complex-diagonal representations, respectively. Earlier RNNs combine nonlinear activations with unitary (Arjovsky et al., 2016) or Householder-parameterized (Mhammedi et al., 2017) orthogonal recurrent matrices, unlike the affine updates we consider. RotRNN (Biegun et al., 2024) uses rotation-based linear recurrences, while Orvieto et al. (2024) analyze the benefits of complex eigenvalues for finite-window memory reconstruction. DeltaProduct (Siems et al., 2025) realizes planar rotations using two Householder factors per token, requiring an additional rank-one update. Separately, higher-order and block-diagonal LRUs (Dubinin et al., 2026) enrich state mixing outside the delta-rule family. Selective RoPE (Movahedi et al., 2026) adds input-dependent rotations to gated attention, Mamba-3 (Lahoti et al., 2026) uses complex dynamics, and Adaptive Unitary SSMs (Karuvally et al., 2025) use input-dependent unitary transitions sharing a diagonalizing basis. MDN (Huang et al., 2026) adds an auxiliary momentum state, yielding second-order dynamics with complex-conjugate eigenvalues. Semidirect Fourier Delta Attention (Zhang, 2026) combines explicit phase/decay gates with delta updates and chunk-WY algorithms; its phases admit real 2×2 rotation implementations and realize cyclic counters even without delta updates. CKDA instead retains KDA's first-order, real diagonal-plus-rank-one recurrence: signed gates $\boldsymbol{\alpha}_i \in [-1, 1]^n$ (Sarrof et al., 2024) and $\beta_i \in [0, 2]$ (Grazzi et al., 2025) let coordinate and delta-rule reflections compose into noncommuting rotation families, without auxiliary recurrent states or explicit phase gates. See Table 3 for a comparison between CKDA, MDN, and SFDA.

Formal languages and recurrent expressivity. Finite monoids recognize exactly the regular languages; non-commutative group problems require order-sensitive composition, with S_5 linked to the computational complexity class NC^1 through permutation branching programs (Barrington, 1986). We consider every group element as an input, not only generators. Under the finite-precision assumptions of Shakerinava et al. (2026), single-layer input-dependent complex-diagonal SSMs track exactly the finite abelian groups, excluding non-abelian tracking despite complex eigenvalues. Alternative transitions use bilinear interactions (Ebrahimi & Memisevic, 2026), selective SSM parameterizations for automaton emulation (Terzic et al., 2025a), or fixed-point iterations (Movahedi et al., 2025). Our multilayer results adapt finite-state and rational weighted-automaton constructions (Siems et al., 2025; Peng et al., 2025; Merrill et al., 2026a), replacing the two-layer DeltaNet clock with one CKDA layer. Circuit-complexity bounds depend on precision and evaluation assumptions (Merrill et al., 2024; 2026a); algebraic analyses show that discretization and evaluation order can change expressivity (Nowak et al., 2026). Recent work highlights a gap between the expressive capacity of linear RNNs and their robustness in long-horizon state tracking, emphasizing limitations in error correction that can allow perturbations to accumulate and compromise state representations (Dankowiakowski & Ronca, 2025; Chung et al., 2026).

3 Motivation: From Symmetry to Rotation

We begin by isolating the mechanism through which channel-wise signed gating changes the transition geometry: We show that scalar gates commute with every matrix, whereas diagonal (channel-wise) gates can combine with the Householder update to produce rotations.

Symmetry. A KDA state-transition matrix $\boldsymbol{A} = \boldsymbol{H_k} \boldsymbol{D}$, with $\boldsymbol{H_k} := \boldsymbol{I} - \beta \boldsymbol{k}\boldsymbol{k}^\top$ and $\boldsymbol{D} := \operatorname{Diag}(\boldsymbol{\alpha})$, is the product of two symmetric matrices. $\boldsymbol{A}$ is symmetric if and only if the factors commute, since $\boldsymbol{A}^\top = \boldsymbol{D}\boldsymbol{H_k}$. Entrywise, the commutation condition $\boldsymbol{H_k}\boldsymbol{D} = \boldsymbol{D}\boldsymbol{H_k}$ becomes

$$(\boldsymbol{H_k}\boldsymbol{D} - \boldsymbol{D}\boldsymbol{H_k})_{ij} = \beta k_i k_j (\alpha_i - \alpha_j) = 0 \qquad \text{for all } i, j.$$

Scalar Gate. For Gated DeltaNet, $\alpha_i = \alpha$ for every coordinate, so the commutation condition above is always true. Hence $\boldsymbol{A} = \alpha(\boldsymbol{I} - \beta \boldsymbol{k}\boldsymbol{k}^\top)$ is symmetric and has a real spectrum. Moreover, over

several recurrent steps the scalar gates factor entirely out: let $\boldsymbol{A}_t = \alpha_t(\boldsymbol{I} - \beta_t \boldsymbol{k}_t \boldsymbol{k}_t^\top)$, then

$$\boldsymbol{A}_T \cdots \boldsymbol{A}_1 = \left(\prod_{t=1}^{T} \alpha_t\right) \left(\boldsymbol{I} - \beta_T \boldsymbol{k}_T \boldsymbol{k}_T^\top\right) \cdots \left(\boldsymbol{I} - \beta_1 \boldsymbol{k}_1 \boldsymbol{k}_1^\top\right).$$

Thus extending the scalar gate to $[-1, 1]$ can change the overall scale and introduce a global sign, but it cannot contribute an additional independently oriented transformation (Figure 2 bottom row).

Diagonal Gate. For a channel-wise gate, differences $\alpha_i - \alpha_j$ need not vanish. Whenever $\beta \neq 0$ and the key has nonzero components on two coordinates with different gate values, the two symmetric factors do not commute and the resulting transition is nonsymmetric. This property alone, however, is not sufficient to produce a non-real spectrum. For strictly positive gates, $\boldsymbol{H_k D}$ is similar to the symmetric matrix $\boldsymbol{D}^{1/2}\boldsymbol{H_k}\boldsymbol{D}^{1/2}$, so all its eigenvalues are real. The same conclusion holds when some gates are zero, by continuity. Hence standard nonnegative KDA still has a real spectrum. As we show next, allowing the diagonal gate to change sign removes this restriction.

We see the resulting geometry in a simple 2D case. Consider the KDA transition with β=2 given by

$$\boldsymbol{A}_{\alpha,\theta} = \boldsymbol{H_k D}_\alpha, \qquad \boldsymbol{H_k} := \boldsymbol{I} - 2\boldsymbol{k}\boldsymbol{k}^\top, \qquad \boldsymbol{k} = (\cos\theta, \sin\theta)^\top, \qquad \boldsymbol{D}_\alpha := \mathrm{Diag}(\alpha, 1).$$

Expanding the product gives

$$\boldsymbol{A}_{\alpha,\theta} = \begin{pmatrix} -\alpha\cos 2\theta & -\sin 2\theta \\ -\alpha\sin 2\theta & \cos 2\theta \end{pmatrix}. \tag{2}$$

Since $\boldsymbol{A}_{\alpha,\theta} \in \mathbb{R}^{2\times 2}$, its eigenvalues are non-real precisely when its discriminant is negative,

$$\Delta := \mathrm{tr}(\boldsymbol{A}_{\alpha,\theta})^2 - 4\det(\boldsymbol{A}_{\alpha,\theta}) = (1-\alpha)^2\cos^2(2\theta) + 4\alpha < 0.$$

Hence, for $\alpha \geq 0$, the spectrum is necessarily real, whereas for $\alpha < 0$, a complex-conjugate pair can exist. Whenever the eigenvalues are non-real, their product is $\det(\boldsymbol{A}_{\alpha,\theta}) = -\alpha$, and hence $|\lambda| = \sqrt{-\alpha}$. Thus, as α moves from 0 toward -1, the complex pair moves outward toward the unit circle (see Figure 2). At the endpoint $\alpha = -1$, we have $\mathrm{Diag}(-1, 1) = \boldsymbol{I} - 2\boldsymbol{e}_1\boldsymbol{e}_1^\top = \boldsymbol{H}_{\boldsymbol{e}_1}$, and hence $\boldsymbol{A}_{-1,\theta} = \boldsymbol{H_k}\boldsymbol{H}_{\boldsymbol{e}_1}$. The transition is therefore the composition of two reflections whose mirror lines differ by an angle θ. Their composition is a rotation by 2θ.

This perspective extends coordinate by coordinate. A diagonal matrix can be decomposed as

$$\mathrm{Diag}(\boldsymbol{\alpha}) = \left(\boldsymbol{I} - (1-\alpha_1)\boldsymbol{e}_1\boldsymbol{e}_1^\top\right)\left(\boldsymbol{I} - (1-\alpha_2)\boldsymbol{e}_2\boldsymbol{e}_2^\top\right) \cdots \left(\boldsymbol{I} - (1-\alpha_n)\boldsymbol{e}_n\boldsymbol{e}_n^\top\right). \tag{3}$$

Thus each factor is an axis-aligned generalized Householder transformation with normal $\boldsymbol{e}_i$ and learning rate $\beta_i = 1 - \alpha_i$. This can also be viewed as a special case of the generalized Householder composition studied by Siems et al. (2025). In particular, when $\alpha_i = -1$, the corresponding factor is an exact coordinate reflection. Hence a KDA state transition $\boldsymbol{A}_t = (\boldsymbol{I} - \beta_t \boldsymbol{k}_t \boldsymbol{k}_t^\top)\,\mathrm{Diag}(\boldsymbol{\alpha}_t)$ combines one freely oriented generalized Householder transformation with n axis-aligned ones.

Complex KDA. We define CKDA as KDA with $\boldsymbol{\alpha}_t \in [-1, 1]^n$ and $\beta_t \in [0, 2]$; our implementation uses $\beta_t = 2\sigma(b_t)$ and $\alpha_{t,i} = 2\sigma(a_{t,i}) - 1$, with $b_t, \boldsymbol{a}_t$ functions of the input $\boldsymbol{x}_t$ and σ the sigmoid function, following Sarrof et al. (2024) and Grazzi et al. (2025) with the magnitude floor and backward convention specified in Appendix E.

4 Structure and Spectrum of Complex KDA

The planar construction shows how signed gating enables rotations within a rank-one diagonal-plus-low-rank (DPLR) transition. This observation motivates the question of how far the same mechanism extends beyond two dimensions. We now study CKDA in arbitrary dimensions and show that it captures the entire family of orthogonal rank-one DPLR (DPR1) transitions. Orthogonality preserves norms, supporting persistent memory in linear RNNs and allows state-tracking.

Theorem 1 (DPR1 and CKDA)**.** *Every orthogonal DPR1 matrix $\boldsymbol{A} = \boldsymbol{D} + \boldsymbol{u}\boldsymbol{v}^\top \in \mathbb{R}^{n\times n}$ with diagonal $\boldsymbol{D}$ can be written in the form $\boldsymbol{A} = (\boldsymbol{I} - 2\boldsymbol{k}\boldsymbol{k}^\top)\boldsymbol{S}$, $\|\boldsymbol{k}\|_2 = 1$, $\boldsymbol{S} = \mathrm{Diag}(s_i)$, $s_i \in \{-1, +1\}$. We call this form a* signed-Householder matrix*, i.e. a CKDA matrix with $\alpha_i \in \{+1, -1\}$, $\beta = 2$.*

The intuition is that orthogonality forces the rank-one factors $\boldsymbol{u}$ and $\boldsymbol{v}$ to align up to coordinate signs, so after absorbing these signs into $\boldsymbol{S}$, the matrix $\boldsymbol{AS}$ becomes symmetric. The DPR1 structure then forces it to be a single Householder reflection (see Figure 3). Proof in Appendix A.1.

Table 1: **Expressivity of linear recurrent models**

	CKDA $(\boldsymbol{I}-\beta\boldsymbol{k}\boldsymbol{k}^\top)D(\boldsymbol{\alpha})$	**GDN** $\alpha(\boldsymbol{I}-\beta\boldsymbol{k}\boldsymbol{k}^\top)$	**Gated DeltaProduct** $\alpha\prod_{j=1}^{k}(\boldsymbol{I}-\beta_j\boldsymbol{k}_j\boldsymbol{k}_j^\top)$	**RWKV-7 / GDN-2** $D(\boldsymbol{\alpha})-\boldsymbol{k}(\boldsymbol{v}\odot\boldsymbol{k})^\top$
Param. ranges	$\beta\in[0,2]$, $\boldsymbol{\alpha}\in[-1,1]^d$	$\alpha\in[0,1]$, $\beta\in[0,2]$	$\alpha\in[0,1]$, $\beta_j\in[0,2]$	$\boldsymbol{\alpha}\in[0,1]^d$, $\boldsymbol{v}\in[0,2]^d$
Complex pairs	$\leq$ **1 (Thm. 8)** 0 if $\beta\leq 1$ or $\boldsymbol{\alpha}\geq 0$	0	$\leq \lfloor k/2\rfloor$	0
S_2 (parity)	✓ 1 via [U3]†	✓ 1 via [U3]	✓ 1 ($k\geq 1$) via [U3]	✓ 1 [RL1]‡
$\mathbb{Z}_n, D_n$ ($n>2$)	✓ **1 (Thm. 3)**†	✓ 2[D7] × 1 FP[U2]	✓ 1 ($k\geq 2$) [D7]	✓ 2[D7]‡ × 1 FP[U2]
S_4, A_5	✓ **1 (Thm. 3)**†	✓ 4[D1] × 1 FP[U2]	✓ 1 ($k\geq 2$) [D4]	✓ 4[D1]‡ × 1 FP[U2]
S_n ($n\geq 5$)	✓ **3 (Thm. 5)** × **1** SS **(Thm. 4)**	✓ 4[D1] × 1 FP[U2]	✓ 1: $k\geq n-1$ [D1] ✓ 3: $2\leq k\leq n-2$ [D1] × **1** SS: $k\leq 3$ **(Thm. 4)**	✓ 4[D1]‡ × 1 FP[U2]
Regular languages	✓ L_q via [D2]	✓ L_q [D2]	✓ L_q ($k\geq 1$) [D2]	✓ L_q via [D2]‡
(unstable)	✓ **3 (Thm. 5)** allow $\beta>2$	✓ 4 via [M11, P25] $\beta\in[0,3]$	✓ 3 ($k\geq 2$); 1 ($k\geq 3q-3$) $\beta\in[0,3]$ via [M11, P25]	✓ 4 [R3]
WFAs over $\mathbb{Q}$ (PP)	✓ **3 (Thm. 5)** $\beta\in[0,\infty)$	✓ 4 [M11] $\beta\in\mathbb{R}$	✓ 3 ($k\geq 2$); 1 ($k\geq B_q$) $\beta\in\mathbb{R}$ via [M11, P25]	✓ 4 [M6] unrestricted

✓ L: depth L is sufficient; × L A: depth L is impossible under A. L_q: a sufficient finite depth depending on the language's q-state DFA. Unstable / amber: expansive transitions allowed. † CKDA: $\beta\in\{0,2\}$, $\boldsymbol{\alpha}\in\{\pm 1\}^d$ ($\boldsymbol{\alpha}=\mathbf{1}$ for parity). ‡ RWKV-7/GDN-2 constructions use $D(\boldsymbol{\alpha})=\gamma\boldsymbol{I}$, $\boldsymbol{v}=\gamma\beta\mathbf{1}$, with $\gamma\in[0,1]$, $\beta\in[0,2]$ (stable: $\|\boldsymbol{A}\|_2\leq 1$); reflections use $\gamma=1$, $\beta=2$. RWKV-7/GDN-2: unit keys, nonnegative gates (RWKV-7's factor $c=2$ is absorbed into $\boldsymbol{v}$). Both allow expansive transitions by default; GDN-2 entries use its extended erase range $[0,2]^d$. $D(\boldsymbol{\alpha})=\mathrm{Diag}(\boldsymbol{\alpha})$; q: automaton size; $B_q=7q^2+5q+1$ ($2q+1$ coordinates). Group inputs: all elements. FP: fixed precision under the cited result's convention. PP: polynomial bit length over a fixed algebraic number field (App. B.5). SS: bounded exact affine states with states separability (Def. B.5).
U (Grazzi et al., 2025), D (Siems et al., 2025), R (Peng et al., 2025), M (Merrill et al., 2026a): [D1]/[RL1]: Theorem/Lemma 1; "via": derived bounds. P25: our factor bounds. Comparison details: App. D.5.

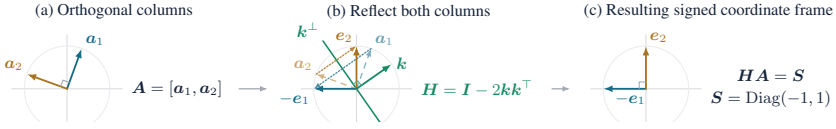

Figure 3: 2D case: Let $\boldsymbol{A}$ be any orthogonal 2D matrix. A single reflection aligns a 2D orthogonal frame with the coordinate axes, giving $\boldsymbol{H}\boldsymbol{A}=\boldsymbol{S}$ and hence $\boldsymbol{A}=\boldsymbol{H}\boldsymbol{S}$.

Thus replacing KDA's structured rank-one term by a non-symmetric $\boldsymbol{u}\boldsymbol{v}^\top$, such as in RWKV-7/GDN-2, adds no orthogonal transitions. The next result generalizes the 2D case of the motivation section to higher dimensions: each CKDA transform can be viewed as an element-wise scaling followed by a transform which can be a rotation only in a 2D subspace.

Proposition 2 (Sign-magnitude decomposition)**.** *Let* $\|\boldsymbol{k}\|_2=1$, $\beta\in[0,2]$, *and* $\alpha_i\in[-1,1]$. *Write* $\boldsymbol{A}=(\boldsymbol{I}-\beta\boldsymbol{k}\boldsymbol{k}^\top)\,\mathrm{Diag}(\boldsymbol{\alpha})=\widetilde{\boldsymbol{A}}\,\mathrm{Diag}(|\boldsymbol{\alpha}|)$, *where* $\widetilde{\boldsymbol{A}}=(\boldsymbol{I}-\beta\boldsymbol{k}\boldsymbol{k}^\top)\boldsymbol{S}$ *and* $\boldsymbol{S}=\mathrm{Diag}(\mathrm{sign}\,\boldsymbol{\alpha})$, *choosing* $\mathrm{sign}(0)=1$ *for this factorization. Let* $E_\pm=\ker(\boldsymbol{S}\mp\boldsymbol{I})$ *and decompose* $\boldsymbol{k}=\boldsymbol{k}_-+\boldsymbol{k}_+$ *with* $\boldsymbol{k}_\pm\in E_\pm$, $\|\boldsymbol{k}_-\|=\cos\theta$, *and* $\|\boldsymbol{k}_+\|=\sin\theta$, $\theta\in[0,\pi/2]$. *When both components are nonzero, the restriction of* $\widetilde{\boldsymbol{A}}$ *to* $\mathcal{U}=\mathrm{span}\{\boldsymbol{k}_-,\boldsymbol{k}_+\}$, *in the orthonormal basis* $\boldsymbol{k}_-/\|\boldsymbol{k}_-\|,\boldsymbol{k}_+/\|\boldsymbol{k}_+\|$, *is*

$$\boldsymbol{B}(\beta,\theta)=\begin{pmatrix}\beta\cos^2\theta-1 & -\beta\sin\theta\cos\theta\\ \beta\sin\theta\cos\theta & 1-\beta\sin^2\theta\end{pmatrix}.$$

On $\mathcal{U}^\perp$, $\widetilde{\boldsymbol{A}}$ *agrees with* $\boldsymbol{S}$. *The block is a rotation by* 2θ *when* $\beta=2$, *and its eigenvalues are non-real exactly when* $\beta^2\cos^2(2\theta)<4(\beta-1)$. *If either component vanishes,* $\widetilde{\boldsymbol{A}}$ *has only real eigenvalues.*

Therefore, after the initial coordinate-wise scaling, a 2D rotation with complex eigenvalues can occur only if (i) at least two gate coordinates differ in sign, (ii) the key vector spans coordinates with opposite signs and (iii) $\beta>1$. The 2D example in the previous section fits exactly these criteria.

The analysis of the full CKDA transition adds complexity and requires a separate argument: Theorem 8 in the appendix proves that any CKDA matrix also has at most one non-real conjugate eigenvalue pair, requiring both $\beta>1$ and a negative gate entry. Thus both range extensions are required for non-real eigenvalues. Moreover, this restriction extends beyond CKDA: every non-expansive DPR1 matrix has at most one non-real conjugate eigenvalue pair on the unit circle, counted with

algebraic multiplicity (Theorem 9). Products of CKDA transitions can represent any square non-expansive matrix, despite the one-plane restriction on each individual transition. Generalized Householder products already have this universality (Grazzi et al., 2025, Prop. 1); Proposition 10 shows that CKDA needs at most $\max\{1, 2n-2\}$ factors (vs $3n$ for products of Householders) in dimension n. For orthogonal matrices, $\max\{1, n-1\}$ factors (vs n for products of Householders) suffice and this bound is sharp. In contrast, products of RoPE (Su et al., 2024) and Selective RoPE (Movahedi et al., 2026) rotations remain block-diagonal with 2×2 blocks and can't thus model general $n \times n$ matrices.

5 State-Tracking Expressivity

Tracking a state-transition system. Let $\mathcal{S}$ be a state space with initial state s_0 and an update $T_a : \mathcal{S} \to \mathcal{S}$ for each input symbol a. A recurrent model *tracks* this system if a fixed decoder f recovers its state from the model's hidden state h_t for every input sequence: $s_t = T_{a_t}(s_{t-1}), (t \geq 1), f(h_t) = s_t, (t \geq 0)$. Several hidden states may decode to the same target state. The systems we consider include finite-group products, deterministic automata, and weighted finite automata (WFAs). Our constructions allow sufficiently expressive feed-forward decoders; our arithmetic and precision conventions are defined in Appendix B.5. A contextualized summary of our state-tracking results is in Table 1.

Single-layer expressivity. Consider a finite group G. A *faithful orthogonal representation* assigns each group element a distinct orthogonal matrix, so that composing group elements corresponds exactly to multiplying their matrices. We call a construction that uses these matrices directly as its transitions a *realization*, and write d for their dimension. Tracking does not require a faithful representation: a many-to-one decoder can map distinct hidden states to the same group element, and the input transitions need not themselves form a representation. Any finite group is isomorphic to a subgroup of a permutation group and can be realized by one Linear RNN layer using permutation matrices as transitions. However, CKDA cannot model arbitrary permutations in a single transition. We consider the case where inputs range over *all* of G. If inputs are restricted to identity and swaps, one-layer DeltaNet suffices (Grazzi et al., 2025).

Theorem 3 (Single-layer finite-group expressivity)**.** *A single CKDA layer (one head) tracks every finite group isomorphic to a subgroup of $SO(3)$. Specifically, it realizes every finite cyclic ($\mathbb{Z}_n$) and dihedral group (D_n) in $d = 2$ and A_4, S_4 in $d = 3$, and tracks A_5 in $d = 4$. These constructions use orthogonal transitions and a fixed exact datatype.*

Proof sketch. The planar rotations from the motivation section, together with reflections, give the cyclic and dihedral groups. Every three-dimensional rotation is a product of two Householder reflections, but, apart from the identity and $180°$ rotations, CKDA requires the rotation axis to have a zero coordinate. Reorienting the cube satisfies this condition for S_4 and its subgroup A_4, whereas no orientation works for the icosahedral rotations of A_5. In four dimensions, however, A_5 can be tracked using a many-to-one decoder. See Appendices C.1 to C.4 for the constructions and proofs.

Limits of a single CKDA layer. Increasing dimension and allowing arbitrary decoders do not remove every obstruction. Under boundedness and states separability, one-layer S_5 tracking remains impossible for CKDA and (Gated) DeltaProduct$_k$ with $k \leq 3$. *States separability* requires a uniform positive gap between differently decoded states. Without it, a decoder cannot tolerates an arbitrarily small observation error on some input sequences.

Theorem 4 (Spectral obstruction to S_5 tracking)**.** *Suppose every head transition $\boldsymbol{A}$ satisfies $\|\boldsymbol{A}\|_2 \leq 1$ and has at most one non-real conjugate eigenvalue pair on the unit circle, counted with algebraic multiplicity. Then a single layer of exact affine updates with any finite number of independent heads cannot track all S_5 inputs with bounded reachable states and states separability.*

Proof sketch. Our finite-replacement theorem (Theorem 22) removes contracting directions and additive terms from the original layer, then produces orthogonal transitions generating a finite group mapping onto S_5. This preserves headwise spectral bounds without increasing dimensions. From a five-cycle and its conjugate by a transposition, we construct two transition-matrix products $\boldsymbol{R}_1$ and $\boldsymbol{R}_2$ whose blocks in each head are either identities or planar rotations with the same angle and order divisible by five. Their commutator $\boldsymbol{C} = \boldsymbol{R}_1\boldsymbol{R}_2\boldsymbol{R}_1^{-1}\boldsymbol{R}_2^{-1}$ satisfies $\boldsymbol{C}^{10} = \boldsymbol{I}$. Yet it decodes to a three-cycle, whose tenth power is not the identity, a contradiction. Appendix C.5 gives the argument.

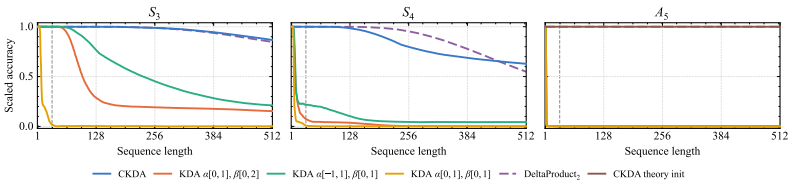

Figure 5: One-layer KDA variants on the S_3, S_4, and A_5 word problems, by the eigenvalue range allowed for the diagonal and for the Householder component. Combining a signed gate with a reflection-enabled Householder update gives the strongest extrapolation on S_3 and S_4 among the tested KDA range settings; S_4 accuracy still declines at long lengths. The A_5 result uses the separate theory-initialized setup described in the text (see Figure 17 for baseline results).

Multi-layer expressivity. The same planar rotations that strengthen one-layer tracking also reduce the depth needed to simulate general state-transition systems.

Theorem 5 (Multi-layer expressivity)**.** *Three CKDA layers solve every finite group-word problem with a fixed exact datatype. If $\beta > 2$ is allowed, three layers also recognize every regular language with a fixed exact datatype and compute every WFA over $\mathbb{Q}$ in polynomial precision using exact arithmetic over a fixed algebraic number field (Appendices B.5 and D).*

We adapt the existing clock–buffer–accumulator construction (Peng et al., 2025; Siems et al., 2025; Merrill et al., 2026a). A clock tracks position modulo a fixed period to schedule updates. The buffer factors each block of transitions; the accumulator applies one factor per step, and a completion readout corrects the delay. CKDA implements the clock with a planar rotation in one layer, saving one layer relative to DeltaNet/GDN which uses 4 layers for the same result.

6 Experiments

We evaluate whether CKDA's added expressivity improves state tracking and periodic waveform extrapolation, and assess its language-modeling performance and computational efficiency.

Implementation. KDA stores gate magnitudes in log-space, so gates allowing negative values require separating signs from magnitudes. We implement the same sign transformation in three ways: (1) compiled PyTorch operations that absorb cumulative signs into keys and queries, without changing kernels (gauge); (2) Triton (Tillet et al., 2019) kernels that fuse these signs into normalization and its backward pass; and (3) a TileLang-backed (Wang et al., 2026) hybrid that combines these Triton changes with TileLang backward kernels. The kernel implementations retain approximately 96–97% of KDA throughput (Figure 4); Appendix E gives the derivation, implementation details, and benchmark protocol. The ungated DeltaProduct$_2$ baseline achieves competitive throughput while performing two delta-rule updates per token, each with its own value: its measured throughput includes processing an additional value input relative to CKDA. This is due to its identity-plus-rank-one factors, which avoid CKDA's channel-wise gating computations; the gated variant adds only scalar decay. We therefore view CKDA as an alternative way to obtain expressivity similar to DeltaProduct$_2$ but using channel-wise gates, without claiming inherent efficiency gains over it. We also use *spread* gate initialization, with approximately equal proportions of positive and negative entries, alongside the original, positive-only *standard* initialization (In Appendix G.1, we also propose a variant for β). Finally, the SiLU activation inherited from DeltaNet (Yang et al., 2024b) on key and query projections can bias keys toward positive entries (Appendix F). We remove SiLU for KDA state-tracking experiments and ablate this choice for language modeling.

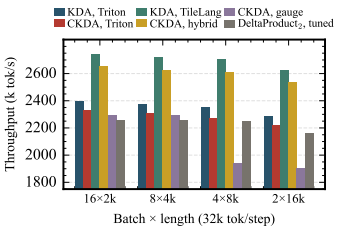

Figure 4: Forward–backward kernel throughput on an H100 in BF16, with 16 heads, $d_k = d_v = 128$, and 32k tokens per step. DeltaProduct$_2$ uses no forget gate. Implementation and timing details are in Appendix E.

1. State-Tracking. We train one-layer models on S_3, S_4, and A_5 group word problems (Merrill et al., 2024; Terzic et al., 2025b; Movahedi et al., 2025), predicting cumulative products from inputs spanning each group. Training lengths reach 32; report the best of three seeds (standard for this difficult to learn task) and scale accuracy from chance (0) to perfect (1). Training details are in Appendix G.2. CKDA ($\boldsymbol{\alpha} \in [-1, 1]^n$, $\beta \in [0, 2]$) extrapolates well on S_3 and S_4 while other settings of

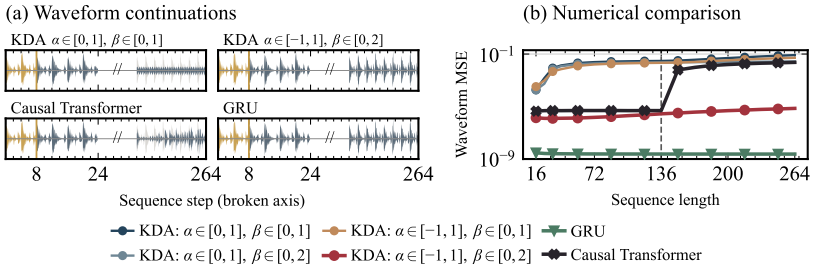

Figure 6: Periodic waveform continuation with single-layer models. (a) Predictions shortly after the half-bar cue and far beyond the training horizon; gold marks the cue, grey the target, and dark blue the prediction. (b) Waveform MSE versus sequence length; the dashed line marks the maximum training length (136). CKDA extrapolates accurately where the other KDA variants and causal Transformer fail, while the GRU achieves the lowest error.

Table 2: Language modeling and zero-shot common-sense reasoning at 1.3B parameters on 100B tokens of FineWeb-Edu in comparison to values from Hatamizadeh et al. (2026).

	ppl ↓		accuracy (%) ↑										recall (%) ↑		
Model	Wiki.	LMB.	LMB.	PIQA	Hella.	Wino.	ARC-e	ARC-c	OBQA	SIQA	BoolQ	Avg.	SQuAD	SWDE	FDA
Recurrent models															
Mamba-2	16.79	12.38	45.24	72.58	55.51	55.33	70.68	35.26	31.00	40.63	60.19	51.82	–	–	–
Gated DeltaNet	16.40	11.89	49.62	72.31	56.50	56.75	68.81	35.15	30.20	40.53	58.78	52.07	–	–	–
KDA	16.81	11.68	48.13	72.09	55.75	55.72	70.83	35.92	30.40	40.99	60.67	52.28	–	–	–
Mamba-3 (SISO)	16.30	12.99	45.06	72.31	55.58	56.20	70.45	34.56	31.00	**41.76**	55.90	51.42	–	–	–
Mamba-3 (MIMO)	16.45	11.66	47.82	72.36	56.49	55.78	72.38	38.07	30.00	40.89	57.74	52.39	–	–	–
Gated DeltaNet-2	15.90	11.41	48.09	72.80	56.84	57.85	72.43	38.23	**31.60**	40.58	59.54	53.11	–	–	–
KDA, bounded gate (ours)	**15.73**	10.53	50.46	72.85	**59.32**	**61.33**	73.23	38.40	28.80	41.35	**61.10**	**54.09**	38.17	**49.59**	30.40
CKDA (ours)	15.78	**10.08**	**51.66**	**73.12**	59.23	58.88	**73.99**	**39.08**	28.60	41.61	60.34	54.06	**39.68**	48.42	**31.49**
Attention or hybrid models															
Transformer (SWA-hybrid)	19.22	13.72	48.32	70.21	56.12	55.85	69.23	33.84	25.00	39.74	59.42	50.86	–	–	–
Mamba-2 (hybrid)	17.46	11.29	48.05	71.47	57.52	56.17	70.50	34.73	29.80	40.35	59.31	51.99	–	–	–
Gated DeltaNet (hybrid)	16.00	10.82	48.71	70.06	57.50	56.83	70.41	35.15	30.60	40.97	60.00	52.25	–	–	–
KDA (hybrid)	16.01	10.66	49.21	71.06	56.89	57.77	71.59	35.07	30.00	40.53	62.03	52.68	–	–	–
Mamba-3 (SISO, hybrid)	15.54	10.65	49.19	71.01	58.75	57.30	70.54	36.35	32.00	41.20	57.86	52.69	–	–	–
Mamba-3 (MIMO, hybrid)	15.81	10.92	49.82	71.98	58.19	57.06	70.54	**38.48**	29.40	40.99	57.98	52.72	–	–	–
Gated DeltaNet-2 (hybrid)	15.62	10.43	50.90	72.20	58.46	58.56	71.89	36.69	**33.00**	41.50	62.57	53.97	–	–	–
KDA + attn 3:1 (ours)	**15.04**	9.93	51.60	**73.88**	59.57	**60.62**	**72.69**	37.20	27.00	42.43	60.31	53.92	**42.69**	**64.81**	**64.79**
CKDA + attn 3:1 (ours)	15.34	**9.90**	**52.57**	73.56	**59.68**	60.14	72.56	36.86	28.20	**42.58**	**63.15**	**54.37**	37.77	63.91	58.89

KDA fail (Figure 5; Figure 17 in Appendix I), consistent with Theorem 3. Extending only the gate or β yields long-length S_3 scaled accuracy near 0.2, matching parity-only discrimination of the two A_3 cosets. A successful CKDA head learns $\beta \approx 2$, nearly sign-valued gates, and complex-conjugate eigenvalues near the unit circle (Figure 14), recovering the mechanism of Section 3 and Proposition 2. Standard training fails to learn A_5, while a smaller, fully trainable model initialized near our quaternion construction (Appendix C.4.2) achieves length extrapolation. Training retains next-state cross-entropy and a standard MLP readout, consistent with representation discovery being an optimization obstacle. The architecture and training schedule also differ, so this does not isolate the effect of initialization.

2. Audio continuation. To test whether complex eigenvalues help preserve phase beyond symbolic state tracking, we train single-layer models to continue a synthetic periodic groove after a half-bar cue, followed by zero inputs. Whereas the autoregressive audio experiments of Mamba-1 (Gu & Dao, 2024) and SaShiMi (Goel et al., 2022) use prediction feedback that can induce nonlinear state dynamics, we compute the continuation in a single forward pass without output feedback. The task requires inferring phase from the cue and maintaining a periodic continuation after the cue ends. Among the four KDA range settings, only CKDA with both range extensions accurately continues the waveform (Figure 6). At length 264, beyond the maximum training length of 136, it retains 38.1 dB SNR, while the causal Transformer falls to 2.8 dB. Audio samples are available online. See Appendix G.3 for experimental details. These results connect learned rotational dynamics to periodic extrapolation. The GRU (Cho et al., 2014), a nonlinear RNN, is more accurate on this task.

3. Language modeling. We first train small language models of 340 million parameters on 15 billion tokens from the Nemotron-CC (Su et al., 2025) dataset, using CKDA as the transformer token mixer along with softmax/dense attention, DeltaProduct, GDN, and KDA as baselines. For architectures that introduce more projections, we reduce the latent dimension of the SwiGLU MLP

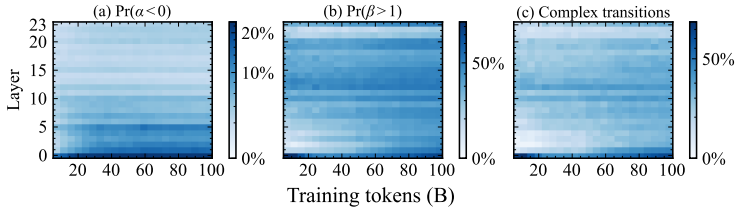

Figure 7: Extended-range use in non-hybrid CKDA 1.3B with standard initialization: per-layer fractions of negative gates (a), $\beta > 1$ (b), and transitions with complex eigenvalues (c). (a) and (c) emerge during training. Negative gates occur most frequently in the early layers while $\beta > 1$ in every layer, but particularly in the first and second. Statistics over the validation split of FineWeb-Edu at sequence length 4096. Colors use square-root scaling. Hybrid CKDA results in Figure 20, statistics over the final checkpoint across all layers in Figure 21.

by 256 to ensure comparable parameter counts. Detailed configurations are provided in Table 4. In the Nemotron-CC block of Table 7, CKDA with $\boldsymbol{\alpha} \in [-1, 1]$, $\beta \in [0, 2]$, and spread initialization achieves the highest observed average downstream accuracy: 52.30%, versus 51.32% for standard KDA. We provide additional results for models trained on 45B tokens of FineWeb in Table 7.

To compare CKDA to KDA at larger scale, we train 1.3B-parameter models on 100B tokens of FineWeb-Edu (Lozhkov et al., 2024), following the training recipe of GDN-2 by Hatamizadeh et al. (2026). Our recurrent and hybrid CKDA models achieve similar average downstream accuracy to the corresponding KDA controls and the reported Mamba-3 (Lahoti et al., 2026) and GDN-2 (Hatamizadeh et al., 2026) baselines (Table 2). The hybrid recipes differ: ours use full gated attention (Qiu et al., 2025) without RoPE at a $3 : 1$ recurrent-to-attention ratio, following Kimi Team et al. (2026), whereas the reported hybrid baselines use a $1 : 1$ ratio with sliding-window attention. Our KDA controls also use the safe sigmoid gate of Kimi Team et al. (2026), which may contribute to the differences from the KDA results as reported in Hatamizadeh et al. (2026). Comparing architecture scaling laws trained on Nemotron-CC, CKDA (and KDA) outperform a Transformer baseline across scales (Ajroldi et al., 2026), see Figure 15. We don't see a crossing point favoring Transformers going to larger over-training ratios (data to model parameter ratio) as observed for xLSTM in Beck et al. (2026), although small model behavior indicates a slight advantage reduction.

To test whether the mechanism of Section 3 also emerges in language modeling, we track the fraction of negative gate entries, fraction of $\beta > 1$, and of complex transitions in every layer of CKDA during training of the 1.3B parameter models (Figure 7). We find that despite the standard initialization keeping the gates close to 1, the model learns to use the negative gates early on during training primarily in the first layers. Similarly, the model learns to use the extended β range, first in the initial layers of the model but then also in deeper layers. The combination leads to complex eigenvalue pairs in the state-transition primarily in the first two layers of the trained model. We leave a mechanistic analysis of how the gate and β are being used in the model for future work.

7 Conclusion

We showed that channel-wise gates in $[-1, 1]$ and $\beta \in [0, 2]$ enable planar rotations within a single non-expansive diagonal-plus-rank-one KDA transition. CKDA captures every orthogonal matrix of this form and strengthens state-tracking expressivity. Among tested KDA settings, it extrapolates best on S_3, S_4, and periodic waveforms, with comparable downstream accuracy at 1.3B parameters and throughput close to standard KDA. Learned transitions exhibit complex eigenvalues in both state tracking and language modeling. **Limitations.** Our constructions use exact arithmetic and endpoint parameters; they establish expressivity without guarantees of learnability. CKDA did not learn A_5 under our training setup. In addition, the general WFA construction requires $\beta > 2$, sacrificing guaranteed non-expansiveness, although this is the case also for GDN/KWKV-7/DeltaProduct/GDN-2. CKDA offers an alternative route to the expressivity capabilities established for DeltaProduct$_2$ through channel-wise gates in $[-1, 1]$ and a single rank-one correction. This structural distinction does not establish an inherent computational advantage over (Gated) DeltaProduct$_2$, which achieves competitive throughput in our kernel benchmarks. **Future Work.** Future work could combine CKDA with the separate gates of GDN-2 (Hatamizadeh et al., 2026) that control the additive term and the forgetting of the previous state. While we show that the extended gate and β ranges are learned to be used during language model training, their functional roles remain unclear.

AI USE STATEMENT

The central idea of extending KDA's gate to signed values and combining it with an extended range for the Householder coefficient originated with the authors independently of AI assistance after reading the Kimi K3 report. This included the hypothesis that the combination could produce complex eigenvalues, which was subsequently developed with assistance from Claude Opus 4.8. The change of variables described in Appendix E, which supports gates with negative values without modifying the existing KDA kernels, was proposed by Claude Opus 5. The authors contextualized and implemented this idea in relation to similar changes of variables used by other models. The integration into the chunk-wise parallel form was proposed by the authors and implemented by the same model. The TileLang kernels were modified using ChatGPT Astra (medium) and then checked through consistency with the naive Python recurrence and the state-tracking experiments by the authors. The orthogonal DPLR characterization (Theorem 1) and norm-constrained DPLR result (Theorem 9) were developed with assistance from ChatGPT 5.6 Sol (High). ChatGPT 5.6 Sol and 6 Astra produced the first complete versions of the proofs for the single-layer group-word-problem expressivity results (Theorems 3 and 4; proofs in Appendix C). The authors subsequently checked, corrected, and substantially edited these proofs. The adaptation in Theorem 5 and Appendix D showing that three layers suffice to recognize general weighted finite automata, based on (Merrill et al., 2026a), was proposed by ChatGPT 5.6 Sol and edited with assistance from Claude Opus 5 and ChatGPT 6 Astra. Claude Opus 5 was used to assist in implementation of language modeling adaptations, implementations, and experimentation. The authors reviewed all AI-assisted work and take responsibility for the final content.

REPRODUCIBILITY STATEMENT

Our code is based on Flash-Linear-Attention (Yang & Zhang, 2024) and available here `https://anonymous.4open.science/r/ComplexKDA-BE14/README.md`. The repository contains instructions to reproduce the state-tracking, audio, and language modeling experiments. Additionally, in Appendix Section G.3, we describe in detail the training procedure for waveform continuation. In Appendix Section G.4 we describe the language modeling experiment details, with the scaling sizes, hyperparameters and scaling law fitting procedure described in Appendix H.

For the theoretical results, Appendix A provides proofs of the orthogonal DPR1 characterization, the spectral bounds for CKDA and non-expansive DPR1 matrices, and the factorization bounds for products of CKDA transitions. Appendix B specifies the state-tracking model, assumptions, and arithmetic and precision conventions. Appendix C gives the single-layer constructions and proofs, including the A_5 construction and dimension obstruction, the S_5 impossibility result (Appendix C.5), and the finite-replacement theorem (Appendix C.6). Appendix D proves the three-layer expressivity result and provides the supporting clock construction and accumulator factorizations. Appendix E proves the sign transformation used to implement signed gates with existing KDA kernels.

REFERENCES

Niccolò Ajroldi, Diana Alexandra Onutu, Haider Al-Tahan, Jörg Franke, Sampo Pyysalo, Jenia Jitsev, and Aaron Klein. Deriving Scaling Laws for OpenEuroLLM Models: Learning Rate, Batch Size and Loss, August 2026. URL `http://arxiv.org/abs/2608.28308`. arXiv:2608.28308 [cs.LG].

Jinpeng An, Jiu-Kang Yu, and Jun Yu. On the dimension datum of a subgroup and its application to isospectral manifolds. *Journal of Differential Geometry*, 94(1):59–85, 2013. URL `https://archive.ymsc.tsinghua.edu.cn/pacm_download/21/970-JDG-94-1-a02-an.pdf`.

Martin Arjovsky, Amar Shah, and Yoshua Bengio. Unitary evolution recurrent neural networks. In *International conference on machine learning*, pp. 1120–1128. PMLR, 2016.

Simran Arora, Brandon Yang, Sabri Eyuboglu, Avanika Narayan, Andrew Hojel, Immanuel Trummer, and Christopher Ré. Language models enable simple systems for generating structured views of heterogeneous data lakes. *arXiv preprint arXiv:2304.09433*, 2023.

David A Barrington. Bounded-width polynomial-size branching programs recognize exactly those languages in NC^1. In *Proceedings of the eighteenth annual ACM symposium on Theory of computing*, pp. 1–5, 1986.

Maximilian Beck, Korbinian Pöppel, Markus Spanring, Andreas Auer, Oleksandra Prudnikova, Michael Kopp, Günter Klambauer, Johannes Brandstetter, and Sepp Hochreiter. xLSTM: Extended long short-term memory. *Advances in Neural Information Processing Systems*, 37: 107547–107603, 2024.

Maximilian Beck, Kajetan Schweighofer, Sebastian Böck, Sebastian Lehner, and Sepp Hochreiter. xLSTM Scaling Laws: Competitive Performance with Linear Time-Complexity. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=bpbU549sSg.

Kai Biegun, Rares Dolga, Jake Cunningham, and David Barber. RotRNN: Modelling long sequences with rotations. *arXiv preprint arXiv:2407.07239*, 2024.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. PIQA: Reasoning about physical commonsense in natural language. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7432–7439, Apr. 2020. doi: 10.1609/aaai.v34i05.6239. URL https://ojs.aaai.org/index.php/AAAI/article/view/6239.

Armand Borel and Jean-Pierre Serre. Sur certains sous-groupes des groupes de Lie compacts. *Commentarii Mathematici Helvetici*, 27:128–139, 1953. doi: 10.1007/BF02564557.

Dan Busbridge, Amitis Shidani, Floris Weers, Jason Ramapuram, Etai Littwin, and Russell Webb. Distillation Scaling Laws. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=1nEBAkpfb9.

Yutian Chen, Zhiyuan Li, Yucheng Wang, and Ming Wei. FlashKDA: Flash Kimi Delta Attention. https://github.com/MoonshotAI/FlashKDA, 2026.

Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, eighth workshop on syntax, semantics and structure in statistical translation*, pp. 103–111, 2014.

Jihyun Choi and Jae-Hyouk Lee. Binary icosahedral group and 600-cell. *Symmetry*, 10(8):326, 2018.

Jiwan Chung, Heechan Choi, and Seon Joo Kim. Rethinking state tracking in recurrent models through error control dynamics. *arXiv preprint arXiv:2605.07755*, 2026.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? Try ARC, the AI2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018. URL https://arxiv.org/abs/1803.05457.

Keith Conrad. Simplicity of A_n. Expository notes, University of Connecticut, n.d. URL https://kconrad.math.uconn.edu/blurbs/grouptheory/Ansimple.pdf. Accessed September 14, 2026.

Federico Danieli, Pau Rodriguez, Miguel Sarabia, Xavier Suau, and Luca Zappella. ParaRNN: Unlocking parallel training of nonlinear RNNs for large language models. In *The Fourteenth International Conference on Learning Representations*, 2026.

Adam Dankowiakowski and Alessandro Ronca. Metric automata theory: A unifying theory of RNNs. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Tri Dao and Albert Gu. Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality. In *Forty-first International Conference on Machine Learning*, 2024.

Gianna M Del Corso, Federico Poloni, Leonardo Robol, and Raf Vandebril. When is a matrix unitary or hermitian plus low rank? *Numerical Linear Algebra with Applications*, 26(6):e2266, 2019.

Igor Dubinin, Antonio Orvieto, and Felix Effenberger. Improved state mixing in higher-order and block diagonal linear recurrent networks. *arXiv preprint arXiv:2602.12021*, 2026.

Reza Ebrahimi and Roland Memisevic. Revisiting bi-linear state transitions in recurrent neural networks. *Advances in Neural Information Processing Systems*, 38:69615–69642, 2026.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. The language model evaluation harness, 07 2024. URL https://zenodo.org/records/12608602.

Karan Goel, Albert Gu, Chris Donahue, and Christopher Ré. It's raw! audio generation with state-space models. In *International conference on machine learning*, pp. 7616–7633. PMLR, 2022.

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

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *First Conference on Language Modeling*, 2024.

Albert Gu, Karan Goel, and Christopher Re. Efficiently modeling long sequences with structured state spaces. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=uYLFoz1vlAC.

Ali Hatamizadeh, Yejin Choi, and Jan Kautz. Gated DeltaNet-2: Decoupling erase and write in linear attention. *arXiv preprint arXiv:2605.22791*, 2026.

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, March 2022. URL http://arxiv.org/abs/2203.15556. arXiv:2203.15556 [cs].

Alston S Householder. Unitary triangularization of a nonsymmetric matrix. *Journal of the ACM (JACM)*, 5(4):339–342, 1958.

Yulong Huang, Xiang Liu, Hongxiang Huang, Xiaopeng Lin, Zunchang Liu, Xiaowen Chu, Zeke Xie, and Bojun Cheng. MDN: Parallelizing stepwise momentum for delta linear attention. *arXiv preprint arXiv:2605.05838*, 2026.

inclusionAI. Ling-3.0-flash. Hugging Face model card, 2026. URL https://huggingface.co/inclusionAI/Ling-3.0-flash. Accessed: 2026-09-01.

Eugen J. Ionascu. Rank-one perturbations of diagonal operators. *Integral Equations and Operator Theory*, 39(4):421–440, 2001. ISSN 1420-8989. doi: 10.1007/BF01203323. URL https://doi.org/10.1007/BF01203323.

Emmanuel Jeandel. Topological automata. *Theory of Computing Systems*, 40(4):397–407, 2007. doi: 10.1007/s00224-006-1314-y.

Keller Jordan, Yuchen Jin, Vlado Boza, Jiacheng You, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks, 2024. URL https://kellerjordan.github.io/posts/muon/.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling Laws for Neural Language Models. *arXiv:2001.08361 [cs, stat]*, January 2020. URL http://arxiv.org/abs/2001.08361. arXiv: 2001.08361.

Arjun Karuvally, Franz Nowak, T. Anderson Keller, Carmen Amo Alonso, Terrence Sejnowski, and Hava T Siegelmann. Bridging expressivity and scalability with adaptive unitary SSMs. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=s4zitEu2R8.

Kimi Team. Kimi Linear: An expressive, efficient attention architecture. Technical report, Moonshot AI, 2025. arXiv:2510.26692.

Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, Jianfeng Cai, Xinyuan Cai, Peizhou Cao, Yuxuan Cao, Ziwei Chai, Y Charles, et al. Kimi K3: Open frontier intelligence. *arXiv preprint arXiv:2607.24653*, 2026.

Aakash Lahoti, Kevin Li, Berlin Chen, Caitlin Wang, Aviv Bick, J Zico Kolter, Tri Dao, and Albert Gu. Mamba-3: Improved sequence modeling using state space principles. In *The Fourteenth International Conference on Learning Representations*, 2026.

Hector J. Levesque, Ernest Davis, and Leora Morgenstern. The Winograd schema challenge. In *Proceedings of the Thirteenth International Conference on Principles of Knowledge Representation and Reasoning*, pp. 552–561, 2012. URL https://cdn.aaai.org/ocs/4492/4492-21843-1-PB.pdf.

Colin Lockard, Prashant Shiralkar, and Xin Luna Dong. Openceres: When open information extraction meets the semi-structured web. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 3047–3056, 2019.

Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Anton Lozhkov, Loubna Ben Allal, Leandro von Werra, and Thomas Wolf. Fineweb-edu: the finest collection of educational content, 2024. URL https://huggingface.co/datasets/HuggingFaceFW/fineweb-edu.

Johan Ernest Mebius. A matrix-based proof of the quaternion representation theorem for four-dimensional rotations. *arXiv preprint math/0501249*, 2005.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *International Conference on Learning Representations*, 2017.

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

William Merrill, Hongjian Jiang, Yanhong Li, Anthony Widjaja Lin, and Ashish Sabharwal. Why are linear RNNs more parallelizable? In *Forty-third International Conference on Machine Learning*, 2026a.

William Merrill, Yanhong Li, Tyler Romero, Anej Svete, Caia Costello, Pradeep Dasigi, Dirk Groeneveld, David Heineman, Bailey Kuehl, Nathan Lambert, Chuan Li, Kyle Lo, Saumya Malik, D. J. Matusz, Benjamin Minixhofer, Jacob Morrison, Luca Soldaini, Finbarr Timbers, Pete Walsh, Noah A. Smith, Hannaneh Hajishirzi, and Ashish Sabharwal. OLMo Hybrid: From theory to practice and back. In *Third Conference on Language Modeling*, 2026b.

Zakaria Mhammedi, Andrew Hellicar, Ashfaqur Rahman, and James Bailey. Efficient orthogonal parametrisation of recurrent neural networks using Householder reflections. In *International Conference on Machine Learning*, pp. 2401–2409. PMLR, 2017.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2381–2391, 2018. doi: 10.18653/v1/D18-1260. URL https://aclanthology.org/D18-1260/.

Mayank Mishra, Shawn Tan, Ion Stoica, Joseph Gonzalez, and Tri Dao. M2RNN: Non-linear RNNs with matrix-valued states for scalable language modeling. *arXiv preprint arXiv:2603.14360*, 2026.

Sajad Movahedi, Felix Sarnthein, Nicola Muca Cirone, and Antonio Orvieto. Fixed-point RNNs: Interpolating from diagonal to dense. *Advances in Neural Information Processing Systems*, 38: 44873–44908, 2025.

Sajad Movahedi, Timur Carstensen, Arshia Afzal, Frank Hutter, Antonio Orvieto, and Volkan Cevher. Selective rotary position embedding. In *International Conference on Learning Representations (ICLR)*, 2026.

Yoshihiro Nakamura. One-dimensional perturbations of isometries. *Integral Equations and Operator Theory*, 9(2):286–294, 1986.

Franz Nowak, Ryan Cotterell, and Reda Boumasmoud. An algebraic view of the expressivity of recurrent language models. *arXiv preprint arXiv:2606.01765*, 2026.

Marc Olive. Effective computation of so (3) and o (3) linear representation symmetry classes. *Mathematics and Mechanics of Complex Systems*, 7(3):203–237, 2019.

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.

Antonio Orvieto, Soham De, Caglar Gulcehre, Razvan Pascanu, and Samuel L Smith. Universality of linear recurrences followed by non-linear projections: Finite-width guarantees and benefits of complex eigenvalues. In *Forty-first International Conference on Machine Learning*, 2024.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc Quan Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. The LAMBADA dataset: Word prediction requiring a broad discourse context. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1525–1534, 2016. doi: 10.18653/v1/P16-1144. URL https://aclanthology.org/P16-1144/.

Bo Peng, Ruichong Zhang, Daniel Goldstein, Eric Alcaide, Xingjian Du, Haowen Hou, Jiaju Lin, Jiaxing Liu, Janna Lu, William Merrill, Guangyu Song, Kaifeng Tan, Saiteja Utpala, Nathan Wilce, Johan S. Wind, Tianyi Wu, Daniel Wuttke, and Christian Zhou-Zheng. RWKV-7 "goose" with expressive dynamic state evolution. In *Second Conference on Language Modeling*, 2025.

Korbinian Pöppel, Maximilian Beck, and Sepp Hochreiter. FlashRNN: I/O-Aware Optimization of Traditional RNNs on modern hardware. In *The Thirteenth International Conference on Learning Representations*, 2025.

Zihan Qiu, Zekun Wang, Bo Zheng, Zeyu Huang, Kaiyue Wen, Songlin Yang, Rui Men, Le Yu, Fei Huang, Suozhi Huang, Dayiheng Liu, Jingren Zhou, and Junyang Lin. Gated Attention for Large Language Models: Non-linearity, Sparsity, and Attention-Sink-Free. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=1b7whO4SfY.

Zihan Qiu, Zekun Wang, Xiao Li, Yanpeng Li, Yang Xu, Yixuan Wang, Huaqing Zhang, Rui Men, Bochao Mao, Chengruidong Zhang, Fan Zhou, Hao Luo, Haofeng Huang, Haoran Lian, Haoyan Huang, Hongqing Chen, Jianwei Zhang, Jing Xu, Junjie Wang, Langshi Chen, Liangyu Wang, Linlang Jiang, Man Yuan, Minmin Sun, Peng Jin, Siqi Zhang, Siyu Wang, Xingzhang Ren, Yakai Wang, Yi Zhang, Yiming Dong, Yizhong Cao, Yubo Ma, Yunfei Mao, Bo Zheng, and Dayiheng Liu. On the Design of Qwen3.8-Next Architecture: Evaluation, Efficiency, and Training Stability, 2026. URL https://arxiv.org/abs/2608.30320.

Qwen Team. Qwen3.5-Omni technical report. *arXiv preprint arXiv:2604.15804*, 2026.

Michael O. Rabin. Probabilistic automata. *Information and Control*, 6(3):230–245, 1963. doi: 10.1016/S0019-9958(63)90290-0.

Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for squad. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 784–789, 2018.

Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S. Gordon. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *AAAI Spring Symposium on Logical Formalizations of Commonsense Reasoning*, 2011. URL https://cdn.aaai.org/ocs/2418/2418-10878-1-PB.pdf.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. WinoGrande: An adversarial Winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021. doi: 10.1145/3474381. URL https://doi.org/10.1145/3474381.

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.

Peter Scherk. On the decomposition of orthogonalities into symmetries. *Proceedings of the American Mathematical Society*, 1(4):481–491, 1950.

Imanol Schlag, Kazuki Irie, and Jürgen Schmidhuber. Linear transformers are secretly fast weight programmers. In *International conference on machine learning*, pp. 9355–9366. PMLR, 2021.

Robert Schreiber and Beresford Parlett. Block reflectors: Theory and computation. *SIAM Journal on Numerical Analysis*, 25(1):189–205, 1988.

Mehran Shakerinava, Behnoush Khavari, Siamak Ravanbakhsh, and Sarath Chandar. The expressive limits of diagonal SSMs for state-tracking. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=5bg5Ru5OML.

Julien Siems, Timur Carstensen, Arber Zela, Frank Hutter, Massimiliano Pontil, and Riccardo Grazzi. DeltaProduct: Improving State-Tracking in Linear RNNs via Householder Products. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. arXiv:2502.10297.

Julien Siems, Riccardo Grazzi, Korbinian Pöppel, Kirill Kalinin, Hitesh Ballani, and Babak Rahmani. Learning State-Tracking from Code Using Linear RNNs. *arXiv preprint arXiv:2602.14814*, 2026.

Dan Su, Kezhi Kong, Ying Lin, Joseph Jennings, Brandon Norick, Markus Kliegl, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro. Nemotron-CC: Transforming Common Crawl into a refined long-horizon pretraining dataset. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2459–2475, 2025.

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

Xiaobai Sun and Christian Bischof. A basis-kernel representation of orthogonal matrices. *SIAM journal on matrix analysis and applications*, 16(4):1184–1196, 1995.

Yutao Sun, Li Dong, Shaohan Huang, Shuming Ma, Yuqing Xia, Jilong Xue, Jianyong Wang, and Furu Wei. Retentive network: A successor to transformer for large language models. *arXiv preprint arXiv:2307.08621*, 2023.

Aleksandar Terzic, Michael Hersche, Giacomo Camposampiero, Thomas Hofmann, Abu Sebastian, and Abbas Rahimi. On the expressiveness and length generalization of selective state space models on regular languages. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 20876–20884, 2025a.

Aleksandar Terzic, Nicolas Menet, Michael Hersche, Thomas Hofmann, and Abbas Rahimi. Structured sparse transition matrices to enable state tracking in state-space models. *Advances in Neural Information Processing Systems*, 38:83072–83111, 2025b.

Philippe Tillet, Hsiang-Tsung Kung, and David Cox. Triton: an intermediate language and compiler for tiled neural network computations. In *Proceedings of the 3rd ACM SIGPLAN International Workshop on Machine Learning and Programming Languages*, pp. 10–19, 2019.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open Foundation and Fine-Tuned Chat Models, July 2023. URL http://arxiv.org/abs/2307.09288. arXiv:2307.09288 [cs.CL].

Upstage Solar Team. Solar open 2 technical report. Technical report, Upstage, 2026. arXiv:2607.20062.

Mathurin Videau, Badr Youbi-Idrissi, David Lopez-Paz, and Kartik Ahuja. Skaling: Chinchilla's Exponents Meet Kaplan's Coupling, August 2026. URL http://arxiv.org/abs/2608.07222. arXiv:2608.07222 [cs.CL].

Lei Wang, Yu Cheng, Yining Shi, Zhiwen Mo, Zhengju Tang, Wenhao Xie, Tong Wu, Lingxiao Ma, Yuqing Xia, Jilong Xue, Fan Yang, and Zhi Yang. Tilelang: Bridge programmability and performance in modern neural kernels. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=Jb1WkNSfUB.

Songlin Yang and Yu Zhang. FLA: A Triton-based library for hardware-efficient implementations of linear attention mechanism, January 2024. URL https://github.com/fla-org/flash-linear-attention.

Songlin Yang, Bailin Wang, Yikang Shen, Rameswar Panda, and Yoon Kim. Gated linear attention transformers with hardware-efficient training. In *Forty-first International Conference on Machine Learning*, 2024a.

Songlin Yang, Bailin Wang, Yu Zhang, Yikang Shen, and Yoon Kim. Parallelizing linear transformers with the delta rule over sequence length. *Advances in neural information processing systems*, 37:115491–115522, 2024b.

Songlin Yang, Jan Kautz, and Ali Hatamizadeh. Gated delta networks: Improving Mamba2 with Delta Rule. In *International Conference on Learning Representations (ICLR)*, 2025.

Z.ai. GLM-5.3-Flash: Frontier intelligence, flash cost. https://z.ai/blog/glm-5.3-flash, August 2026. Accessed: 2026-09-15.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a Machine Really Finish Your Sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4791–4800, 2019. doi: 10.18653/v1/P19-1472. URL https://aclanthology.org/P19-1472/.

Tiantian Zhang. Semidirect fourier delta attention: Phase-controlled delta memory with constructive chunk-WY kernels. *arXiv preprint arXiv:2607.11897*, 2026.

SUPPLEMENTARY MATERIAL

The supplementary material is structured as follows:

- Appendix A characterizes orthogonal DPLR matrices, analyzes the spectrum of CKDA, bounds persistent rotations, and studies products of CKDA transitions.
- Appendix B defines the state-tracking tasks, recurrent model, precision and states separability assumptions, and basic group notation.
- Appendix C proves the single-layer finite-group results: the planar and cube realizations, the three-dimensional obstruction and four-dimensional tracker for A_5, and the S_5 obstruction. It ends with the finite-replacement theorem under states separability and its compact-group ingredients.
- Appendix D explains the clock–buffer–accumulator mechanism, proves the three-layer automata result, and gives the factor counts and comparison-table details.
- Appendix E derives the sign transformation that enables reuse of existing KDA kernels and reports the fused implementation's throughput.
- Appendix F discusses how the SiLU key activation affects learning group-word problems.
- Appendix G gives experimental details for state tracking, periodic waveform continuation, and language modeling.
- Appendix H describes the scaling-law methodology and results.
- Appendix I presents additional state-tracking baselines and language-modeling validation curves.

Our code is based on Flash-Linear-Attention (Yang & Zhang, 2024) and available here `https://anonymous.4open.science/r/ComplexKDA-BE14/README.md`.

Notation. Matrices and vectors are denoted by bold uppercase and lowercase letters, respectively. We write $\mathbb{Q}$, $\mathbb{R}$, and $\mathbb{C}$ for the rational, real, and complex numbers. The matrix $\boldsymbol{I}_d$ is the $d\times d$ identity, with the subscript omitted when its size is clear, and $\boldsymbol{e}_i$ is the ith standard coordinate vector. For a scalar z, $|z|$ denotes its absolute value or complex modulus; for a finite set X, $|X|$ denotes its cardinality. We use $\overline{z}$ or z^* for complex conjugation, $\boldsymbol{A}^\top$ for transpose, and $\boldsymbol{A}^*$ for conjugate transpose.

For a vector $\boldsymbol{v}$, $\mathrm{Diag}(\boldsymbol{v})$ is the diagonal matrix with entries $\boldsymbol{v}$, and $\odot$ denotes the element-wise (Hadamard) product. The direct sum $\boldsymbol{A}\oplus\boldsymbol{B}$ places $\boldsymbol{A}$ and $\boldsymbol{B}$ on the diagonal of a block-diagonal matrix. We write $\ker\boldsymbol{A}$, $\mathrm{im}\,\boldsymbol{A}$, and $\mathrm{rank}\,\boldsymbol{A}$ for the kernel, image, and rank of $\boldsymbol{A}$. The notation $\mathrm{span}\{\boldsymbol{v}_1,\dots,\boldsymbol{v}_k\}$ denotes the linear span of the indicated vectors, and $V^\perp$ denotes the orthogonal complement of a subspace V. For $\boldsymbol{v}\in\mathbb{R}^n$, we similarly write $\boldsymbol{v}^\perp := \{\boldsymbol{x}\in\mathbb{R}^n : \boldsymbol{v}^\top\boldsymbol{x}=0\}$; when $\boldsymbol{v}\neq 0$, this is a hyperplane through the origin. For a subspace V, $\boldsymbol{A}|_V$ denotes the restriction of $\boldsymbol{A}$ to vectors in V. If V is invariant under $\boldsymbol{A}$, meaning $\boldsymbol{A}V\subseteq V$, this restriction is a linear map from V to itself.

The vector norm $\|\boldsymbol{v}\| = \|\boldsymbol{v}\|_2$ is Euclidean, while $\|\boldsymbol{A}\| = \|\boldsymbol{A}\|_2$ is the induced operator norm. The Frobenius norm is $\|\boldsymbol{A}\|_F = (\sum_{i,j}|A_{ij}|^2)^{1/2}$. For a real symmetric matrix, $\boldsymbol{A}\succeq 0$ and $\boldsymbol{A}\succ 0$ mean positive semidefinite and positive definite, respectively. We write $\sigma(\boldsymbol{A})$ for the spectrum (the set of eigenvalues), $\rho(\boldsymbol{A})$ for the spectral radius, and $\mathrm{tr}(\boldsymbol{A})$ and $\det(\boldsymbol{A})$ for the trace and determinant.

For an alphabet Σ, Σ^* is the set of all finite words, including the empty word. Transitions act on column states, so the product $\boldsymbol{A}_t\cdots\boldsymbol{A}_1$ applies $\boldsymbol{A}_1$ first.

A CHARACTERIZATION OF A CKDA TRANSITION

A.1 ORTHOGONAL DPLR MATRICES REDUCE TO SIGNED HOUSEHOLDERS

Rank-one perturbations of isometries and low-rank representations of orthogonal transformations have a classical literature; see, e.g., Scherk (1950); Nakamura (1986); Schreiber & Parlett (1988);

Table 3: **Spectral structure of complex-capable delta recurrences**

	CKDA	MDN (quadrant-constrained)	SFDA (default complex)
Transition	$(\boldsymbol{I} - \beta \boldsymbol{k}\boldsymbol{k}^\top) D(\boldsymbol{\alpha})$	$\begin{bmatrix} \alpha(\boldsymbol{I} - \beta\eta \boldsymbol{k}\boldsymbol{k}^\top) & -\beta\mu \boldsymbol{I} \\ \alpha\eta \boldsymbol{k}\boldsymbol{k}^\top & \mu \boldsymbol{I} \end{bmatrix}$	$(\boldsymbol{I} - \beta \boldsymbol{k}\boldsymbol{k}^*)\boldsymbol{\Lambda}$
Recurrence order	First order	First order in $(\boldsymbol{S}, \boldsymbol{M})$; generically second order in $\boldsymbol{S}$	First order over $\mathbb{C}$
State per value column	d real	$2d$ real	m complex $\simeq 2m$ real
Ranges compared	$\beta \in [0, 2]$ $\boldsymbol{\alpha} \in [-1, 1]^d$	$\alpha \in (0, 1), \mu \in [e^{-1}, 1)$ $0 \le \beta \le 1 - \alpha$ $\eta \in (0, 2)$	$\beta \in [0, 1]$ $\boldsymbol{\alpha} \in [0, 1]^m$ $\boldsymbol{\theta} \in [-\theta_{\max}, \theta_{\max}]^m$
Non-real conjugate pairs	$\le \mathbf{1}$ **(Thm. 8)**	≤ 1	Up to m after realification
Origin of complex modes	Signed gate–delta interaction	Memory–momentum coupling	Explicit phases plus delta correction
Non-real $\lvert\lambda\rvert = 1$ modes†	✓ Any angle in a two-dimensional block	× None under the stated constraints	✓ Undamped phase modes
Unit-circle support†	**Full circle**	Only $\{+1\}$	$\{e^{\pm i\theta} : \lvert\theta\rvert \le \theta_{\max}\}$; full circle if $\theta_{\max} \ge \pi$
Whole-state isometries†	Orthogonal DPR1; need not commute	None in the interior; only $\boldsymbol{I}$ in the closure	Diagonal unit phases ($\beta = 0$, $\boldsymbol{\alpha} = \mathbf{1}$); commute
Non-expansive by construction	✓ $\lVert\boldsymbol{A}\rVert_2 \le 1$	× Not guaranteed in the augmented Euclidean norm	✓ $\lVert\boldsymbol{A}\rVert_2 \le 1$

Time subscripts are suppressed; keys have unit norm, $d \ge 2$, $m \ge 1$, and $\theta_{\max} > 0$. $D(\boldsymbol{\alpha}) = \mathrm{Diag}(\boldsymbol{\alpha})$ and $\boldsymbol{\Lambda} = \mathrm{Diag}(\alpha_j e^{i\theta_j})$. Dimensions and pair counts concern the transition acting on one value column, not the vectorized whole memory. † Unit-circle support and isometry entries include the boundaries/closures of the stated parameter families; sigmoid/tanh parameterizations need not attain these endpoints at finite logits. Isometry means $\boldsymbol{A}^*\boldsymbol{A} = \boldsymbol{I}$ on the entire state space. MDN uses the constraint family in its Sec. 3.2; its experiments also report other momentum floors. SFDA refers specifically to its complex Eq. (13): its exact realification has rotation–decay blocks plus a correction of real rank at most two. Its isometry statement does not apply to an extension with $\beta = 2$ or to enlarged reflection/control families, and does not assert that general SFDA transitions commute. All spectral statements are single-step.

Sun & Bischof (1995). The following result gives a more specific rigidity statement for diagonal-plus-rank-one matrices: orthogonality forces the diagonal factor to reduce, up to the rank-one correction, to a sign matrix.

Theorem 6 (DPR1 and CKDA; restatement of Theorem 1)**.** *Every orthogonal DPR1 matrix* $\boldsymbol{A} = \boldsymbol{D} + \boldsymbol{u}\boldsymbol{v}^\top \in \mathbb{R}^{n\times n}$ *with diagonal* $\boldsymbol{D}$ *can be written in the form* $\boldsymbol{A} = (\boldsymbol{I} - 2\boldsymbol{k}\boldsymbol{k}^\top)\boldsymbol{S}$, $\lVert\boldsymbol{k}\rVert_2 = 1$, $\boldsymbol{S} = \mathrm{Diag}(s_i)$, $s_i \in \{-1, +1\}$. *We call this form a* signed-Householder matrix*, i.e. a CKDA with* $\alpha_i \in \{+1, -1\}$, $\beta = 2$.

Proof. Write $\boldsymbol{D} = \mathrm{Diag}(d_1, \ldots, d_n)$. If $\boldsymbol{A}$ is diagonal, orthogonality makes it a diagonal sign matrix. Take $\boldsymbol{k} = \boldsymbol{e}_1$ and let $\boldsymbol{H} = \boldsymbol{I} - 2\boldsymbol{e}_1\boldsymbol{e}_1^\top$, which flips the first coordinate. Then $\boldsymbol{S} := \boldsymbol{H}\boldsymbol{A}$ is also a diagonal sign matrix, and $\boldsymbol{H}^2 = \boldsymbol{I}$ gives $\boldsymbol{A} = \boldsymbol{H}\boldsymbol{S}$. Its eigenvalues are already ± 1.

If instead $\boldsymbol{A}$ is not diagonal, so $\boldsymbol{u}, \boldsymbol{v} \neq 0$, we seek column sign flips $\boldsymbol{S}$ that turn $\boldsymbol{A}\boldsymbol{S}$ into a reflection. Since reflections are symmetric, we first ask whether these flips can make the rank-one term symmetric. Orthogonality gives $\boldsymbol{A}^\top\boldsymbol{A} = \boldsymbol{A}\boldsymbol{A}^\top = \boldsymbol{I}$, so the ith column and row both have squared norm one:

$$1 = \lVert d_i\boldsymbol{e}_i + v_i\boldsymbol{u}\rVert^2 = d_i^2 + 2d_iu_iv_i + \lVert\boldsymbol{u}\rVert^2 v_i^2,$$
$$1 = \lVert d_i\boldsymbol{e}_i + u_i\boldsymbol{v}\rVert^2 = d_i^2 + 2d_iu_iv_i + \lVert\boldsymbol{v}\rVert^2 u_i^2.$$

Subtracting the two equations yields $\lVert\boldsymbol{u}\rVert^2 v_i^2 = \lVert\boldsymbol{v}\rVert^2 u_i^2$, hence $\lvert v_i\rvert = c\lvert u_i\rvert$ with $c := \lVert\boldsymbol{v}\rVert/\lVert\boldsymbol{u}\rVert > 0$. Thus $u_i = 0$ exactly when $v_i = 0$; on these coordinates, the preceding equations also give $d_i^2 = 1$.

We can therefore choose signs so that $\boldsymbol{S}\boldsymbol{v} = -c\boldsymbol{u}$, making the rank-one term a negative multiple of $\boldsymbol{u}\boldsymbol{u}^\top$, as in a Householder reflection. Explicitly, take $\boldsymbol{S} = \mathrm{Diag}(s_1, \ldots, s_n)$ with

$$s_i = \begin{cases} -cu_i/v_i, & u_i \neq 0, \\ d_i, & u_i = 0. \end{cases}$$

Each s_i is ± 1. Consequently, $\boldsymbol{Q} := \boldsymbol{A}\boldsymbol{S} = \boldsymbol{D}\boldsymbol{S} + \boldsymbol{u}(\boldsymbol{S}\boldsymbol{v})^\top = \boldsymbol{D}\boldsymbol{S} - c\boldsymbol{u}\boldsymbol{u}^\top$ is symmetric and orthogonal, hence $\boldsymbol{Q}^2 = \boldsymbol{I}$.

A reflection moves every vector along its normal direction. To find that direction, consider the displacement $\boldsymbol{r} = \boldsymbol{x} - \boldsymbol{Q}\boldsymbol{x}$ of any vector $\boldsymbol{x}$. Since $\boldsymbol{Q}\boldsymbol{r} = \boldsymbol{Q}\boldsymbol{x} - \boldsymbol{Q}^2\boldsymbol{x} = \boldsymbol{Q}\boldsymbol{x} - \boldsymbol{x} = -\boldsymbol{r}$, every displacement is reversed by $\boldsymbol{Q}$. Substituting the formula for $\boldsymbol{Q}$ gives $\boldsymbol{D}\boldsymbol{S}\boldsymbol{r} - c\boldsymbol{u}(\boldsymbol{u}^\top\boldsymbol{r}) = -\boldsymbol{r}$, or coordinatewise, $(1 + d_i s_i) r_i = c u_i (\boldsymbol{u}^\top \boldsymbol{r})$.

We can solve these equations coordinatewise. Indeed, $|Q_{ii}| \leq 1$ because $\boldsymbol{Q}$ is orthogonal, so for $u_i \neq 0$ we have $1 + d_i s_i = 1 + Q_{ii} + c u_i^2 > 0$. For $u_i = 0$, the same denominator equals $1 + d_i^2 = 2$. Thus $\boldsymbol{r} = c(\boldsymbol{u}^\top \boldsymbol{r})\boldsymbol{w}$, where $w_i := u_i/(1 + d_i s_i)$. Every displacement therefore lies along the same nonzero vector $\boldsymbol{w}$.

Since $\boldsymbol{A}$ is not diagonal, $\boldsymbol{Q} \neq \boldsymbol{I}$; otherwise $\boldsymbol{A} = \boldsymbol{S}$. Hence some displacement is a nonzero multiple of $\boldsymbol{w}$, and $\boldsymbol{Q}\boldsymbol{r} = -\boldsymbol{r}$ implies $\boldsymbol{Q}\boldsymbol{w} = -\boldsymbol{w}$. For arbitrary $\boldsymbol{x}$, write $\boldsymbol{x} - \boldsymbol{Q}\boldsymbol{x} = t\boldsymbol{w}$. Symmetry determines the coefficient:

$$t\|\boldsymbol{w}\|^2 = \boldsymbol{w}^\top(\boldsymbol{x} - \boldsymbol{Q}\boldsymbol{x}) = \boldsymbol{w}^\top\boldsymbol{x} - (\boldsymbol{Q}\boldsymbol{w})^\top\boldsymbol{x} = 2\boldsymbol{w}^\top\boldsymbol{x}.$$

Therefore $\boldsymbol{Q}\boldsymbol{x} = \boldsymbol{x} - 2\boldsymbol{w}(\boldsymbol{w}^\top\boldsymbol{x})/\|\boldsymbol{w}\|^2$, so $\boldsymbol{Q} = \boldsymbol{I} - 2\boldsymbol{k}\boldsymbol{k}^\top$ with $\boldsymbol{k} := \boldsymbol{w}/\|\boldsymbol{w}\|$. Since $\boldsymbol{S}^2 = \boldsymbol{I}$, we obtain $\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{S} = (\boldsymbol{I} - 2\boldsymbol{k}\boldsymbol{k}^\top)\boldsymbol{S}$.

The factorization also shows that $\boldsymbol{A}$ has at most one non-real conjugate eigenvalue pair, counted with algebraic multiplicity. Indeed, $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{S}\boldsymbol{x} - 2\boldsymbol{k}(\boldsymbol{S}\boldsymbol{k})^\top\boldsymbol{x}$. The correction singles out $\boldsymbol{k}$ and $\boldsymbol{S}\boldsymbol{k}$, so consider $\mathcal{U} := \mathrm{span}\{\boldsymbol{k}, \boldsymbol{S}\boldsymbol{k}\}$. The matrix $\boldsymbol{S}$ exchanges these two vectors and is orthogonal, so it preserves both $\mathcal{U}$ and $\mathcal{U}^\perp$. The formula for $\boldsymbol{A}\boldsymbol{x}$ shows that $\boldsymbol{A}$ also preserves both spaces and agrees with $\boldsymbol{S}$ on $\mathcal{U}^\perp$. Its eigenvalues there are therefore ± 1. All non-real eigenvalues lie in the restriction to $\mathcal{U}$, whose dimension is at most two, allowing at most one conjugate pair. □

Remark. The characterization above can also be derived from (Ionascu, 2001, Proposition 3.1) which characterizes normal rank-one perturbations of normal operators, specialized to the finite-dimensional real orthogonal case; here we state it explicitly in the signed-Householder form relevant to CKDA.

A.2 The Sign-Magnitude Decomposition of CKDA

Proposition 7 (Expanded Proposition 2)**.** *Let $\|\boldsymbol{k}\| = 1$, $\beta \in [0, 2]$, and $\alpha_i \in [-1, 1]$. Define*

$$\boldsymbol{A} = (\boldsymbol{I} - \beta\boldsymbol{k}\boldsymbol{k}^\top)\,\mathrm{Diag}(\boldsymbol{\alpha}) = \widetilde{\boldsymbol{A}}\,\mathrm{Diag}(|\boldsymbol{\alpha}|), \qquad \widetilde{\boldsymbol{A}} := (\boldsymbol{I} - \beta\boldsymbol{k}\boldsymbol{k}^\top)\boldsymbol{S}, \qquad \boldsymbol{S} := \mathrm{Diag}(\mathrm{sign}\,\boldsymbol{\alpha}),$$

with $\mathrm{sign}(0) = 1$. Let $E_- := \ker(\boldsymbol{S} + \boldsymbol{I})$ be the space spanned by the coordinate vectors whose signs are flipped by $\boldsymbol{S}$, with dimension m_-, and $E_+ := \ker(\boldsymbol{S} - \boldsymbol{I})$ with dimension m_+, and write

$$\boldsymbol{k} = \boldsymbol{k}_- + \boldsymbol{k}_+, \qquad \boldsymbol{k}_- \in E_-, \quad \boldsymbol{k}_+ \in E_+, \qquad \|\boldsymbol{k}_-\| = \cos\theta, \quad \|\boldsymbol{k}_+\| = \sin\theta,$$

where $\theta \in [0, \pi/2]$ is the angle of $\boldsymbol{k}$ from E_-. When nonzero, let $\boldsymbol{e}_- := \boldsymbol{k}_-/\|\boldsymbol{k}_-\|$, $\boldsymbol{e}_+ := \boldsymbol{k}_+/\|\boldsymbol{k}_+\|$, and let $\boldsymbol{P}$ be an orthogonal matrix beginning with the relevant vectors $\boldsymbol{e}_-, \boldsymbol{e}_+$ and then completed by orthonormal bases of E_- first and then E_+. Then

$$\boxed{\begin{aligned} 0 < \theta < \frac{\pi}{2}: &\qquad \widetilde{\boldsymbol{A}} = \boldsymbol{P}\big[\boldsymbol{B}(\beta, \theta) \oplus (-\boldsymbol{I}_{m_- - 1}) \oplus \boldsymbol{I}_{m_+ - 1}\big]\boldsymbol{P}^\top, \\ \theta = 0: &\qquad \widetilde{\boldsymbol{A}} = \boldsymbol{P}\big[(\beta - 1) \oplus (-\boldsymbol{I}_{m_- - 1}) \oplus \boldsymbol{I}_{m_+}\big]\boldsymbol{P}^\top, \\ \theta = \frac{\pi}{2}: &\qquad \widetilde{\boldsymbol{A}} = \boldsymbol{P}\big[(1 - \beta) \oplus (-\boldsymbol{I}_{m_-}) \oplus \boldsymbol{I}_{m_+ - 1}\big]\boldsymbol{P}^\top. \end{aligned}}$$

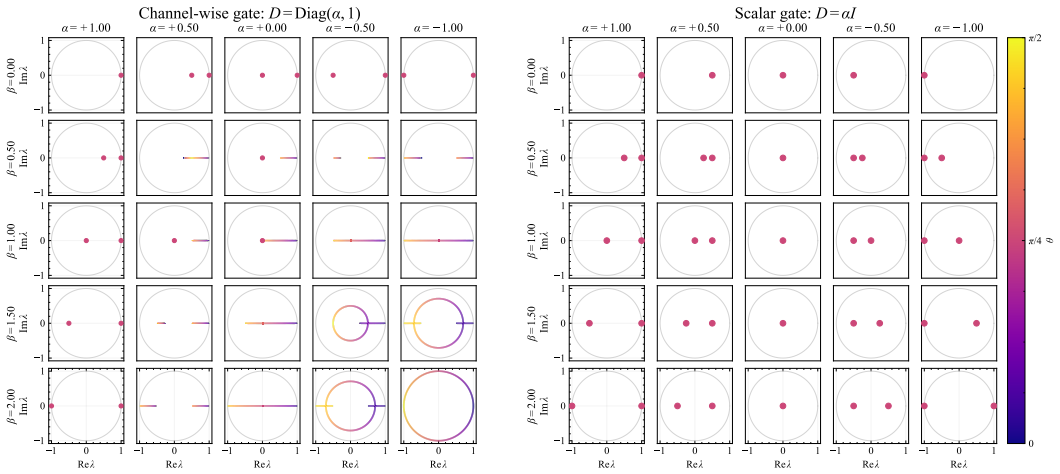

Figure 8: Extension of Figure 2 varying both β and α. The figure demonstrates that both components need to be extended from their standard ranges to obtain complex eigenvalues.

Here $\oplus$ denotes the direct sum of matrix blocks, which places the two operands as the blocks of a block-diagonal matrix. The only nontrivial block is

$$\boldsymbol{B}(\beta,\theta)=\begin{pmatrix}\beta\cos^2\theta-1 & -\beta\sin\theta\cos\theta\\ \beta\sin\theta\cos\theta & 1-\beta\sin^2\theta\end{pmatrix},\qquad \boldsymbol{B}(2,\theta)=\begin{pmatrix}\cos 2\theta & -\sin 2\theta\\ \sin 2\theta & \cos 2\theta\end{pmatrix}.$$

More specifically, $\widetilde{\boldsymbol{A}}$ has one non-real conjugate pair exactly when the eigenvalues of $\boldsymbol{B}(\beta,\theta)$ are non-real, i.e. when $\beta^2\cos^2(2\theta)<4(\beta-1)$, which only holds when $\beta>1$. When $\beta=2$, $\boldsymbol{B}(\beta,\theta)$ becomes a pure rotation by 2θ.

Proof. For $x\in E_-\cap\boldsymbol{k}_-^{\perp}$ and $y\in E_+\cap\boldsymbol{k}_+^{\perp}$,

$$\widetilde{\boldsymbol{A}}x=-x,\qquad \widetilde{\boldsymbol{A}}y=y,$$

so only the span of the nonzero components of $\boldsymbol{k}$ remains. When both are nonzero,

$$\boldsymbol{k}=\cos\theta\,\boldsymbol{e}_-+\sin\theta\,\boldsymbol{e}_+,\qquad \boldsymbol{S}\boldsymbol{e}_-=-\boldsymbol{e}_-,\qquad \boldsymbol{S}\boldsymbol{e}_+=\boldsymbol{e}_+,$$

and evaluating $\widetilde{\boldsymbol{A}}$ on $\boldsymbol{e}_-,\boldsymbol{e}_+$ gives $\boldsymbol{B}(\beta,\theta)$. The endpoint cases follow by setting $\boldsymbol{k}=\boldsymbol{k}_-$ or $\boldsymbol{k}=\boldsymbol{k}_+$, and the form of $\boldsymbol{B}(2,\theta)$ follows from the double-angle identities. □

A.3 The Window for Complex Eigenvalues

We now consider the eigenvalues of the full CKDA transition.

Theorem 8 (Complex Eigenvalues in Diagonal Householder Product)**.** *Let $\boldsymbol{k}\in\mathbb{R}^n$ satisfy $\|\boldsymbol{k}\|_2=1$, and let $\beta\in\mathbb{R}$ and $\boldsymbol{\alpha}\in\mathbb{R}^n$. The matrix $\boldsymbol{A}=(\boldsymbol{I}-\beta\boldsymbol{k}\boldsymbol{k}^\top)\,\mathrm{Diag}(\boldsymbol{\alpha})$ has at most one non-real conjugate eigenvalue pair, counted with algebraic multiplicity. Such a pair can occur only if $\beta>1$ and $\alpha_i<0$ for at least one i.*

This means it is not sufficient to extend $\beta\in[0,2]$ alone within KDA as in Grazzi et al. (2025), nor is it sufficient to extend only the range of α_i while keeping $\beta\le 1$. This is similar to the requirement shown in (Siems et al., 2025, Prop. 1.3), where both βs in a product of two generalized Householders need to be greater than 1 for the product to obtain complex eigenvalues.

Proof. Let $\boldsymbol{H}_{\boldsymbol{k}}^{\beta} := \boldsymbol{I} - \beta \boldsymbol{k}\boldsymbol{k}^{\top}$ and $\boldsymbol{D} := \mathrm{Diag}(\boldsymbol{\alpha})$. The matrix $\boldsymbol{H}_{\boldsymbol{k}}^{\beta}$ has eigenvalues $(1-\beta, 1, \ldots, 1)$, with $\boldsymbol{k}$ as an eigenvector for $1-\beta$ and $\boldsymbol{k}^{\perp}$ as the eigenspace for 1. Thus we can choose an orthogonal matrix $\boldsymbol{Q}$ with $\boldsymbol{k}$ as its first column such that

$$\boldsymbol{Q}^{\top} \boldsymbol{H}_{\boldsymbol{k}}^{\beta} \boldsymbol{Q} = \boldsymbol{\Lambda} := \mathrm{Diag}(1-\beta, 1, \ldots, 1).$$

Defining $\boldsymbol{B} := \boldsymbol{Q}^{\top} \boldsymbol{D} \boldsymbol{Q}$, which is symmetric, we obtain

$$\boldsymbol{Q}^{\top} \boldsymbol{A} \boldsymbol{Q} = \boldsymbol{\Lambda} \boldsymbol{B}. \tag{4}$$

First suppose $\beta < 1$. Then $\boldsymbol{\Lambda} \succ 0$, and

$$\boldsymbol{\Lambda}^{-1/2} (\boldsymbol{\Lambda} \boldsymbol{B}) \boldsymbol{\Lambda}^{1/2} = \boldsymbol{\Lambda}^{1/2} \boldsymbol{B} \boldsymbol{\Lambda}^{1/2},$$

which is symmetric. Hence all eigenvalues are real.

For $\beta = 1$, write

$$\boldsymbol{B} = \begin{pmatrix} b_{11} & \boldsymbol{b}^{\top} \\ \boldsymbol{b} & \boldsymbol{B}_{22} \end{pmatrix}.$$

Then

$$\boldsymbol{\Lambda} \boldsymbol{B} = \begin{pmatrix} 0 & 0 \\ \boldsymbol{b} & \boldsymbol{B}_{22} \end{pmatrix}.$$

This matrix has eigenvalues 0 together with the eigenvalues of the symmetric matrix $\boldsymbol{B}_{22}$, so its spectrum is again real. Therefore non-real eigenvalues require $\beta > 1$.

Now suppose $\beta > 1$. Define

$$\boldsymbol{\Sigma} := \mathrm{Diag}(\beta - 1, 1, \ldots, 1), \qquad \boldsymbol{J} := \mathrm{Diag}(-1, 1, \ldots, 1),$$

so that $\boldsymbol{\Lambda} = \boldsymbol{J} \boldsymbol{\Sigma}$. Applying the similarity transformation with $\boldsymbol{\Sigma}^{1/2}$ gives

$$\boldsymbol{\Sigma}^{-1/2} (\boldsymbol{\Lambda} \boldsymbol{B}) \boldsymbol{\Sigma}^{1/2} = \boldsymbol{J} \underbrace{\boldsymbol{\Sigma}^{1/2} \boldsymbol{B} \boldsymbol{\Sigma}^{1/2}}_{\boldsymbol{S}} =: \boldsymbol{T}, \tag{5}$$

where $\boldsymbol{S}$ is symmetric. Hence $\boldsymbol{T}^{*} \boldsymbol{J} = \boldsymbol{J} \boldsymbol{T}$.

Let $\boldsymbol{T}\boldsymbol{x} = \lambda \boldsymbol{x}$ with $\lambda \notin \mathbb{R}$. Then

$$\boldsymbol{x}^{*} \boldsymbol{J} \boldsymbol{T} \boldsymbol{x} = \lambda \boldsymbol{x}^{*} \boldsymbol{J} \boldsymbol{x} = \boldsymbol{x}^{*} \boldsymbol{S} \boldsymbol{x} \in \mathbb{R}. \tag{6}$$

Since $\boldsymbol{x}^{*} \boldsymbol{J} \boldsymbol{x} \in \mathbb{R}$ and $\lambda \notin \mathbb{R}$, this implies $\boldsymbol{x}^{*} \boldsymbol{J} \boldsymbol{x} = 0$. Similarly, for two eigenvectors $\boldsymbol{T}\boldsymbol{x} = \lambda \boldsymbol{x}$ and $\boldsymbol{T}\boldsymbol{y} = \mu \boldsymbol{y}$,

$$(\mu - \overline{\lambda}) \boldsymbol{x}^{*} \boldsymbol{J} \boldsymbol{y} = 0. \tag{7}$$

Thus eigenvectors corresponding to eigenvalues in the upper half-plane are mutually $\boldsymbol{J}$-orthogonal and $\boldsymbol{J}$-isotropic.

The same statement holds for the corresponding generalized eigenspace. Let $\boldsymbol{X}$ span the generalized eigenspace associated with all eigenvalues in the upper half-plane, and write $\boldsymbol{T}\boldsymbol{X} = \boldsymbol{X}\boldsymbol{R}$. Setting $\boldsymbol{G} := \boldsymbol{X}^{*} \boldsymbol{J} \boldsymbol{X}$, the identity $\boldsymbol{T}^{*} \boldsymbol{J} = \boldsymbol{J} \boldsymbol{T}$ gives

$$\boldsymbol{R}^{*} \boldsymbol{G} = \boldsymbol{G} \boldsymbol{R}.$$

The spectrum of $\boldsymbol{R}$ lies in the upper half-plane, while that of $\boldsymbol{R}^{*}$ lies in the lower half-plane. Hence the corresponding Sylvester equation has the unique solution $\boldsymbol{G} = 0$, so the generalized eigenspace is totally $\boldsymbol{J}$-isotropic.

Since $\boldsymbol{J} = \mathrm{Diag}(-1, 1, \ldots, 1)$ has only one negative direction, every totally $\boldsymbol{J}$-isotropic subspace has dimension at most one. Indeed, if a vector in such a subspace has first coordinate zero, then

$$\boldsymbol{x}^{*} \boldsymbol{J} \boldsymbol{x} = \sum_{i=2}^{n} |x_i|^2 = 0,$$

and hence $\boldsymbol{x} = 0$. Projection onto the first coordinate is therefore injective. Thus there is at most one eigenvalue in the upper half-plane, counting algebraic multiplicity, and since $\boldsymbol{A}$ is real, at most one corresponding complex-conjugate pair.

Finally, if $\alpha_i \geq 0$ for all i, then $\boldsymbol{D} \succeq 0$. The matrices $\boldsymbol{H}_{\boldsymbol{k}}^{\beta} \boldsymbol{D}$ and

$$\boldsymbol{D}^{1/2} \boldsymbol{H}_{\boldsymbol{k}}^{\beta} \boldsymbol{D}^{1/2}$$

have the same characteristic polynomial, while the latter is symmetric. Hence all eigenvalues of $\boldsymbol{A}$ are real. Therefore non-real eigenvalues require $\alpha_i < 0$ for at least one i. □

A.4 NON-EXPANSIVE DPLR MATRICES

Theorem 9 (Persistent rotations in non-expansive DPLR)**.** *Let $\boldsymbol{A} = \boldsymbol{D} + \boldsymbol{R} \in \mathbb{R}^{n\times n}$, where $\boldsymbol{D} = \boldsymbol{D}^\top$ is diagonal and $\operatorname{rank}(\boldsymbol{R}) \le r$, and suppose $\|\boldsymbol{A}\|_2 \le 1$. Then $\boldsymbol{A}$ has at most r non-real conjugate pairs of eigenvalues on the unit circle, counted with algebraic multiplicity. In particular, a non-expansive rank-one DPLR transition has at most one such pair.*

Proof. Since $\boldsymbol{D}$ is symmetric, the difference between $\boldsymbol{A}$ and its transpose comes entirely from the low-rank term:

$$\operatorname{rank}(\boldsymbol{A} - \boldsymbol{A}^\top) = \operatorname{rank}(\boldsymbol{R} - \boldsymbol{R}^\top) \le \operatorname{rank}(\boldsymbol{R}) + \operatorname{rank}(\boldsymbol{R}^\top) \le 2r.$$

We will show that each non-real unit-circle pair contributes two independent vectors to this range.

Let $\boldsymbol{A}\boldsymbol{x} = \lambda\boldsymbol{x}$ with $\boldsymbol{x} \in \mathbb{C}^n\backslash\{0\}$ and $|\lambda| = 1$, and write $\boldsymbol{x}^*$ for conjugate transpose. This eigenvector loses no norm under $\boldsymbol{A}$; we first show that $\boldsymbol{A}^\top$ reverses its action. Since $\|\boldsymbol{A}^\top\|_2 = \|\boldsymbol{A}\|_2 \le 1$,

$$\begin{aligned}\|\boldsymbol{A}^\top\boldsymbol{x} - \bar\lambda\boldsymbol{x}\|^2 &= \|\boldsymbol{A}^\top\boldsymbol{x}\|^2 + \|\boldsymbol{x}\|^2 - 2\operatorname{Re}(\bar\lambda\boldsymbol{x}^*\boldsymbol{A}\boldsymbol{x}) \\ &= \|\boldsymbol{A}^\top\boldsymbol{x}\|^2 + \|\boldsymbol{x}\|^2 - 2\operatorname{Re}(\bar\lambda\lambda\|\boldsymbol{x}\|^2) \\ &= \|\boldsymbol{A}^\top\boldsymbol{x}\|^2 - \|\boldsymbol{x}\|^2 \le 0.\end{aligned}$$

A squared norm cannot be negative, so $\boldsymbol{A}^\top\boldsymbol{x} = \bar\lambda\boldsymbol{x}$. Consequently, $(\boldsymbol{A} - \boldsymbol{A}^\top)\boldsymbol{x} = (\lambda - \bar\lambda)\boldsymbol{x}$. If λ is non-real, the coefficient is nonzero, and $\boldsymbol{x} = (\lambda - \bar\lambda)^{-1}(\boldsymbol{A} - \boldsymbol{A}^\top)\boldsymbol{x}$ belongs to the range of $\boldsymbol{A} - \boldsymbol{A}^\top$.

To count these vectors with multiplicity, observe that $\boldsymbol{A}$ also preserves $\boldsymbol{x}^\perp$: whenever $\boldsymbol{x}^*\boldsymbol{y} = 0$, we have $\boldsymbol{x}^*\boldsymbol{A}\boldsymbol{y} = (\boldsymbol{A}^\top\boldsymbol{x})^*\boldsymbol{y} = \lambda\boldsymbol{x}^*\boldsymbol{y} = 0$. Thus $\operatorname{span}\{\boldsymbol{x}\}$ and its orthogonal complement are both invariant, and the restriction to the complement remains a contraction. Splitting off this one-dimensional block and repeating produces independent eigenvectors for every unit-circle eigenvalue, counted with algebraic multiplicity.

If there are c non-real conjugate pairs on the unit circle, we therefore obtain $2c$ independent vectors in the range of $\boldsymbol{A} - \boldsymbol{A}^\top$. A real matrix has the same rank over $\mathbb{R}$ and $\mathbb{C}$, so $2c \le 2r$, giving $c \le r$. □

A.5 EXPRESSIVITY OF PRODUCTS OF CKDA TRANSITIONS

The diagonal gate reduces the number of factors needed to represent a non-expansive matrix compared with the $3n$ generalized-Householder construction of Grazzi et al. (2025, Prop. 1, item 2).

Proposition 10 (Expressivity of CKDA products)**.** *Let $n \ge 1$ and $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ satisfy $\|\boldsymbol{A}\|_2 \le 1$. Then $\boldsymbol{A}$ is a product of at most $\max\{1, 2n-2\}$ CKDA transitions, each with a unit key, $\beta \in [0, 2]$, and diagonal gate entries in $[-1, 1]$. If $\boldsymbol{A}$ is orthogonal, at most $\max\{1, n-1\}$ factors suffice, and this orthogonal bound is sharp. In particular, the permutation matrix of an n-cycle cannot be expressed using fewer than $n-1$ CKDA factors.*

Proof. The zero matrix is a single zero-gate transition, and for $n = 1$ any scalar $A \in [-1, 1]$ is a single diagonal gate with $\beta = 0$. The scalar $A = -1$ also shows sharpness of the orthogonal bound in dimension one. Henceforth let $n \ge 2$ and $\boldsymbol{A} \ne 0$.

Orthogonal matrices. By Householder QR decomposition, any orthogonal matrix can be written as $\boldsymbol{Q} = \boldsymbol{H}_1 \cdots \boldsymbol{H}_{n-1}\boldsymbol{S}$, where $\boldsymbol{S}$ is a diagonal sign matrix and each $\boldsymbol{H}_i$ is a reflection or an identity padding factor. Absorbing $\boldsymbol{S}$ into the last factor gives $n-1$ CKDA transitions.

General contractions. Take a singular value decomposition $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$, with $\boldsymbol{\Sigma} = \operatorname{Diag}(\sigma_1, \ldots, \sigma_n)$ and $\sigma_i \in [0, 1]$. Apply Householder QR to both orthogonal factors:

$$\boldsymbol{U} = \boldsymbol{H}_1 \cdots \boldsymbol{H}_{n-1}\boldsymbol{S}_U, \qquad \boldsymbol{V} = \boldsymbol{G}_1 \cdots \boldsymbol{G}_{n-1}\boldsymbol{S}_V.$$

Each $\boldsymbol{H}_i$ and $\boldsymbol{G}_i$ is a reflection or an identity padding factor, and $\boldsymbol{S}_U, \boldsymbol{S}_V$ are diagonal sign matrices. Thus, with $\boldsymbol{D} := \boldsymbol{S}_U\boldsymbol{\Sigma}\boldsymbol{S}_V$, whose entries lie in $[-1, 1]$,

$$\boldsymbol{A} = \boldsymbol{H}_1 \cdots \boldsymbol{H}_{n-2}(\boldsymbol{H}_{n-1}\boldsymbol{D})\boldsymbol{G}_{n-1} \cdots \boldsymbol{G}_1.$$

Every reflection is a CKDA transition with $\beta = 2$ and identity gate, and an identity factor uses $\beta = 0$. Absorbing $\boldsymbol{D}$ into $\boldsymbol{H}_{n-1}$ gives $(n-2)+1+(n-1) = 2n-2$ CKDA factors.

Sharpness for orthogonal matrices. Each CKDA factor is diagonal plus a matrix of rank at most one. Inductively, a product of m such factors is diagonal plus a matrix of rank at most m: multiplying $\boldsymbol{D} + \boldsymbol{R}$, with $\operatorname{rank}(\boldsymbol{R}) \le m$, by $\boldsymbol{E} + \boldsymbol{u}\boldsymbol{v}^\top$ gives the diagonal part $\boldsymbol{E}\boldsymbol{D}$ and correction $\boldsymbol{E}\boldsymbol{R} + \boldsymbol{u}\boldsymbol{v}^\top(\boldsymbol{D} + \boldsymbol{R})$, of rank at most $m+1$.

Now let $\boldsymbol{C}_n$ be the cyclic permutation matrix defined by $\boldsymbol{C}_n\boldsymbol{e}_i = \boldsymbol{e}_{i+1}$ for $i < n$ and $\boldsymbol{C}_n\boldsymbol{e}_n = \boldsymbol{e}_1$. For any diagonal $\boldsymbol{D} = \operatorname{Diag}(d_1, \dots, d_n)$, the equation $(\boldsymbol{C}_n - \boldsymbol{D})\boldsymbol{x} = 0$ reads

$$x_{i-1} = d_i x_i \quad (i = 2, \dots, n), \qquad x_n = d_1 x_1.$$

Starting from x_n, the first $n-1$ equations determine all other coordinates; the last equation can only impose an additional constraint. Thus $\dim\ker(\boldsymbol{C}_n - \boldsymbol{D}) \le 1$, so $\operatorname{rank}(\boldsymbol{C}_n - \boldsymbol{D}) \ge n-1$ for every diagonal $\boldsymbol{D}$. Recall that a product of m CKDA factors differs from some diagonal matrix by a correction of rank at most m. Thus any such factorization of $\boldsymbol{C}_n$ would give a diagonal $\boldsymbol{D}$ satisfying $n-1 \le \operatorname{rank}(\boldsymbol{C}_n - \boldsymbol{D}) \le m$, forcing $m \ge n-1$. Since $\boldsymbol{C}_n$ is orthogonal, the upper bound is attained. □

Remark. This bound is also related to the Hermitian-plus-low-rank characterization of Del Corso et al. (2019, Theorem 12); the proof above gives a direct contraction-specific rank argument.

B State-tracking preliminaries and arithmetic conventions

This section fixes the tasks, recurrent model, and arithmetic conventions used in the single-layer results of Appendix C and the multilayer results of Appendix D. We follow the state-tracking setup of Grazzi et al. (2025); Siems et al. (2025), making explicit the distinction between tracking a state, recognizing a language, and computing a weighted-automaton output. We begin with the group concepts used throughout the proofs. The matrix-group ingredients for the reduction are collected in Appendix C.6.1.

B.1 Groups: basic concepts and notation

Groups, subgroups, and examples. A *group* G is a set with an associative composition rule: $(ab)c = a(bc)$. It has an identity e, satisfying $eg = ge = g$, and every g has an inverse g^{-1}, satisfying $gg^{-1} = g^{-1}g = e$. One can think of group elements as reversible updates: multiplication composes updates, e does nothing, and the inverse undoes an update. The order of composition matters in general; a group is *abelian* if $ab = ba$ for all its elements. A *subgroup* $H \subseteq G$ contains e and is closed under composition and inverses, so it is itself a group with the same operation.

We write $|G|$ for the number of elements of a finite group. The *order* of g is the smallest positive integer m such that $g^m = e$, if one exists. The subgroup *generated* by $g_1, \dots, g_k$, written $\langle g_1, \dots, g_k \rangle$, consists of all finite products of these elements and their inverses, including e. A group is *cyclic* if one element generates it. Our examples are $\mathbb{Z}_n$, the integers under addition modulo n; S_n, the permutations of n objects; and D_n, the dihedral group of $2n$ elements describing rotations and reflections of a regular n-gon (for $n \ge 3$). The subgroup $A_n \subseteq S_n$ consists of *even permutations*, those expressible as an even number of swaps of two objects.

Permutation notation. In *cycle notation*, (12) swaps 1 and 2, while (1245) sends $1 \mapsto 2 \mapsto 4 \mapsto 5 \mapsto 1$, leaving unlisted objects fixed. A swap is a *transposition*. Disjoint cycles act on separate objects; for example, $(1245)(36)$ also swaps 3 and 6. Products are composed from right to left: $(12)(23) = (123)$ applies (23) first. Disjoint cycles commute, but overlapping cycles need not: reversing this product gives $(23)(12) = (132)$.

Maps that preserve composition.

Definition B.1 (Homomorphisms and isomorphisms)**.** A map $\phi : G \to H$ between groups is a *homomorphism* if $\phi(ab) = \phi(a)\phi(b)$ for every $a, b \in G$. It is an *isomorphism* if it is also bijective (one-to-one and onto); we write $G \cong H$ when such a map exists.



Figure 9: Example of the S_3 permutation task, analogous to the shell game with 3 hidden objects.

A homomorphism preserves how updates compose but may map different elements to the same output. An isomorphism is a relabelling that preserves the entire group structure. A homomorphism also preserves the identity and inverses. The *kernel* $\ker \phi = \{g \in G : \phi(g) = e_H\}$ contains the elements mapped to the target identity. It is trivial, $\ker \phi = \{e\}$, exactly when ϕ is injective. The *fiber* $\phi^{-1}(h) = \{g \in G : \phi(g) = h\}$ collects all elements with one specified output h. For example, permutation parity is a homomorphism $S_n \to \mathbb{Z}_2$ whose kernel is A_n.

Cosets and index. For a subgroup $H \subseteq G$, the *left coset* $aH = \{ah : h \in H\}$ is a copy of H obtained by multiplying all its elements by a on the left. The distinct cosets partition G: every element belongs to exactly one of them. The *index* $[G : H]$ counts how many distinct cosets are needed to cover G. Each coset has the same size as H, so for finite groups $[G : H] = |G|/|H|$. Thus index two means that G splits into two copies of H: H itself and one other coset. For example, the rotation subgroup of D_n has index two: one coset contains the rotations, the other the reflections. We use this count in the three-dimensional A_5 obstruction (Theorem 17). Every nonempty fiber of a homomorphism is a coset of its kernel: if $\phi(a) = h$, then $\phi^{-1}(h) = a \ker \phi$. Thus all attained outputs have the same number of preimages when the kernel is finite.

Conjugation and normal subgroups. Elements $a, b \in G$ are *conjugate* if $b = uau^{-1}$ for some $u \in G$. This operation is *conjugation* by u; for permutations it corresponds to relabelling the objects. Similarly, $uHu^{-1} = \{uhu^{-1} : h \in H\}$ is a conjugate subgroup. A subgroup N is *normal* if $uNu^{-1} = N$ for every $u \in G$: conjugation preserves the subgroup as a set, though it may permute its elements. Kernels are normal, since $\phi(ugu^{-1}) = \phi(u)\phi(g)\phi(u)^{-1} = e_H$ whenever $g \in \ker \phi$. A surjective (onto) homomorphism also maps normal subgroups to normal subgroups of its codomain.

Quotient groups. For a normal subgroup $N \subseteq G$, the *quotient group* G/N treats each coset as one element:

$$G/N = \{aN : a \in G\}, \qquad (aN)(bN) = (ab)N.$$

We multiply cosets by multiplying their representatives; normality ensures that the result does not depend on which representatives we choose. The identity is the coset N. For example, S_n/A_n records only whether a permutation is even or odd and is isomorphic to $\mathbb{Z}_2$ for $n \geq 2$.

B.2 State-transition systems and tracking

Definition B.2 (State tracking)**.** A state-transition system consists of a finite input alphabet Σ, a state space $\mathcal{S}$, an initial state $s_0 \in \mathcal{S}$, and a map $T_a : \mathcal{S} \to \mathcal{S}$ for each $a \in \Sigma$. For a word $w = a_1 \cdots a_T \in \Sigma^*$, its states satisfy $s_t = T_{a_t}(s_{t-1})$. For a recurrent model with hidden updates F_a and initial state h_0, a hidden state is *reachable* if it is obtained from h_0 by processing some finite input word. This includes h_0 itself, reached by the empty word. The model *tracks* the system if a fixed decoder d satisfies

$$d(h_0) = s_0, \qquad d(F_a(h)) = T_a(d(h)) \quad \text{for every reachable } h \text{ and every } a \in \Sigma.$$

Thus $d(h_t) = s_t$ at every prefix of every word, including the empty word. The decoder need not be injective, and $\mathcal{S}$ need not be finite.

Definition B.3 (Finite reachability)**.** A recurrent model has *finite reachability*, or is *finite-state*, if its reachable set $\mathcal{R}$ is finite under its exact updates.

For each target system, the network dimensions, parameters, initial state, and decoder are fixed independently of input length. Under the polynomial-precision convention of Appendix B.5, each scalar is represented exactly using a number of bits bounded by a fixed polynomial in the input length.

Group prefix products. Using the group notation in Appendix B.1, take a finite group G with identity e and set $\Sigma = \mathcal{S} = G$, $s_0 = e$, and $T_g(s) = gs$. The desired state is $s_t = g_t \cdots g_1$. Throughout our group results, inputs range over *all* of G. Restricting inputs to a chosen set of generators gives a different task. For example, tracking $\mathbb{Z}_n$ means computing the running sum modulo n. We use "solving the group-word problem of G" to mean tracking this system in the sense of Definition B.2, that is, recovering every prefix product.

Deterministic finite automata. A DFA is a tuple $(Q, \Sigma, \delta, q_0, F)$, where Q is a finite state set, $q_0 \in Q$ is the initial state, $F \subseteq Q$ is the accepting set, and $\delta : Q \times \Sigma \to Q$ is the transition function. Indexing coordinates by Q, define the column-state transition matrix by

$$\boldsymbol{M}_a \boldsymbol{e}_q = \boldsymbol{e}_{\delta(q,a)}, \qquad \boldsymbol{p}_0 = \boldsymbol{e}_{q_0}, \qquad \boldsymbol{p}_t = \boldsymbol{M}_{a_t} \boldsymbol{p}_{t-1}.$$

Each column of $\boldsymbol{M}_a$ is one-hot. Tracking recovers the current state q_t; recognition only asks whether $q_t \in F$. For example, a two-state parity automaton swaps its states on input 1, preserves them on input 0, and accepts in its even state. The functions $Q \to Q$ induced by words form a finite monoid under composition: composition is associative and the empty word acts as the identity. If all symbol maps are bijective, this monoid is a group.

Rational weighted finite automata. A WFA over $\mathbb{Q}$ has a finite alphabet Σ, an initial vector $\boldsymbol{\alpha}_0 \in \mathbb{Q}^n$, a matrix $\boldsymbol{M}_a \in \mathbb{Q}^{n \times n}$ for each symbol, and a final vector $\boldsymbol{\omega} \in \mathbb{Q}^n$. Its weighted state and scalar output are

$$\boldsymbol{p}_0 = \boldsymbol{\alpha}_0, \qquad \boldsymbol{p}_t = \boldsymbol{M}_{a_t} \boldsymbol{p}_{t-1}, \qquad y_t = \boldsymbol{\omega}^\top \boldsymbol{p}_t = \boldsymbol{\omega}^\top \boldsymbol{M}_{a_t} \cdots \boldsymbol{M}_{a_1} \boldsymbol{\alpha}_0. \tag{8}$$

Unlike a DFA state, $\boldsymbol{p}_t$ can range over an infinite set. Tracking $\boldsymbol{p}_t$ suffices to compute y_t by applying the final vector $\boldsymbol{\omega}$. The converse need not hold: distinct weighted states can have the same scalar output, so computing y_t need not recover $\boldsymbol{p}_t$.

B.3 Recurrent model, initialization, and readout

All matrices are real, and $\boldsymbol{I}_d$ denotes the $d \times d$ identity matrix. A linear RNN head has a matrix-valued state and an affine update

$$\boldsymbol{S}_t = \boldsymbol{A}(\boldsymbol{x}_t) \boldsymbol{S}_{t-1} + \boldsymbol{B}(\boldsymbol{x}_t), \qquad \boldsymbol{S}_t \in \mathbb{R}^{n \times d_v}. \tag{9}$$

The functions are position-independent: time dependence comes through the layer input $\boldsymbol{x}_t$. In CKDA,

$$\boldsymbol{A}(\boldsymbol{x}_t) = (\boldsymbol{I} - \beta_t \boldsymbol{k}_t \boldsymbol{k}_t^\top) \operatorname{Diag}(\boldsymbol{\alpha}_t), \qquad \boldsymbol{B}(\boldsymbol{x}_t) = \beta_t \boldsymbol{k}_t \boldsymbol{v}_t^\top, \qquad \|\boldsymbol{k}_t\|_2 = 1.$$

Unless stated otherwise, $\beta_t \in [0, 2]$ and $\boldsymbol{\alpha}_t \in [-1, 1]^n$. These theoretical ranges include their endpoints and zero gates. The practical gate parameterization and its magnitude floor are described separately in Appendix E. The regular-language and WFA simulation in Appendix D explicitly allows $\beta_t > 2$. The S_5 lower bound allows arbitrary additive matrices $\boldsymbol{B}(\boldsymbol{x}_t)$, and therefore also covers the CKDA additive term.

Heads and layers. Independent heads have separate recurrent states and receive the same layer input; their updates do not depend on the previous states of other heads. A joint decoder may use their outputs. Stacked layers may apply nonlinear feed-forward maps between recurrent layers, and may pass the current input along with the recurrent readout.

Initial state and decoder. We permit a prescribed initial state; in particular, the group constructions use $\boldsymbol{S}_0 = \boldsymbol{I}_d$ and $\boldsymbol{B}(\boldsymbol{x}_t) = 0$ at every step. A head is usually read through $\boldsymbol{S}_t^\top \boldsymbol{q}_t$, followed by a feed-forward decoder, with $\boldsymbol{q}_t$ determined by the current layer input. We allow sufficiently expressive feed-forward maps to implement finite lookups, as in Grazzi et al. (2025); Siems et al. (2025); Merrill et al. (2026a). In the WFA construction, the readout is a linear functional of the accumulator with coefficients selected from a finite lookup.

Query projection. For the finite orthogonal constructions below, a single fixed query retains all the information in the hidden matrix. Here $O(d)$ denotes the group of $d \times d$ orthogonal matrices.

Lemma 11 (A fixed query distinguishes finite orthogonal states)**.** *For every finite subgroup $\Gamma \subset O(d)$, there is a unit query $\boldsymbol{q} \in \mathbb{R}^d$ such that $\boldsymbol{S} \mapsto \boldsymbol{S}^\top \boldsymbol{q}$ is injective on Γ. For a group tracker with reachable states in Γ, initial state $\boldsymbol{S}_0 = \boldsymbol{I}_d$, updates $\boldsymbol{S} \mapsto \boldsymbol{A}_h \boldsymbol{S}$, and state decoder p, the output decoder $\pi(\boldsymbol{S}^\top \boldsymbol{q}) := p(\boldsymbol{S})$ is therefore well defined and satisfies*

$$\pi(\boldsymbol{q}) = e, \qquad \pi((\boldsymbol{A}_h \boldsymbol{S})^\top \boldsymbol{q}) = h\, \pi(\boldsymbol{S}^\top \boldsymbol{q}). \tag{10}$$

Distinct reachable query outputs have positive separation, so the precision needed to distinguish them is independent of sequence length.

Proof. Choose a nonzero vector $\boldsymbol{q}$ outside the fixed-point subspaces of all nonidentity matrices in Γ, and normalize it. Each fixed-point subspace of $\boldsymbol{S} \in \Gamma \setminus \{\boldsymbol{I}_d\}$ is proper, and a finite union of proper linear subspaces cannot cover $\mathbb{R}^d$. Let $\boldsymbol{S}_1, \boldsymbol{S}_2 \in \Gamma$. If $\boldsymbol{S}_1^\top \boldsymbol{q} = \boldsymbol{S}_2^\top \boldsymbol{q}$, then $\boldsymbol{S}_2 \boldsymbol{S}_1^\top \boldsymbol{q} = \boldsymbol{q}$. By the choice of $\boldsymbol{q}$, the only matrix in Γ that fixes $\boldsymbol{q}$ is $\boldsymbol{I}_d$. Hence $\boldsymbol{S}_2 \boldsymbol{S}_1^\top = \boldsymbol{I}_d$, which gives $\boldsymbol{S}_1 = \boldsymbol{S}_2$. The output tracking rule follows from that of p. The distinct outputs form a finite set, which has a positive minimum pairwise distance whenever it has at least two elements; a singleton needs no distinction. □

The query can be chosen with algebraic coordinates: first choose a rational vector outside the finitely many excluded proper subspaces, then normalize it. Positive separation concerns the readout of exact reachable states; it does not by itself bound errors accumulated in the recurrence.

B.4 Realization and decoded tracking

Group representations. Recall that a *homomorphism* preserves composition (Definition B.1). An *orthogonal representation* assigns each $g \in G$ a $d \times d$ orthogonal matrix $\rho(g)$ with $\rho(gh) = \rho(g)\rho(h)$. It is *faithful* if distinct group elements receive distinct matrices. Thus a faithful representation identifies G with a matrix group.

Permitted transitions, realization, and tracking. For the orthogonal group constructions we specialize to

$$\boldsymbol{S}_0 = \boldsymbol{I}_d, \qquad \boldsymbol{S}_t = \boldsymbol{A}_{g_t} \boldsymbol{S}_{t-1}.$$

For a unit vector $\boldsymbol{u} \in \mathbb{R}^d$, the Householder matrix $\boldsymbol{H}_{\boldsymbol{u}} = \boldsymbol{I}_d - 2\boldsymbol{u}\boldsymbol{u}^\top$ reflects across the hyperplane $\boldsymbol{u}^\perp$. A *sign diagonal* is a diagonal matrix with entries in $\{-1, +1\}$. We consider all orthogonal transition matrices of a CKDA layer:

$$\mathcal{SH}_d := \{\boldsymbol{H}_{\boldsymbol{u}} \boldsymbol{D} : \|\boldsymbol{u}\| = 1,\ \boldsymbol{D} \text{ a sign diagonal}\}.$$

Its elements are *signed-Householder matrices*. This family contains $\boldsymbol{I}_d = \boldsymbol{H}_{\boldsymbol{e}_1} \boldsymbol{H}_{\boldsymbol{e}_1}$, since the first coordinate vector $\boldsymbol{e}_1$ gives the sign diagonal $\boldsymbol{H}_{\boldsymbol{e}_1} = \mathrm{Diag}(-1, 1, \ldots, 1)$. The family is not generally closed under multiplication.

A realization represents the accumulated group element directly by a matrix. Tracking allows several hidden matrices to represent the same group element, provided a fixed decoder recovers the correct product.

Definition B.4 (Signed-Householder realization and tracking)**.** For a finite group G, choose transitions $\boldsymbol{A}_h \in \mathcal{SH}_d$, $h \in G$, and let $\mathcal{R}$ be the states reachable from $\boldsymbol{I}_d$ under these updates. They *realize G in dimension d* if $h \mapsto \boldsymbol{A}_h$ is a faithful representation, and *track G* if a decoder $p : \mathcal{R} \to G$ satisfies

$$p(\boldsymbol{I}_d) = e, \qquad p(\boldsymbol{A}_h \boldsymbol{S}) = h\, p(\boldsymbol{S}) \quad (h \in G,\ \boldsymbol{S} \in \mathcal{R}). \tag{11}$$

Finite-state tracking means finite reachability as in Definition B.3.

For finite-state tracking, the update rule itself forces the decoder to preserve multiplication. This lets us use group structure even when the decoder was initially allowed to be an arbitrary function.

Lemma 12 (Finite tracking induces a group homomorphism)**.** *For a finite-state signed-Householder tracker, $\mathcal{R} = \Gamma := \langle \boldsymbol{A}_h : h \in G \rangle$, and its decoder $p : \Gamma \to G$ is a surjective homomorphism. Such a tracker has a bijective decoder exactly when its transitions form a realization.*

Proof. The finite set $\mathcal{R}$ contains the identity and is closed under multiplication. For each $\boldsymbol{Q} \in \mathcal{R}$, two powers coincide; invertibility gives $\boldsymbol{Q}^m = \boldsymbol{I}_d$ for some $m \geq 1$. Hence $\boldsymbol{Q}^{-1} = \boldsymbol{Q}^{m-1} \in \mathcal{R}$, so $\mathcal{R} = \Gamma$. Writing $\boldsymbol{Q}_1 = \boldsymbol{A}_{h_t} \cdots \boldsymbol{A}_{h_1}$ and iterating the tracking rule gives $p(\boldsymbol{Q}_1\boldsymbol{Q}_2) = h_t \cdots h_1 p(\boldsymbol{Q}_2) = p(\boldsymbol{Q}_1)p(\boldsymbol{Q}_2)$. Also $p(\boldsymbol{A}_h) = h$, proving surjectivity. If p is bijective, its inverse is a faithful representation with $p^{-1}(h) = \boldsymbol{A}_h$. Conversely, a realization ρ has reachable set $\rho(G)$ and decoder ρ^{-1}. □

B.5 Precision, finite reachability, and states separability

Rounding after each recurrent update generally makes the dynamics nonlinear and differs from casting the exact unrolled state only at the end. The latter convention is used by Merrill et al. (2024); Grazzi et al. (2025); Merrill et al. (2026a) instead use exact rational arithmetic with bounds on representation size. We specify our convention below.

Finite exact datatypes. Our finite-group and DFA constructions use only finitely many exact values, including intermediate computations. They therefore admit a fixed finite datatype $\mathbb{D} \subset \mathbb{R}$, possibly containing irrationals, as in Grazzi et al. (2025, Appendix A.4) and Siems et al. (2025, Appendix B). It need only support the operations reached by the construction, not arbitrary arithmetic. In the block simulation of Peng et al. (2025, Appendix D.2) and our adaptation, finiteness follows from the finite clocks, buffers, DFA states, and partial factor products.

States separability and decoding accuracy. The reduction uses a weaker condition than finite reachability. For joint matrix states, set $\|\boldsymbol{S}\|_F^2 = \sum_{h=1}^{H} \|\boldsymbol{S}^{(h)}\|_F^2$.

Definition B.5 (States separability)**.** A tracker with reachable set $\mathcal{R}$ and decoder $d : \mathcal{R} \to G$ satisfies *states separability* if there is an $\varepsilon > 0$ such that

$$d(\boldsymbol{S}) \neq d(\boldsymbol{S}') \quad \Longrightarrow \quad \|\boldsymbol{S} - \boldsymbol{S}'\|_F \geq \varepsilon \qquad (\boldsymbol{S}, \boldsymbol{S}' \in \mathcal{R}). \tag{12}$$

The reachable states are *bounded* if $\sup_{\boldsymbol{S} \in \mathcal{R}} \|\boldsymbol{S}\|_F < \infty$. The gap and bound must be independent of input length.

Finite exact reachability implies boundedness and states separability, but the converse fails: $T_g(x, y) = (gx, y/2)$ from $(1, 1)$ tracks $G = \{+1, -1\}$ via $d(x, y) = x$, with infinitely many bounded states and label separation 2. Boundedness is also separate from non-expansion: affine translations such as $T(x) = x + 1$ can produce unbounded states.

With separation ε, errors smaller than $\delta < \varepsilon/2$ cannot make differently labelled states indistinguishable; boundedness then permits a finite grid with this decoding accuracy. Without separability, for every $\delta > 0$ two differently labelled states lie less than 2δ apart. An error smaller than δ can move either state to the same point halfway between them, where a decoder cannot tell which group element is correct. Hence no fixed positive error tolerance guarantees correct decoding for all inputs. A particular rounding rule may still distinguish arbitrarily close values on opposite sides of a rounding threshold.

Scope of the lower bounds. Theorem 22 assumes bounded reachable states and states separability for exact affine updates with non-expansive head transitions and a joint decoder. These are the assumptions behind the S_5 lower bounds in Table 1. This concerns decoding accuracy, not accumulated recurrent error: nonlinear rounding or error correction after each update is not covered solely because the stored states are finite.

The real-spectrum obstructions of Grazzi et al. (2025, Thms. 1–2 and Appendix B.2) instead use their finite-datatype casting convention, without finite exact reachability. They exclude parity for nonnegative spectra and one-layer counting modulo $n > 2$ for real spectra. Our S_5 result allows a non-real pair per head and uses the finite-replacement theorem.

Polynomial precision and exact encoding. Our constructions use *polynomial precision over a fixed real algebraic number field* $K \subset \mathbb{R}$, extending the rational setting of Merrill et al. (2026a) to include constants such as $\sqrt{3}/2$ for rotations and $1/\sqrt{2}$ for normalized keys. For a fixed basis $\eta_1, \ldots, \eta_r$ of K over $\mathbb{Q}$, store $x = \sum_{j=1}^{r} c_j \eta_j$ through its rational coefficients c_j, whose numerators

and denominators have polynomial bit length in the input length T. The field, basis, and network parameters are independent of T. Arithmetic is exact because each $\eta_i\eta_j$ is a fixed rational combination of basis elements. Floating-point approximations require a separate error analysis: rotation-angle errors, for example, can accumulate with sequence length.

C SINGLE-LAYER FINITE-GROUP EXPRESSIVITY

We use the tracking and realization definitions of Appendix B.4 and the exact arithmetic convention of Appendix B.5. Inputs range over every element of the target group. We first construct planar and cube realizations, then prove the three-dimensional obstruction and four-dimensional tracker for A_5. We then prove the S_5 obstruction using a finite-replacement theorem. The reduction and its compact-group ingredients are collected at the end of this section, in Appendix C.6.

C.1 CYCLIC AND DIHEDRAL GROUPS

We write $\mathbb{Z}_n$ for the cyclic group of n elements and D_n for its dihedral extension of $2n$ elements, represented by planar rotations and reflections. The following realizations also give finite-state tracking.

Proposition 13 (planar groups)**.** *Every cyclic group $\mathbb{Z}_n$ and every dihedral group D_n has a signed-Householder realization in two dimensions.*

Proof. In the plane, take $\boldsymbol{D}_0 = \mathrm{Diag}(1, -1)$, reflection in the horizontal axis. If $\boldsymbol{S}_\alpha$ denotes reflection in the line making angle α with the horizontal axis, then $\boldsymbol{S}_\alpha\boldsymbol{D}_0$ is a rotation through 2α. Choosing $\alpha = \pi k/n$ realizes every rotation through $2\pi k/n$, so every element in the standard representation of $\mathbb{Z}_n$ has the required form.

The dihedral group adds reflections of the regular n-gon. Each is already a Householder matrix, so choose $\boldsymbol{D} = \boldsymbol{I}$. Thus the same is true for the standard planar representation of D_n. □

C.2 AXIS CRITERION FOR THREE-DIMENSIONAL ROTATIONS

A rotation in $\mathbb{R}^3$ fixes its axis and rotates the perpendicular plane. The following proposition characterizes exactly when such a rotation is a signed-Householder matrix. Its proof uses only the elementary fact that the product of reflections in two planes is a rotation about their line of intersection.

Proposition 14 (Axis test)**.** *Let $\boldsymbol{R} \in SO(3)$ be a rotation about the axis spanned by a unit vector $\boldsymbol{a} \in \mathbb{R}^3$.*

1. *The identity and every half-turn belong to $\mathcal{SH}_3$.*
2. *If its angle is different from 0 and π, then $\boldsymbol{R} = \boldsymbol{H_u D}$ for some sign diagonal $\boldsymbol{D}$ if and only if its rotation axis $\boldsymbol{a}$ is contained in a coordinate plane. Explicitly, this means that $\boldsymbol{a} \perp \boldsymbol{e}_i$ for some coordinate axis $\mathbb{R}\boldsymbol{e}_i$, or equivalently that $\boldsymbol{a}$ has at least one zero coordinate.*

Proof. For part 1, the matrix $2\boldsymbol{a}\boldsymbol{a}^\top - \boldsymbol{I}$ fixes $\boldsymbol{a}$ and sends every $\boldsymbol{x} \perp \boldsymbol{a}$ to $-\boldsymbol{x}$, so it is the half-turn about $\mathbb{R}\boldsymbol{a}$. Moreover,

$$2\boldsymbol{a}\boldsymbol{a}^\top - \boldsymbol{I} = (\boldsymbol{I} - 2\boldsymbol{a}\boldsymbol{a}^\top)(-\boldsymbol{I}) = \boldsymbol{H_a}(-\boldsymbol{I}),$$

which has the required form. The identity case was established when introducing the permitted family.

For part 2, suppose $\boldsymbol{R} = \boldsymbol{H_u D}$. Since $\det \boldsymbol{H_u} = -1$ and $\det \boldsymbol{R} = 1$, the diagonal $\boldsymbol{D}$ has either one or three negative entries. In the latter case $\boldsymbol{D} = -\boldsymbol{I}$ and $\boldsymbol{R} = 2\boldsymbol{u}\boldsymbol{u}^\top - \boldsymbol{I}$, the half-turn from part 1, contradicting the assumption on the angle. Hence $\boldsymbol{D}$ has exactly one negative entry. If that entry is in position i, then $\boldsymbol{D} = \boldsymbol{I} - 2\boldsymbol{e}_i\boldsymbol{e}_i^\top$, which is precisely the reflection across the coordinate plane $\boldsymbol{e}_i^\perp$.

Both factors are now reflections in planes: $\boldsymbol{D}$ reflects in $\boldsymbol{e}_i^\perp$ and $\boldsymbol{H_u}$ reflects in $\boldsymbol{u}^\perp$. Their product rotates about the line $\boldsymbol{e}_i^\perp \cap \boldsymbol{u}^\perp$. This is the axis $\mathbb{R}\boldsymbol{a}$, so $\boldsymbol{a} \perp \boldsymbol{e}_i$; in other words, $\boldsymbol{a}$ lies in the coordinate plane $\boldsymbol{e}_i^\perp$.

Conversely, every rotation in $SO(3)$ is a product of reflections in two planes through its axis, separated by half the rotation angle. If $\boldsymbol{a} \perp \boldsymbol{e}_i$, we may choose one plane to be the coordinate plane $\boldsymbol{e}_i^{\perp}$. Its reflection is $\boldsymbol{D}_i = \boldsymbol{I} - 2\boldsymbol{e}_i\boldsymbol{e}_i^{\top}$; the other reflection is therefore $\boldsymbol{H}_{\boldsymbol{u}} := \boldsymbol{R}\boldsymbol{D}_i$, and $\boldsymbol{R} = \boldsymbol{H}_{\boldsymbol{u}}\boldsymbol{D}_i$. □

C.3 Realizing S_4 via the cube rotational symmetries

The group S_4 has a familiar geometric realization: it is isomorphic to the group of rotational symmetries of a cube. We now show that rotating the cube to a suitable orientation makes all of these rotations admissible at once. Since restricting a faithful representation to a subgroup preserves both faithfulness and the permitted matrix form, the same construction will also realize $A_4 \subset S_4$.

Figure 10 shows the face and body-diagonal axes; edge axes give only half-turns.

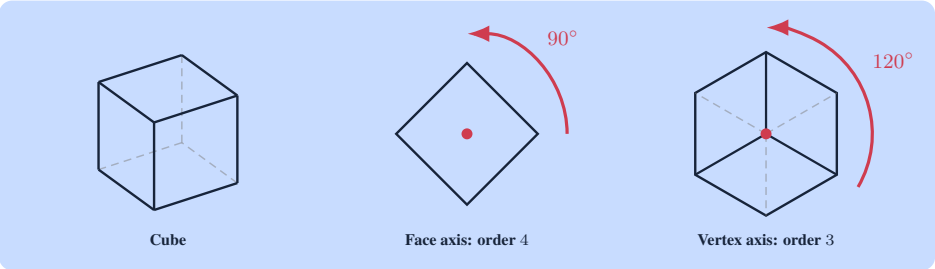

Figure 10: Smallest symmetry-preserving rotations (excluding half-turns) for the cube, grouped by axis. The red point marks the axis viewed end-on; visible edges are solid and hidden edges are dashed.

Proposition 15 (cube realization)**.** *The group S_4, realized as the rotational symmetry group of a cube, has a signed-Householder realization in three dimensions. Restricting it to the subgroup A_4 gives a realization of A_4 in the same dimension.*

Proof. The orientation-preserving symmetry group of a cube is isomorphic to S_4: it permutes the four unoriented body diagonals faithfully. Its non-half-turn rotation axes are

- the three face axes, supporting rotations through $90°$.
- the four body-diagonal axes, supporting rotations through $120°$.

All remaining non-identity rotations are half-turns and are automatically admissible by the axis test (Proposition 14).

Place the cube first in its standard orientation. The three face axes and four body-diagonal axes are represented, respectively, by

$$\mathcal{F} = \{(1,0,0),(0,1,0),(0,0,1)\}, \qquad \mathcal{B} = \{(1,1,1),(1,1,-1),(1,-1,1),(-1,1,1)\}.$$

Every vector in $\mathcal{B}$ has three nonzero coordinates, so this orientation does not pass the axis test of Proposition 14.

We can change the orientation of these axes while preserving the group being represented. Write $\rho(h)$ for the matrix of a cube symmetry in the standard orientation. Reorienting the whole cube by a rotation $\boldsymbol{R}$ replaces each symmetry matrix by $\boldsymbol{R}\rho(h)\boldsymbol{R}^{\top}$, an operation called *conjugation*. Using the same $\boldsymbol{R}$ for every symmetry preserves multiplication, since

$$(\boldsymbol{R}\rho(h)\boldsymbol{R}^{\top})(\boldsymbol{R}\rho(k)\boldsymbol{R}^{\top}) = \boldsymbol{R}\rho(hk)\boldsymbol{R}^{\top}.$$

At the same time, each rotation axis $\boldsymbol{u}$ moves to $\boldsymbol{R}\boldsymbol{u}$. Our task is therefore to find one rotation that places all seven axes in coordinate planes. We choose a $45°$ rotation about the third coordinate axis:

$$\boldsymbol{R} = \begin{pmatrix} 1/\sqrt{2} & 1/\sqrt{2} & 0 \\ -1/\sqrt{2} & 1/\sqrt{2} & 0 \\ 0 & 0 & 1 \end{pmatrix} \in SO(3).$$

Applying $\boldsymbol{R}$ to each of the seven axes gives

$$\boldsymbol{R}\mathcal{F} = \left\{ \frac{(1,-1,0)}{\sqrt{2}}, \frac{(1,1,0)}{\sqrt{2}}, (0,0,1) \right\},$$
$$\boldsymbol{R}\mathcal{B} = \{(\sqrt{2},0,1), (\sqrt{2},0,-1), (0,-\sqrt{2},1), (0,\sqrt{2},1)\}.$$

Thus the rotation does not spoil the three face axes: they still have a zero coordinate, as do all four transformed body diagonals. The axis test (Proposition 14) therefore shows that every cube rotation is a signed-Householder matrix. Restriction to A_4 proves the remaining claim. □

C.4 How to track A_5

The orientation-preserving symmetry group of the icosahedron is isomorphic to A_5. We prove that A_5 has no finite-state signed-Householder tracker in three dimensions, then construct one in four dimensions. The latter uses a many-to-one decoder, so the accumulated hidden matrix carries more information than the group element being tracked.

An icosahedron has 12 vertices, 20 faces, and 30 edges. Pairing each feature with its opposite gives the rotation axes below. An axis has *order* k if the rotations about it form a cyclic group of k elements; equivalently, its smallest positive rotation has angle $2\pi/k$ and returns to the identity after k applications. Thus there are

$$6 \text{ axes of order } 5, \qquad 10 \text{ axes of order } 3, \qquad 15 \text{ axes of order } 2.$$

The order-2 axes do not matter: the axis test (Proposition 14) already makes every half-turn admissible. The difficulty comes from the other sixteen axes. Figure 11 illustrates these two relevant axis types.

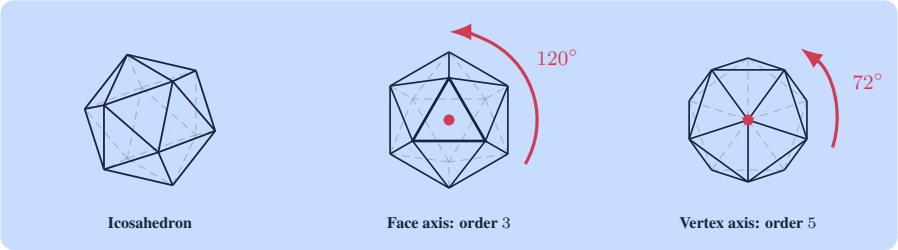

Figure 11: Smallest symmetry-preserving rotations (excluding half-turns) for the icosahedron, grouped by axis. The red point marks the axis viewed end-on; visible edges are solid and hidden edges are dashed.

By the axis test (Proposition 14), representing all odd-order rotations would require three coordinate planes to cover all sixteen axes. We prove the stronger, orientation-independent fact that *any* plane covers at most four.

C.4.1 The obstruction to finite-state tracking in three dimensions

A group with more than one element is *simple* if its only normal subgroups are $\{e\}$ and the whole group. We use the classical fact that A_5 is a noncyclic simple group with 60 elements (Conrad, n.d.).

Lemma 16 (icosahedral plane bound)**.** *Among the six order-5 axes and ten order-3 axes of an icosahedron, no plane through the center contains more than four axes.*

Proof. Let $\varphi = (1+\sqrt{5})/2$ be the golden ratio. In the usual coordinates, representative vectors for the six unoriented vertex axes (the order-5 axes) are

$$\mathcal{V} = \{(0,1,\pm\varphi),\ (1,\pm\varphi,0),\ (\varphi,0,\pm 1)\}.$$

The face centers form the vertices of the dual dodecahedron. Representatives of their ten unoriented axes (the order-3 axes) are

$$\mathcal{T} = \{(1,1,1),(1,1,-1),(1,-1,1),(-1,1,1)\}$$
$$\cup\, \{(0,\varphi,\pm(\varphi-1)),(\varphi,\pm(\varphi-1),0),(\varphi-1,0,\pm\varphi)\}.$$

It suffices to check that no three axes in either family are coplanar. Indeed, substituting the displayed vectors and using $\varphi^2 = \varphi + 1$ gives the following possible absolute determinants of three distinct representatives:

family	$\lvert\det(\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3)\rvert$
$\mathcal{V}$	$1+\sqrt{5},\ 3+\sqrt{5}$
$\mathcal{T}$	$\sqrt{5}-1,\ 2,\ 1+\sqrt{5},\ 2\sqrt{5},\ 4$

All these determinants are nonzero. Thus a plane contains at most two vertex axes and at most two face axes, hence at most four axes in total. The bound is tight: the plane $x = 0$ contains

$$(0, 1, \varphi),\ (0, 1, -\varphi),\ (0, \varphi, \varphi - 1),\ (0, \varphi, -(\varphi - 1)).$$

□

Using the group facts from Appendix B.1, we now turn the geometric bound into a tracking obstruction. We also use the classification of finite subgroups of $SO(3)$ (Olive, 2019, Sec. 3): cyclic groups, dihedral groups, and the rotational symmetry groups of the tetrahedron, cube, and icosahedron, isomorphic to A_4, S_4, and A_5. These occur up to an orthogonal change of coordinates.

Theorem 17 (No finite-state three-dimensional A_5 tracker)**.** *The group A_5 admits no finite-state signed-Householder tracking in dimension three, even with an arbitrary decoder. In particular, it admits no signed-Householder realization in that dimension.*

Proof. Suppose a finite-state signed-Householder tracker exists, with transitions $\boldsymbol{A}_h$ and decoder p. The homomorphism property of finite tracking (Lemma 12) gives a finite hidden group $\Gamma = \langle \boldsymbol{A}_h : h \in A_5 \rangle \subset O(3)$ and a surjective homomorphism $p : \Gamma \to A_5$ satisfying $p(\boldsymbol{A}_h) = h$.

Turn the hidden matrices into rotations. In three dimensions, multiplying an orientation-reversing orthogonal matrix by $-\boldsymbol{I}_3$ makes it orientation-preserving. Define

$$\psi(\boldsymbol{Q}) := (\det \boldsymbol{Q})\boldsymbol{Q}, \qquad H := \psi(\Gamma) \subset SO(3).$$

Indeed, $\det \psi(\boldsymbol{Q}) = (\det \boldsymbol{Q})^4 = 1$. This map also preserves multiplication:

$$\psi(\boldsymbol{Q}_1\boldsymbol{Q}_2) = (\det \boldsymbol{Q}_1)(\det \boldsymbol{Q}_2)\boldsymbol{Q}_1\boldsymbol{Q}_2 = \psi(\boldsymbol{Q}_1)\psi(\boldsymbol{Q}_2).$$

Thus H is a finite rotation group. The map identifies at most a matrix and its negative, since its kernel K is contained in $\{\pm\boldsymbol{I}_3\}$. These identifications cannot change the decoded group element. Since K is a kernel, it is normal in Γ. As p is surjective, its image $p(K)$ is therefore normal in A_5. This image has at most two elements, so simplicity of A_5 forces $p(K) = \{e\}$. Consequently the decoder descends to a well-defined surjective homomorphism

$$\bar{p} : H \to A_5, \qquad \bar{p}(\psi(\boldsymbol{Q})) := p(\boldsymbol{Q}).$$

To check that this is well defined, if $\psi(\boldsymbol{Q}_1) = \psi(\boldsymbol{Q}_2)$, then $\boldsymbol{Q}_1\boldsymbol{Q}_2^{-1} \in K$, so $p(\boldsymbol{Q}_1) = p(\boldsymbol{Q}_2)$.

Identify the resulting rotation group. The classification of finite subgroups of $SO(3)$ now forces H to be an icosahedral rotation group. The tetrahedral and cubic groups have only 12 and 24 elements, so cannot map onto A_5. A cyclic or dihedral group has a cyclic normal subgroup of index at most two: the whole group in the cyclic case, or the rotations in the dihedral case. Its image under $\bar{p}$ would be a cyclic normal subgroup of A_5, also with at most two cosets. Indeed, a surjective homomorphism maps each source coset onto a target coset, so their number cannot increase. Simplicity forces that image to be trivial or all of A_5: the former has 60 cosets, while the latter is not cyclic. These cases are therefore also impossible.

It follows that $|H| = 60$, so the surjective map $\bar{p} : H \to A_5$ is an isomorphism. In particular, the matrices $\psi(\boldsymbol{A}_h)$, one for each $h \in A_5$, exhaust H, because $\bar{p}(\psi(\boldsymbol{A}_h)) = h$. Each is still signed-Householder: if $\boldsymbol{A}_h = \boldsymbol{H}_{\boldsymbol{u}_h}\boldsymbol{D}_h$, then

$$\psi(\boldsymbol{A}_h) = \boldsymbol{H}_{\boldsymbol{u}_h}\big((\det \boldsymbol{A}_h)\boldsymbol{D}_h\big),$$

and the factor in parentheses is again a sign diagonal.

Apply the geometric obstruction. We have obtained an orientation of the icosahedron in which every rotation is signed-Householder. But its six order-5 axes and ten order-3 axes cannot all lie in the three

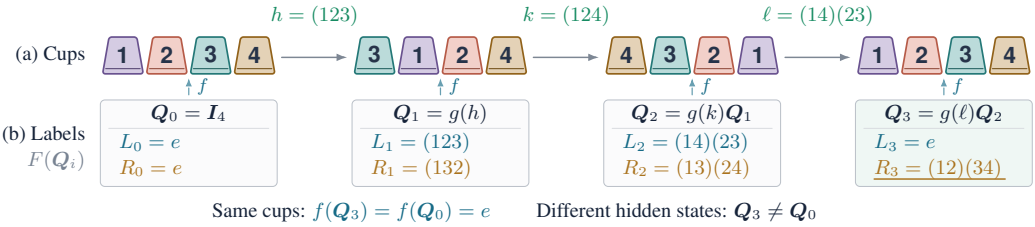

Figure 12: Many-to-one tracking, illustrated by $h = (123)$, $k = (124)$, and $\ell = (kh)^{-1} = (14)(23)$. The cups show identities at successive positions; object 5 is fixed and omitted. Each box groups the two labels $F(\boldsymbol{Q}_i) = (L_i, R_i)$ of one hidden matrix, where $L_i = f(\boldsymbol{Q}_i)$ and $R_i = f_-(\boldsymbol{Q}_i)$. An input a gives $\boldsymbol{Q} \mapsto g(a)\boldsymbol{Q}$ and $(L, R) \mapsto (aL, a^{-1}R)$. The cups and L return to their initial values, but $R_3 = (12)(34) \neq R_0 = e$; hence $\boldsymbol{Q}_3 \neq \boldsymbol{Q}_0$, although both decode to e.

coordinate planes: the bound in Lemma 16 allows at most four axes per plane, hence at most 12 of the 16 axes altogether. A nonidentity rotation about any remaining axis fails the signed-Householder axis test (Proposition 14), a contradiction.

Finally, a signed-Householder realization would itself give a finite-state tracker, with the inverse representation as decoder. It is therefore excluded as well. □

C.4.2 A FOUR-DIMENSIONAL CONSTRUCTION THAT TRACKS A_5

Four dimensions admit the finite-state tracker ruled out above in three dimensions. We encode each input rotation in $SO(3)$ as a signed-Householder transition in $SO(4)$ and decode the accumulated product. The decoder may be many-to-one, as illustrated in Figure 12.

The underlying $SO(4)$ geometry is the classical left/right quaternionic decomposition of four-dimensional rotations (Mebius, 2005); the A_5 case uses the binary icosahedral lift (Choi & Lee, 2018).

The following lemma isolates the two maps used here and later in the S_5 obstruction. A planar rotation fixes the orthogonal complement of one two-dimensional plane; its unsigned angle lies in $[0, \pi]$.

Lemma 18 (Two rotation maps from four to three dimensions)**.** *For $\boldsymbol{Q} \in SO(4)$, write*

$$\boldsymbol{Q} = \begin{pmatrix} \alpha & \boldsymbol{r}^\top \\ \boldsymbol{s} & \boldsymbol{C} \end{pmatrix}, \qquad \alpha \in \mathbb{R}, \quad \boldsymbol{r}, \boldsymbol{s} \in \mathbb{R}^3, \quad \boldsymbol{C} \in \mathbb{R}^{3\times 3}.$$

Let $[\boldsymbol{r}]_\times$ denote the cross-product matrix: $[\boldsymbol{r}]_\times \boldsymbol{x} = \boldsymbol{r} \times \boldsymbol{x}$. The quadratic maps

$$f(\boldsymbol{Q}) = \alpha\boldsymbol{C} - \boldsymbol{s}\boldsymbol{r}^\top - \boldsymbol{C}[\boldsymbol{r}]_\times, \qquad f_-(\boldsymbol{Q}) = \alpha\boldsymbol{C} - \boldsymbol{s}\boldsymbol{r}^\top + \boldsymbol{C}[\boldsymbol{r}]_\times \tag{13}$$

are homomorphisms from $SO(4)$ to $SO(3)$. Their common kernel is $\ker f \cap \ker f_- = \{\boldsymbol{I}_4, -\boldsymbol{I}_4\}$, and each sends a planar rotation to a three-dimensional rotation with the same unsigned angle.

Proof. A decoder that preserves multiplication. For $\boldsymbol{x} \in \mathbb{R}^3$, define the skew-symmetric block matrix

$$\boldsymbol{J}(\boldsymbol{x}) := \begin{pmatrix} 0 & \boldsymbol{x}^\top \\ -\boldsymbol{x} & -[\boldsymbol{x}]_\times \end{pmatrix}.$$

The matrices $\boldsymbol{J}(\boldsymbol{x})$ form a three-dimensional linear space, with $\boldsymbol{x}$ uniquely determined by the top-right block. We will show that $\boldsymbol{Q}\boldsymbol{J}(\boldsymbol{x})\boldsymbol{Q}^\top$ is again of the form $\boldsymbol{J}(\boldsymbol{y})$ for every $\boldsymbol{Q} \in SO(4)$, with $\boldsymbol{y} = f(\boldsymbol{Q})\boldsymbol{x}$. It suffices first to check that this form is preserved by two elementary types of rotation. First, for $\boldsymbol{T}_{\boldsymbol{R}} := \mathrm{Diag}(1, \boldsymbol{R})$ with $\boldsymbol{R} \in SO(3)$, block multiplication and the identity $\boldsymbol{R}[\boldsymbol{x}]_\times \boldsymbol{R}^\top = [\boldsymbol{R}\boldsymbol{x}]_\times$ give

$$\boldsymbol{T}_{\boldsymbol{R}}\boldsymbol{J}(\boldsymbol{x})\boldsymbol{T}_{\boldsymbol{R}}^\top = \boldsymbol{J}(\boldsymbol{R}\boldsymbol{x}).$$

Second, let $\boldsymbol{U}_\phi$ rotate the first two coordinate directions in $\mathbb{R}^4$, and let $\boldsymbol{R}_\phi$ rotate the last two coordinate directions in $\mathbb{R}^3$:

$$\boldsymbol{U}_\phi := \begin{pmatrix} \cos\phi & -\sin\phi & 0 & 0 \\ \sin\phi & \cos\phi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \qquad \boldsymbol{R}_\phi := \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{pmatrix}.$$

Writing $\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3$ for the standard basis of $\mathbb{R}^3$, multiplication gives

$$\begin{aligned} \boldsymbol{U}_\phi \boldsymbol{J}(\boldsymbol{e}_1)\boldsymbol{U}_\phi^\top &= \boldsymbol{J}(\boldsymbol{e}_1), \\ \boldsymbol{U}_\phi \boldsymbol{J}(\boldsymbol{e}_2)\boldsymbol{U}_\phi^\top &= \cos\phi\, \boldsymbol{J}(\boldsymbol{e}_2) + \sin\phi\, \boldsymbol{J}(\boldsymbol{e}_3), \\ \boldsymbol{U}_\phi \boldsymbol{J}(\boldsymbol{e}_3)\boldsymbol{U}_\phi^\top &= -\sin\phi\, \boldsymbol{J}(\boldsymbol{e}_2) + \cos\phi\, \boldsymbol{J}(\boldsymbol{e}_3). \end{aligned}$$

Equivalently, $\boldsymbol{U}_\phi \boldsymbol{J}(\boldsymbol{x})\boldsymbol{U}_\phi^\top = \boldsymbol{J}(\boldsymbol{R}_\phi \boldsymbol{x})$. Rotations among the last three coordinates are included in $\boldsymbol{T}_{\boldsymbol{R}}$. The matrices $\boldsymbol{T}_{\boldsymbol{R}}\boldsymbol{U}_\phi\boldsymbol{T}_{\boldsymbol{R}}^\top$, for suitable $\boldsymbol{R}$, rotate the first coordinate direction with either of the other two coordinate directions. Thus these matrices generate all coordinate-plane rotations in $\mathbb{R}^4$. Every matrix in $SO(4)$ is a product of such rotations, as follows, for example, by eliminating its entries with coordinate-plane rotations. Thus $\boldsymbol{Q}\boldsymbol{J}(\boldsymbol{x})\boldsymbol{Q}^\top$ remains of the form $\boldsymbol{J}(\boldsymbol{y})$ for every $\boldsymbol{Q} \in SO(4)$.

To identify $\boldsymbol{y}$, write $\boldsymbol{Q}$ in the block form used in the statement. The top-right block of $\boldsymbol{Q}\boldsymbol{J}(\boldsymbol{x})\boldsymbol{Q}^\top$ is

$$-(\boldsymbol{r}^\top \boldsymbol{x})\boldsymbol{s}^\top + (\alpha \boldsymbol{x}^\top - \boldsymbol{r}^\top [\boldsymbol{x}]_\times)\boldsymbol{C}^\top = \big(f(\boldsymbol{Q})\boldsymbol{x}\big)^\top,$$

using $\boldsymbol{r}^\top [\boldsymbol{x}]_\times = ([\boldsymbol{r}]_\times \boldsymbol{x})^\top$ and the formula for f in equation 13. Since this block uniquely determines $\boldsymbol{y}$, we obtain

$$\boldsymbol{Q}\boldsymbol{J}(\boldsymbol{x})\boldsymbol{Q}^\top = \boldsymbol{J}(f(\boldsymbol{Q})\boldsymbol{x}) \qquad (\boldsymbol{x} \in \mathbb{R}^3). \tag{14}$$

Applying this identity first with $\boldsymbol{Q}_2$ and then with $\boldsymbol{Q}_1$ gives

$$(\boldsymbol{Q}_1\boldsymbol{Q}_2)\boldsymbol{J}(\boldsymbol{x})(\boldsymbol{Q}_1\boldsymbol{Q}_2)^\top = \boldsymbol{J}\big(f(\boldsymbol{Q}_1)f(\boldsymbol{Q}_2)\boldsymbol{x}\big).$$

The same identity applied directly to $\boldsymbol{Q}_1\boldsymbol{Q}_2$ shows that f preserves multiplication. For the elementary rotations above, f gives $\boldsymbol{R}$ and $\boldsymbol{R}_\phi$, respectively. Both lie in $SO(3)$, so their products do too. We have proved

$$f(\boldsymbol{Q}) \in SO(3), \qquad f(\boldsymbol{I}_4) = \boldsymbol{I}_3, \qquad f(\boldsymbol{Q}_1\boldsymbol{Q}_2) = f(\boldsymbol{Q}_1)f(\boldsymbol{Q}_2). \tag{15}$$

The second map and the common kernel. Define the companion space using

$$\boldsymbol{K}(\boldsymbol{x}) := \begin{pmatrix} 0 & \boldsymbol{x}^\top \\ -\boldsymbol{x} & [\boldsymbol{x}]_\times \end{pmatrix}.$$

The same elementary rotations satisfy

$$\boldsymbol{T}_{\boldsymbol{R}}\boldsymbol{K}(\boldsymbol{x})\boldsymbol{T}_{\boldsymbol{R}}^\top = \boldsymbol{K}(\boldsymbol{R}\boldsymbol{x}), \qquad \boldsymbol{U}_\phi \boldsymbol{K}(\boldsymbol{x})\boldsymbol{U}_\phi^\top = \boldsymbol{K}(\boldsymbol{R}_{-\phi}\boldsymbol{x}).$$

Thus $\boldsymbol{Q}\boldsymbol{K}(\boldsymbol{x})\boldsymbol{Q}^\top$ remains in the space of matrices $\boldsymbol{K}(\boldsymbol{y})$. Reading the top-right block identifies its induced action with f_- in equation 13:

$$\boldsymbol{Q}\boldsymbol{K}(\boldsymbol{x})\boldsymbol{Q}^\top = \boldsymbol{K}(f_-(\boldsymbol{Q})\boldsymbol{x}). \tag{16}$$

Applying this identity twice shows that f_- also preserves multiplication.

The two maps together determine a hidden matrix up to sign. To see this, suppose $f(\boldsymbol{Q}) = f_-(\boldsymbol{Q}) = \boldsymbol{I}_3$. The identities above then give $\boldsymbol{Q}\boldsymbol{J}(\boldsymbol{x})\boldsymbol{Q}^\top = \boldsymbol{J}(\boldsymbol{x})$ and $\boldsymbol{Q}\boldsymbol{K}(\boldsymbol{x})\boldsymbol{Q}^\top = \boldsymbol{K}(\boldsymbol{x})$ for every $\boldsymbol{x}$. Their sums and differences span all skew-symmetric 4×4 matrices. In particular, with $\boldsymbol{e}_0, \ldots, \boldsymbol{e}_3$ now denoting the standard basis of $\mathbb{R}^4$, $\boldsymbol{Q}$ commutes with every $\boldsymbol{B}_{ij} := \boldsymbol{e}_i\boldsymbol{e}_j^\top - \boldsymbol{e}_j\boldsymbol{e}_i^\top$. It therefore also commutes with $-\boldsymbol{B}_{ij}^2$, the orthogonal projector onto the coordinate plane spanned by $\boldsymbol{e}_i, \boldsymbol{e}_j$. Thus $\boldsymbol{Q}$ preserves every coordinate plane, and also every coordinate axis, obtained by intersecting two such planes. This makes $\boldsymbol{Q}$ diagonal. Commutation with $\boldsymbol{B}_{ij}$ then forces its i-th and j-th diagonal entries to agree. Hence $\boldsymbol{Q}$ is a scalar matrix, and orthogonality gives $\boldsymbol{Q} = \pm\boldsymbol{I}_4$. Conversely, both of these matrices satisfy $\boldsymbol{Q}\boldsymbol{X}\boldsymbol{Q}^\top = \boldsymbol{X}$ for every $\boldsymbol{X}$.

Preservation of the unsigned angle. Every planar rotation in $SO(4)$ can be written as $\boldsymbol{V}\boldsymbol{U}_\phi\boldsymbol{V}^\top$ for some $\boldsymbol{V} \in SO(4)$: choose an orthonormal basis aligned with the rotating plane, and orient the two remaining basis vectors so the change of basis has determinant $+1$. Since f preserves products and inverses,

$$f(\boldsymbol{V}\boldsymbol{U}_\phi\boldsymbol{V}^\top) = f(\boldsymbol{V})\boldsymbol{R}_\phi f(\boldsymbol{V})^{-1}.$$

This is $\boldsymbol{R}_\phi$ expressed in another orthonormal basis, so it has the same eigenvalues and unsigned angle. The same argument for f_- uses $f_-(\boldsymbol{U}_\phi) = \boldsymbol{R}_{-\phi}$, whose unsigned angle is identical. □

Proposition 19 (Tracking rotations with signed-Householder transitions)**.** *There is an injective encoder $g : SO(3) \to SO(4)$ such that every $g(\boldsymbol{A})$ is signed-Householder and, for the decoder f in Lemma 18,*

$$f(g(\boldsymbol{A})\boldsymbol{S}) = \boldsymbol{A}\, f(\boldsymbol{S}) \qquad \text{for every } \boldsymbol{A} \in SO(3), \quad \boldsymbol{S} \in SO(4). \tag{17}$$

Iterating this identity decodes any product of encoded rotations to the corresponding product in $SO(3)$; in particular, $f(g(\boldsymbol{A})) = \boldsymbol{A}$.

Explicitly, let $\boldsymbol{A} = R_{\boldsymbol{u},\theta}$ be the three-dimensional rotation through angle $\theta \in [0, \pi]$ about the axis spanned by a unit vector $\boldsymbol{u} \in \mathbb{R}^3$, with the direction of rotation specified by the right-hand rule around $\boldsymbol{u}$. The encoder can be chosen as

$$g(\boldsymbol{A}) = (\boldsymbol{I}_4 - 2\boldsymbol{a}_{\boldsymbol{A}}\boldsymbol{a}_{\boldsymbol{A}}^\top)\boldsymbol{D}_0, \qquad \boldsymbol{a}_{\boldsymbol{A}} := \begin{pmatrix} \cos(\theta/2) \\ \sin(\theta/2)\,\boldsymbol{u} \end{pmatrix}, \qquad \boldsymbol{D}_0 := \mathrm{Diag}(-1, 1, 1, 1). \tag{18}$$

Thus all encoded transitions use the same diagonal sign matrix $\boldsymbol{D}_0$.

Moreover, for every finite subgroup $H \subset SO(3)$, the group

$$\Gamma_H := \langle g(\boldsymbol{A}) : \boldsymbol{A} \in H \rangle \subset SO(4) \qquad \text{satisfies} \qquad |\Gamma_H| \le 2|H|^2. \tag{19}$$

Starting from $\boldsymbol{I}_4$, the reachable matrices therefore form a finite set.

Proof. The encoded transitions. The vector $\boldsymbol{a}_{\boldsymbol{A}}$ in equation 18 has unit norm, so $\boldsymbol{I}_4 - 2\boldsymbol{a}_{\boldsymbol{A}}\boldsymbol{a}_{\boldsymbol{A}}^\top$ is a Householder reflection. The fixed matrix $\boldsymbol{D}_0$ is also a reflection: it changes the sign of the first coordinate. Their product is therefore orthogonal with determinant $+1$, and has the required signed-Householder form. At $\theta = 0$, the construction gives $g(\boldsymbol{I}_3) = \boldsymbol{I}_4$ independently of the chosen axis. At $\theta = \pi$, the two choices $\boldsymbol{u}$ and $-\boldsymbol{u}$ replace $\boldsymbol{a}_{\boldsymbol{A}}$ by its negative and hence give the same Householder matrix. Thus the encoder is well defined for every rotation.

Multiplying the two factors gives

$$g(R_{\boldsymbol{u},\theta}) = \begin{pmatrix} c & -s\boldsymbol{u}^\top \\ s\boldsymbol{u} & \boldsymbol{I}_3 + (c-1)\boldsymbol{u}\boldsymbol{u}^\top \end{pmatrix}, \qquad c := \cos\theta, \quad s := \sin\theta. \tag{20}$$

This matrix rotates the plane spanned by the first coordinate vector $\boldsymbol{e}_0 = (1, 0, 0, 0)^\top$ and $(0, \boldsymbol{u})^\top$ through angle θ, and fixes the orthogonal complement of that plane.

Decoding the accumulated product. For the blocks in equation 20,

$$\alpha\boldsymbol{C} - \boldsymbol{s}\boldsymbol{r}^\top = \boldsymbol{u}\boldsymbol{u}^\top + c(\boldsymbol{I}_3 - \boldsymbol{u}\boldsymbol{u}^\top), \qquad -\boldsymbol{C}[\boldsymbol{r}]_\times = s[\boldsymbol{u}]_\times,$$

where the second identity uses $\boldsymbol{u}^\top[\boldsymbol{u}]_\times = 0$. Hence Rodrigues' rotation formula gives

$$f(g(\boldsymbol{A})) = \boldsymbol{u}\boldsymbol{u}^\top + c(\boldsymbol{I}_3 - \boldsymbol{u}\boldsymbol{u}^\top) + s[\boldsymbol{u}]_\times = R_{\boldsymbol{u},\theta} = \boldsymbol{A}. \tag{21}$$

This also proves injectivity: if $g(\boldsymbol{A}) = g(\boldsymbol{B})$, applying f gives $\boldsymbol{A} = \boldsymbol{B}$. Combining equation 15 and equation 21 proves equation 17 for every hidden matrix in $SO(4)$.

Finitely many hidden states for a finite input group. Use the second homomorphism f_- from Lemma 18. Substituting equation 20 now reverses the sign of the sine term in Rodrigues' formula, so

$$f_-(g(\boldsymbol{A})) = R_{\boldsymbol{u},-\theta} = \boldsymbol{A}^{-1}. \tag{22}$$

The joint map $F(\boldsymbol{Q}) := (f(\boldsymbol{Q}), f_-(\boldsymbol{Q}))$ preserves products. By Lemma 18, $F(\boldsymbol{P}) = F(\boldsymbol{Q})$ exactly when $\boldsymbol{P} = \boldsymbol{Q}$ or $\boldsymbol{P} = -\boldsymbol{Q}$: apply the common-kernel statement to $\boldsymbol{P}\boldsymbol{Q}^{-1}$. Thus each attained pair of outputs corresponds to exactly two matrices, $\boldsymbol{Q}$ and $-\boldsymbol{Q}$.

Now let $H \subset SO(3)$ be finite. For every encoded generator,

$$F(g(\boldsymbol{A})) = (\boldsymbol{A}, \boldsymbol{A}^{-1}) \in H \times H.$$

Since $H \times H$ is closed under multiplication and inverses, $F(\Gamma_H) \subseteq H \times H$. There are at most $|H|^2$ such pairs and at most two hidden matrices for each pair. Consequently $|\Gamma_H| \leq 2|H|^2$, proving equation 19. □

Applying Proposition 19 to the icosahedral group gives our four-dimensional A_5 tracker.

Corollary 20 (Four-dimensional A_5 tracker)**.** *The group A_5 admits finite-state signed-Householder tracking in dimension four, implemented by one CKDA layer using the encoder and decoder of Proposition 19. There are at most* 7200 *reachable hidden matrices, and the decoder is many-to-one on this set.*

Proof. Identify A_5 with the icosahedral rotation group and apply Proposition 19 with $H = A_5$. Take $\boldsymbol{A}_h = g(h)$, $\boldsymbol{S}_0 = \boldsymbol{I}_4$, and zero additive terms. The bound follows from $2|A_5|^2 = 7200$.

For the many-to-one claim, choose noncommuting $h, k \in A_5$ and set $\ell = (hk)^{-1}$. The two maps from Lemma 18 satisfy

$$f(\boldsymbol{A}_h \boldsymbol{A}_k \boldsymbol{A}_\ell) = e, \qquad f_-(\boldsymbol{A}_h \boldsymbol{A}_k \boldsymbol{A}_\ell) = h^{-1}k^{-1}hk \neq e.$$

Thus this reachable product differs from $\boldsymbol{I}_4$, although both decode to the identity. □

C.5 A SPECTRAL OBSTRUCTION UNDER STATES SEPARABILITY FOR S_5

The previous construction tracks A_5, the 60 even permutations of five objects. We now show that the same spectral constraint cannot support S_5, which contains all 120 permutations. The model must handle every permutation and their compositions, even when several hidden states represent the same output.

We prove Theorem 4 for arbitrary state dimensions, additive terms, and joint decoders. The planar-rotation lemma used below is proved immediately afterward.

Proof of Theorem 4. Suppose a tracker exists. By Theorem 22, we may replace it by a tracker with homogeneous orthogonal transitions, without increasing the number of non-real unit-circle pairs in any head. The resulting block-diagonal matrices generate a finite group Γ, with a surjective homomorphism $p : \Gamma \to S_5$ satisfying $p(\boldsymbol{A}_g) = g$. This reduction applies to one head as well as to several.

Construct the rotations in each head. Let $c = (12345)$ and $t = (12)$. Since $p(\boldsymbol{A}_c) = c$, the order of $\boldsymbol{A}_c$ is divisible by five; write it as $5^a r$, with $a \geq 1$ and 5 does not divide r. Choose $E > 0$ divisible by $2r$ and congruent to 1 modulo 5; such a choice exists because $2r$ is coprime to 5. Set

$$\boldsymbol{R}_1 = \boldsymbol{A}_c^E, \qquad \boldsymbol{R}_2 = \boldsymbol{A}_t \boldsymbol{R}_1 \boldsymbol{A}_t^\top.$$

Since r divides E but 5 does not, taking this power removes the part r of the order, leaving $\boldsymbol{R}_1$ with order $5^a r / \gcd(5^a r, E) = 5^a$. Each head has order dividing 5^a. The even exponent also turns all real eigenvalues ± 1 into 1. With at most one non-real conjugate pair per head, each head is therefore the identity or a planar rotation of order 5^j, $1 \leq j \leq a$. The corresponding head of $\boldsymbol{R}_2$ is the same rotation expressed in the orthonormal basis specified by the corresponding block of $\boldsymbol{A}_t$, so it has the same angle and order. Meanwhile,

$$p(\boldsymbol{R}_1) = c^E = c, \qquad p(\boldsymbol{R}_2) = tct^{-1} = (13452) =: b.$$

Compare the matrix product with its decoded permutation. Let $\boldsymbol{R}_{j,h}$ denote the block of $\boldsymbol{R}_j$ in head h, and define

$$\boldsymbol{C}_h = \boldsymbol{R}_{1,h} \boldsymbol{R}_{2,h} \boldsymbol{R}_{1,h}^\top \boldsymbol{R}_{2,h}^\top, \qquad \boldsymbol{C} = \boldsymbol{R}_1 \boldsymbol{R}_2 \boldsymbol{R}_1^\top \boldsymbol{R}_2^\top = \operatorname{diag}(\boldsymbol{C}_1, \ldots, \boldsymbol{C}_H).$$

Each pair of head rotations belongs to a finite group, as the image of Γ on that head. By Lemma 21, $\boldsymbol{C}_h^{10} = \boldsymbol{I}$. Hence $\boldsymbol{C}^{10} = \boldsymbol{I}$. But direct permutation composition gives $p(\boldsymbol{C}) = cbc^{-1}b^{-1} = (142)$, so

$$e = p(\boldsymbol{C}^{10}) = (142)^{10} = (142) \neq e,$$

a contradiction. □

Lemma 21 (A constraint on two planar rotations)**.** *Let $\boldsymbol{R}_1, \boldsymbol{R}_2 \in O(n)$ belong to a finite group. Suppose they are planar rotations through the same angle, possibly in different planes, and their common order is a multiple of five. Angles are measured in $[0, \pi]$, ignoring direction. Then*

$$(\boldsymbol{R}_1 \boldsymbol{R}_2 \boldsymbol{R}_1^\top \boldsymbol{R}_2^\top)^{10} = \boldsymbol{I}.$$

Proof. First prove a three-dimensional version. Let $\boldsymbol{X}, \boldsymbol{Y} \in SO(3)$ satisfy the lemma's assumptions, and write θ for their common angle. In particular, their traces are equal:

$$\operatorname{tr} \boldsymbol{X} = \operatorname{tr} \boldsymbol{Y} = 1 + 2\cos\theta.$$

We claim that $\boldsymbol{D} := \boldsymbol{X}\boldsymbol{Y}\boldsymbol{X}^\top\boldsymbol{Y}^\top$ satisfies $\boldsymbol{D}^5 = \boldsymbol{I}_3$. If their axes coincide, the rotations commute and $\boldsymbol{D} = \boldsymbol{I}_3$. If their axes differ, we use the finite rotation-group classification recalled before Theorem 17. In a cyclic or dihedral rotation group, all rotations of order greater than two share an axis. Tetrahedral and octahedral rotations have orders at most four. Therefore only the rotations of an icosahedron remain possible.

In this case, $\boldsymbol{X}, \boldsymbol{Y}$ rotate about distinct axes through opposite vertices, and θ is $2\pi/5$ or $4\pi/5$. We now choose these axes explicitly. Let $\varphi = (1 + \sqrt{5})/2$. Place the icosahedron at the origin with vertices

$$(0, \pm 1, \pm\varphi), \qquad (\pm 1, \pm\varphi, 0), \qquad (\pm\varphi, 0, \pm 1),$$

where the signs are independent. Each pair of opposite vertices defines one of the six vertex axes. By symmetry, we may take the axis of $\boldsymbol{X}$ through $(0, 1, \varphi)$. Rotation by $2\pi/5$ about this axis cycles the other five vertex axes, so we may take the axis of $\boldsymbol{Y}$ through $(1, \varphi, 0)$. Therefore we can choose unit axis vectors

$$\boldsymbol{u} = \pm\frac{(0, 1, \varphi)^\top}{\sqrt{1 + \varphi^2}}, \qquad \boldsymbol{v} = \pm\frac{(1, \varphi, 0)^\top}{\sqrt{1 + \varphi^2}}.$$

Choose their signs so that $\boldsymbol{X}, \boldsymbol{Y}$ both rotate through θ with the right-hand rule. The signs do not affect the squared inner product:

$$(\boldsymbol{u}^\top \boldsymbol{v})^2 = \frac{\varphi^2}{(1 + \varphi^2)^2} = \frac{1}{5},$$

where the last equality follows from $\varphi^2 = \varphi + 1$ and $(1 + \varphi^2)^2 = 5\varphi^2$.

It remains to show that $\operatorname{tr} \boldsymbol{D} = 1 + 2\cos\theta$. Write $x = \cos\theta$. Rodrigues' formula expresses the two rotations as

$$\boldsymbol{X} = x\boldsymbol{I}_3 + (1 - x)\boldsymbol{u}\boldsymbol{u}^\top + \sin\theta[\boldsymbol{u}]_\times, \qquad \boldsymbol{Y} = x\boldsymbol{I}_3 + (1 - x)\boldsymbol{v}\boldsymbol{v}^\top + \sin\theta[\boldsymbol{v}]_\times,$$

where $[\boldsymbol{w}]_\times \boldsymbol{z} = \boldsymbol{w} \times \boldsymbol{z}$. Substituting into $\boldsymbol{D} = \boldsymbol{X}\boldsymbol{Y}\boldsymbol{X}^\top\boldsymbol{Y}^\top$ and taking the trace gives

$$\operatorname{tr} \boldsymbol{D} = \left[2 - (1 - x)^2\left(1 - (\boldsymbol{u}^\top \boldsymbol{v})^2\right)\right]^2 - 1 = \left[2 - \tfrac{4}{5}(1 - x)^2\right]^2 - 1.$$

Thus the factor $4/5$ comes from the angle between the two vertex axes. It remains to substitute the rotation angle: since $\theta \in \{2\pi/5, 4\pi/5\}$, we have $x = (-1 \pm \sqrt{5})/4$. Both values satisfy $4x^2 + 2x - 1 = 0$, which implies $\frac{4}{5}(1 - x)^2 = 1 - 2x$. Therefore

$$\operatorname{tr} \boldsymbol{D} = (1 + 2x)^2 - 1 = 1 + 2x = \operatorname{tr} \boldsymbol{X} = \operatorname{tr} \boldsymbol{Y},$$

where the second equality again uses $4x^2 + 2x - 1 = 0$. For a three-dimensional rotation, the trace $1 + 2\cos\theta$ uniquely determines the angle in $[0, \pi]$. Hence $\boldsymbol{D}$ also rotates through $2\pi/5$ or $4\pi/5$, so $\boldsymbol{D}^5 = \boldsymbol{I}_3$ as claimed.

Apply this calculation to the original planar rotations. Their two rotating planes span a space U of dimension at most four. Both matrices preserve U and leave every vector in $U^\perp$ unchanged. We can therefore restrict to U and, if needed, add coordinates on which both matrices act as the identity. This gives matrices in $SO(4)$ with the same angles and orders, still generating a finite group.

The two homomorphisms of Lemma 18 preserve unsigned planar angles, so the images of $\boldsymbol{R}_1, \boldsymbol{R}_2$ satisfy the three-dimensional claim for either map. Writing $\boldsymbol{C} = \boldsymbol{R}_1 \boldsymbol{R}_2 \boldsymbol{R}_1^\top \boldsymbol{R}_2^\top$, we obtain

$$f(\boldsymbol{C}^5) = f_-(\boldsymbol{C}^5) = \boldsymbol{I}_3.$$

The lemma's common-kernel statement gives $\boldsymbol{C}^5 = \pm\boldsymbol{I}_4$; squaring yields $\boldsymbol{C}^{10} = \boldsymbol{I}_4$. Since the original matrices fix $U^\perp$, the same identity holds in the original dimension. □

Consequences for CKDA and Gated DeltaProduct. With unit keys, CKDA transitions with $\beta \in [0, 2]$ and $\alpha_i \in [-1, 1]$ are non-expansive, and Theorem 9 bounds their non-real unit-circle pairs by one. Gated DeltaProduct transitions with $\beta_j \in [0, 2]$ and scalar $\alpha_g \in [0, 1]$ are also non-expansive. Expanding the product of k Householder transformations gives

$$\operatorname{rank}(\boldsymbol{A}_g - \alpha_g \boldsymbol{I}) \leq k.$$

Since α_g is real, at most k eigenvalues can be non-real, counted with algebraic multiplicity. They occur in conjugate pairs, so $k \leq 3$ allows at most one such pair per head. Theorem 4 therefore rules out both architectures under boundedness and states separability of the joint reachable states, including input-dependent additive matrices and arbitrary joint decoders.

For sufficiency at $k = 4$, use the permutation-matrix construction of Siems et al. (2025, Thm. 1). Every permutation of five objects is a product of at most four transpositions. Each transposition (ij) has matrix $\boldsymbol{I}_5 - 2\boldsymbol{k}_{ij}\boldsymbol{k}_{ij}^\top$, where $\boldsymbol{k}_{ij} = (\boldsymbol{e}_i - \boldsymbol{e}_j)/\sqrt{2}$ is a unit key. Thus four Householder transformations suffice, padding shorter products with $\beta_j = 0$ and choosing $\beta_j = 2$ for the transpositions. Set the scalar gate to $\alpha = 1$, the additive term to zero at every step, and the initial state to $\boldsymbol{I}_5$. The state is then exactly the permutation matrix of the running group product, so a single head tracks S_5 with precisely 120 reachable states and orthogonal transitions. Any $k > 4$ is also sufficient by padding with identities. Together with the lower bound, this proves that four is the minimum number of Householder transformations needed for single-layer Gated DeltaProduct tracking of S_5 under these assumptions.

C.6 Finite replacement under states separability

In this section, we prove a reduction that extracts a finite tracker from any exact affine tracker of a finite group, provided its head transitions are non-expansive and its reachable states are bounded and satisfy states separability (Definition B.5). The new tracker uses orthogonal transitions with no additive terms, yet tracks the same group without increasing any head's dimension or number of non-real unit-circle eigenvalue pairs. This means that a spectral obstruction proved for finite orthogonal trackers also rules out this broader class of affine trackers, even when their reachable sets are infinite. It is this connection that allows the S_5 argument in Appendix C.5 to extend beyond finite reachability.

C.6.1 Compact matrix-group preliminaries

We use the basic group definitions from Appendix B.1 and introduce the matrix-group concepts needed for the reduction and its supporting finite-subgroup lemma.

Matrix groups. A *matrix group* is a group of invertible square matrices, with matrix multiplication as its operation. Here the main examples are

$$O(d) = \{Q \in \mathbb{R}^{d\times d} : Q^\top Q = I_d\}, \qquad SO(d) = \{Q \in O(d) : \det Q = 1\}.$$

These contain the orthogonal matrices and the orientation-preserving ones, respectively. Matrix *conjugation*, $A \mapsto UAU^{-1}$, is a change of basis and preserves eigenvalues with their multiplicities.

Closure, compactness, and Lie groups. The *closure* $\overline{E}$ is the set of all limits of convergent sequences in E. A matrix set is *compact* if it is closed and bounded. A *Lie group* is a group with the structure of a smooth manifold: locally, its elements have finitely many real coordinates, and multiplication and inversion are smooth. We only need closed subgroups $K \subseteq O(d)$. They are compact, and the closed subgroup theorem ensures that they are Lie groups.

Components and quotients. The *identity component* K° consists of matrices joined to I by a continuous path in K. It is a *normal subgroup*. Its cosets are exactly the connected components, so the quotient K/K° treats each component as one group element. This *component group* is finite: Lie-group components are open, and compactness permits only finitely many of them. A continuous homomorphism to a finite group sends K° to the identity, since a continuous path cannot change a discrete label.

Tori and finite order. A *circle group* describes a planar rotation by one angle, with angles added modulo a full turn. A *d-dimensional torus* is a product of d such circle groups: each element is specified by d independent angles, and multiplication adds corresponding angles. We measure angles in full turns, so $1/4$ means 90° ($\pi/2$ radians), and 0 and 1 describe the same rotation. This gives the notation $\mathbb{R}^d/\mathbb{Z}^d$: angle vectors describe the same element whenever they differ by integers in every coordinate. The case $d = 0$ is the trivial group. Every compact connected abelian Lie group has this torus structure.

Two facts follow directly from these coordinates. For any integer $n \geq 1$, every element t has an *n-th root*: dividing its angles by n gives an element s with $s^n = t$. Also, the *n-torsion subgroup* $T[n] = \{t \in T : t^n = I\}$ consists of angles chosen from $0, 1/n, \ldots, (n-1)/n$ in each coordinate, so it has n^d elements. These are exactly the elements whose order divides n; an element has *finite order* if some positive power is I.

Normalizers and maximal tori. For a subgroup $H \subseteq K$, its *normalizer* is

$$N_K(H) = \{u \in K : uHu^{-1} = H\}.$$

It consists of the elements u for which $h \mapsto uhu^{-1}$ maps H onto itself. Thus H is normal in $N_K(H)$. A *maximal torus* is a torus subgroup contained in no larger torus. All maximal tori lie in K° and are conjugate by elements of K°. For a maximal torus T, the standard maximal-torus theorem gives $N_K(T)^\circ = T$.

C.6.2 The reduction theorem and proof

Theorem 22 (Finite replacement under states separability)**.** *Let G be a finite group with identity e. Consider $H < \infty$ independent heads, updated on input $g \in G$ by*

$$T_g^{(h)}(\boldsymbol{S}^{(h)}) = \boldsymbol{A}_g^{(h)}\boldsymbol{S}^{(h)} + \boldsymbol{B}_g^{(h)}, \qquad \boldsymbol{S}^{(h)} \in \mathbb{R}^{n_h \times m_h}, \qquad \|\boldsymbol{A}_g^{(h)}\|_2 \leq 1.$$

Write $\boldsymbol{S} = (\boldsymbol{S}^{(1)}, \ldots, \boldsymbol{S}^{(H)})$ for the joint state and T_g for its joint update. Suppose the states reachable from $\boldsymbol{S}_0$ form a set $\mathcal{R}$ with a decoder $d : \mathcal{R} \to G$ satisfying $d(\boldsymbol{S}_0) = e$ and $d(T_g(\boldsymbol{S})) = g\, d(\boldsymbol{S})$. Assume this set is bounded and satisfies states separability: there are $M < \infty$ and $\varepsilon > 0$, independent of input length, such that

$$\|\boldsymbol{S}\|_F \leq M, \qquad d(\boldsymbol{S}) \neq d(\boldsymbol{S}') \implies \|\boldsymbol{S} - \boldsymbol{S}'\|_F \geq \varepsilon \quad (\boldsymbol{S}, \boldsymbol{S}' \in \mathcal{R}).$$

Here $\|\boldsymbol{S}\|_F^2 = \sum_h \|\boldsymbol{S}^{(h)}\|_F^2$ is the joint Frobenius norm.

Then there is another tracker of G with the same heads and columns, and at most as many rows in each head. Its updates have no additive terms: $\boldsymbol{X}^{(h)} \mapsto \widetilde{\boldsymbol{A}}_g^{(h)}\boldsymbol{X}^{(h)}$, where $\boldsymbol{X}^{(h)} \in \mathbb{R}^{r_h \times m_h}$ and $0 \leq r_h \leq n_h$. Each reduced transition is a square orthogonal matrix acting on every column of the reduced state. For a real square matrix X, define $\nu(X)$ to be the number of non-real conjugate eigenvalue pairs on the unit circle, counted with algebraic multiplicity. The replacement satisfies

$$\nu(\widetilde{\boldsymbol{A}}_g^{(h)}) \leq \nu(\boldsymbol{A}_g^{(h)}) \qquad \textit{for every } g, h. \tag{23}$$

Moreover, the block-diagonal matrices $\widetilde{\boldsymbol{A}}_g = \operatorname{diag}(\widetilde{\boldsymbol{A}}_g^{(1)}, \ldots, \widetilde{\boldsymbol{A}}_g^{(H)})$ generate a finite group Γ. There is a surjective homomorphism $p : \Gamma \to G$ with $p(\widetilde{\boldsymbol{A}}_g) = g$: matrix products decode to the corresponding group products. The new reachable set is finite and its state decoder may be many-to-one.

Proof. 1. Isolate the common orthogonal part. We seek a subspace on which every transition preserves lengths, and one input sequence that strictly contracts its orthogonal complement. Combine the head transitions into the block-diagonal matrix $\boldsymbol{A}_g = \operatorname{diag}(\boldsymbol{A}_g^{(1)}, \ldots, \boldsymbol{A}_g^{(H)})$, which acts separately on each head's row coordinates. An input sequence $g_1, \ldots, g_\ell$ has linear transition product $P = \boldsymbol{A}_{g_\ell} \cdots \boldsymbol{A}_{g_1}$; the empty sequence gives $P = I$. This product describes how differences between two states evolve under the same sequence, since their additive terms cancel. Choose a product P whose length-preserving subspace $W = \ker(I - P^\top P)$ has the smallest possible dimension. Appending another input sequence cannot create new length-preserving directions, because every transition is non-expansive. Minimality therefore implies that VP preserves lengths on exactly W, for every transition product V.

Taking $V = P$ shows $PW \subseteq W$; since P is an isometry on W, $PW = W$. Since P maps W onto itself and VP preserves lengths there, every V also preserves lengths on W. In particular, both $\boldsymbol{A}_g$ and $P\boldsymbol{A}_g$ do so. Hence $\boldsymbol{A}_g$ maps W into itself: $\boldsymbol{A}_g W \subseteq W$. For $w \in W$, equality of norms gives $\boldsymbol{A}_g^\top \boldsymbol{A}_g w = w$. Thus $\boldsymbol{A}_g$ maps vectors orthogonal to W to vectors orthogonal to $\boldsymbol{A}_g W = W$, so it also maps $W^\perp$ into itself. Let Q_g be the restriction of $\boldsymbol{A}_g$ to W; this restriction is orthogonal. On $W^\perp$, the definition of W gives $\|P|_{W^\perp}\|_2 < 1$ (vacuously if $W^\perp = 0$). Since P is block diagonal, W splits across heads. Restricting to these subspaces can only reduce each head's number of rows and leaves its number of columns unchanged.

2. Preserve labels and remove additive terms. Apply the same further inputs to two reachable states with different labels. Both labels are multiplied on the left by the same group element, so they remain different and their states stay at least ε apart. Let Π denote orthogonal projection onto W, applied to each column in its head. Repeating the input sequence for P, k times, therefore gives

$$d(\boldsymbol{S}) \neq d(\boldsymbol{S}') \implies \varepsilon \leq \|P^k(\boldsymbol{S} - \boldsymbol{S}')\|_F \xrightarrow[k\to\infty]{} \|\Pi(\boldsymbol{S} - \boldsymbol{S}')\|_F. \tag{24}$$

The limit holds because P preserves lengths on W and its powers tend to zero on $W^\perp$. Thus projection retains a well-defined decoder with the same separation gap.

The projected updates $X \mapsto Q_g X + \Pi \boldsymbol{B}_g$ are isometries with a bounded reachable set. Its smallest enclosing ball has a unique center c: two distinct minimizing centers would have a midpoint with a smaller enclosing radius. Each update maps the reachable set into itself and preserves its enclosing radius. The original ball therefore also minimally encloses the image set, whose center is the image of the original center. Uniqueness shows that every update fixes this center.

Since every update fixes c, we have $Q_g c + \Pi \boldsymbol{B}_g = c$ for every $g \in G$. We therefore measure each state relative to c: if the projected state is X, its centered state is $X - c$. After an update, the centered state becomes

$$(Q_g X + \Pi \boldsymbol{B}_g) - c = Q_g X + \Pi \boldsymbol{B}_g - (Q_g c + \Pi \boldsymbol{B}_g) = Q_g(X - c).$$

Thus the additive terms disappear in these coordinates. Translation preserves distances, and the decoder recovers the same labels by adding c back. The centered initial state is $X_0 = \Pi \boldsymbol{S}_0 - c$. In headwise orthonormal bases, $Q_g^{(h)}$ is a restriction of $\boldsymbol{A}_g^{(h)}$, so $\nu(Q_g^{(h)}) \leq \nu(\boldsymbol{A}_g^{(h)})$.

3. Extend the decoder to a compact group. Define

$$K := \overline{\{Q_{g_\ell} \cdots Q_{g_1} : \ell \in \mathbb{Z}_{\geq 0}, \quad g_1, \ldots, g_\ell \in G\}}.$$

The empty product ($\ell = 0$) is I, and the bar denotes closure in the matrix norm topology (Appendix C.6.1). Each Q_g is the joint block-diagonal transition, so every product uses the same input sequence in all heads. It is a compact matrix group: multiplication is continuous, and inverses of orthogonal matrices are limits of nonnegative powers. The set $KX_0 = \{UX_0 : U \in K\}$ consists of the centered reachable states and their limits. The closures of differently labelled state sets remain at distance at least ε, by equation 24, and cover KX_0. Thus the decoder extends uniquely and continuously to this set.

Define $p(U) = d(UX_0)$ using the decoder after projection, centering, and extension to the limits. On finite products, tracking gives $p(UV) = p(U)p(V)$ and $p(Q_g) = g$; continuity extends these identities to all $U, V \in K$. Thus p preserves products, and every label $g \in G$ occurs as $p(Q_g)$. Recall that the *identity component* K° consists of matrices connected to I by a continuous path. Continuity into the finite group G gives $p(K^\circ) = \{e\}$. The same limiting argument gives $d(UX) = p(U)d(X)$ for all $U \in K$ and $X \in KX_0$.

4. Replace each token by a finite-order element. We now replace each Q_g by a finite-order matrix with the same decoded label, leaving its action on each head's $+1$ and -1 eigenspaces unchanged. Any non-real eigenvalues of the replacement must therefore fit within the original rotation subspace, which bounds their number. Fix an input g. The group $C = \overline{\{Q_g^k : k \in \mathbb{Z}\}}$ is compact and abelian. By Appendix C.6.1, its identity component C° is a *torus* (a product of circles), and its *component group* C/C° is finite. Choose an integer $a \geq 1$ with $Q_g^a \in C^\circ$. Every torus element has an a-th root, so choose $V \in C^\circ$ with $V^a = Q_g^a$. The replacement $R_g = Q_g V^{-1}$ has finite order: commutativity gives $R_g^a = I$. It still decodes to g, because $C^\circ \subseteq K^\circ$ implies $p(V) = e$.

On the $+1$ eigenspace of $Q_g^{(h)}$, every power acts as the identity. On its -1 eigenspace, the k-th power acts as $(-1)^k I$, so only I and $-I$ occur. Taking limits introduces no other actions on these spaces. Now join any matrix in C° to I by a continuous path within C. Its restriction to the -1 eigenspace cannot jump between I and $-I$, so it remains I throughout the path. Thus every matrix in C°, including V, leaves every vector in both eigenspaces unchanged. Consequently, $R_g^{(h)} = (Q_g V^{-1})^{(h)}$ agrees with $Q_g^{(h)}$ on these spaces. Any non-real pairs of $R_g^{(h)}$ must lie in the remaining $2\nu(Q_g^{(h)})$ dimensions:

$$\nu(R_g^{(h)}) \le \nu(Q_g^{(h)}) \le \nu(\boldsymbol{A}_g^{(h)}).$$

5. Make the whole replacement finite. Each R_g now has finite order, but together these matrices may still generate an infinite group. We must place all of them in a single finite group while preserving their decoded labels and spectra. By Lemma 23, proved in Appendix C.6.3, we can choose $U_g \in K^\circ$ so that the matrices

$$\widetilde{\boldsymbol{A}}_g = U_g R_g U_g^\top$$

generate a finite group $\Gamma \subseteq K$. Each input may use a different U_g. Within each head, this replacement is a change of orthonormal basis, so it preserves that head's eigenvalues and equation 23. Also $p(U_g) = e$, giving

$$p(\widetilde{\boldsymbol{A}}_g) = p(U_g)p(R_g)p(U_g)^{-1} = g.$$

The finite state set $\Gamma X_0 = \{UX_0 : U \in \Gamma\} \subseteq KX_0$, with the same decoder d, is the required tracker. □

The final transitions may have different eigenvalues from the original ones and need not remain in the original architectural family. The theorem preserves the bound on ν separately in each head; the full matrix's pair count is their sum. This is sufficient to transfer the following spectral obstruction.

Why the reduction treats the heads jointly. Only the joint state is assumed to decode G and satisfy states separability. For example, two heads can each store one component of $G = \mathbb{Z}_2 \times \mathbb{Z}_2$, while neither determines the full group element. The linear-algebra steps preserve each head separately, but the group K retains the relationships imposed by a common input word. It may be smaller than the group obtained by choosing each head's matrix independently. The decoder p, which preserves products, is defined on K; choosing finite replacements independently in the heads could leave K and lose this decoding guarantee.

C.6.3 A COMMON FINITE SUBGROUP

The following lemma supplies the simultaneous finite-group replacement in Step 5 of Theorem 22. Its proof uses the normalizers, tori, and identity component defined in Appendix C.6.1.

Two classical results supply the lemma below. For every finite-order matrix R, some maximal torus T satisfies $RTR^{-1} = T$ (Borel & Serre, 1953, Theorem 1, finite cyclic case); the ambient compact Lie group need not be connected. Also, if a compact Lie group N has a torus T as its identity component, then $N = TF$ for some finite subgroup F (An et al., 2013, Lemma 3.5). Thus every matrix in N can be written as tf with $t \in T$ and $f \in F$; the factors need not be unique.

Lemma 23 (Placing finite-order matrices in a common finite group)**.** *Let $R_1, \dots, R_s$ be finite-order elements of a compact matrix group K. Write K° for its identity component: the matrices joined to I by a continuous path within K. For each i, one can choose $U_i \in K^\circ$ so that all matrices $U_i R_i U_i^{-1}$ lie in one finite subgroup of K. If all matrices in K have the same block-diagonal structure, these replacements preserve the spectrum of every block.*

Proof. 1. Make every matrix preserve the same torus. Fix a maximal torus T of K. By the first classical result above, each R_i preserves some maximal torus. Choose $U \in K^\circ$ that takes this torus to T, and replace R_i by UR_iU^{-1}. Doing this for each i, and keeping the same notation, gives $R_iTR_i^{-1} = T$. All matrices now belong to the normalizer $N = N_K(T)$, whose identity component is T (Appendix C.6.1). The second classical result gives $N = TF$ with F finite. Thus each R_i is a product of a torus element and an element of F.

Write $\tau(x)$ for the torus element with angle vector $x \in \mathbb{R}^d$, using the full-turn coordinates of Appendix C.6.1. Thus $\tau(x) = I$ exactly when $x \in \mathbb{Z}^d$. Choose $n \geq 1$ divisible by the orders of all R_i and all elements of F. We will move every torus factor into the finite set $T[n] = \{t \in T : t^n = I\}$, whose angles are multiples of $1/n$.

2. Adjust one matrix at a time. Fix any one of the matrices and write $R = \tau(x)f$, with $f \in F$. The map $t \mapsto ftf^{-1}$ preserves T and is linear in angle coordinates: for some matrix M,

$$f\tau(v)f^{-1} = \tau(Mv) \qquad (v \in \mathbb{R}^d).$$

Since $f^n = I$, we have $M^n = I$. Using also $R^n = I$ gives

$$I = R^n = \tau\left(\sum_{j=0}^{n-1} M^j x\right), \qquad \bar{x} := \frac{1}{n}\sum_{j=0}^{n-1} M^j x \in \frac{1}{n}\mathbb{Z}^d.$$

So the average angle vector $\bar{x}$ represents an element of $T[n]$. It remains to replace x by $\bar{x}$ using a change of basis in T. For any $y \in \mathbb{R}^d$,

$$\tau(y)^{-1}R\tau(y) = \tau\big(x - (I - M)y\big)f.$$

We therefore need $(I - M)y = x - \bar{x}$. Choose

$$y = -\frac{1}{n}\sum_{j=0}^{n-1} jM^j x.$$

Using $M^n = I$, consecutive terms cancel to give

$$(I - M)y = -\frac{1}{n}\left(\sum_{j=0}^{n-1} M^j x - nx\right) = x - \bar{x}.$$

Hence $\tau(y)^{-1}R\tau(y) = \tau(\bar{x})f \in T[n]F$. This works for each R_i, using the same n and F.

3. Verify that the common set is a finite group. The set $T[n]F = \{tf : t \in T[n],\ f \in F\}$ is finite. For $f \in F$ and $t \in T[n]$, the matrix ftf^{-1} belongs to T and satisfies $(ftf^{-1})^n = ft^n f^{-1} = I$, so it also belongs to $T[n]$. Consequently,

$$(tf)(t'f') = t(ft'f^{-1})(ff') \in T[n]F;$$

inverses belong to this set for the same reason. Thus $T[n]F$ is a finite subgroup containing every replacement.

Both changes of basis use matrices in K°, since $T \subseteq K^\circ$. Their products give the required U_i. If K has a fixed block-diagonal structure, each change of basis acts within the blocks and preserves each block's eigenvalues and multiplicities. □

Relation to earlier reductions. Separation-based reductions to finite automata have precedents in probabilistic automata (Rabin, 1963, Theorem 3), topological automata (Jeandel, 2007), and η-finite dynamical systems (Dankowiakowski & Ronca, 2025, Theorem 1). These results do not supply the per-head dimension and spectral guarantees needed here. Our reduction uses the classical compact-group facts above to retain these guarantees while making the tracker finite.

D THREE-LAYER SIMULATION OF FINITE AND WEIGHTED AUTOMATA

We prove Theorem 5 using the tasks and arithmetic conventions of Appendices B.2 and B.5.

D.1 THE CLOCK, BUFFER, AND ACCUMULATOR

The idea is to give the recurrence several steps to apply a complicated transition. Choose a block length m such that the combined transition of any m inputs can be factored into at most m permitted matrices. Following Siems et al. (2025, Thm. 3 and its proof), collect m consecutive inputs and factor their combined transition, padding with identities when necessary. Apply these factors one at a time while reading the next block:

- The *clock* records the position modulo $2m$, telling the later layers which buffer slot and matrix factor to use.
- The *buffer* retains the last $2m$ tokens. Its feed-forward lookup selects the factors for the previous block while retaining the inputs arriving in the current block.
- The *accumulator* applies the selected factor at each step, carrying forward the product from earlier blocks.

For example, with $m = 3$, inputs $4, 5, 6$ provide three steps to apply the three factors representing inputs $1, 2, 3$. The readout corrects this delay: it completes the unapplied factors and then applies the current block's prefix. The first block is handled directly by lookup.

The same organization applies to the column-state matrix products in Appendix B.2. The buffer and completion readout are those in Peng et al. (2025, Appendix D) and Siems et al. (2025, Thm. 3, Lems. 2–3), extended to rational WFAs by Merrill et al. (2026a, Thms. 6 and 11); we refer to these works for their constructions. The buffer uses DeltaNet updates, which are also CKDA updates with identity diagonal gate.

D.2 Three-layer expressivity

The clock and factorization results used below are proved in the next two subsections.

Proof of Theorem 5. Use the clock–buffer–accumulator construction of Merrill et al. (2026a, Thm. 11), summarized in Appendix D.1, replacing its two clock layers by Lemma 24. The remaining layers stream the permutation, DFA, or rational WFA factors from Proposition 25, with $m = \max(1, n-1)$, $m = \max(1, 2n-2)$, or $m = 7n^2 + 3n$, respectively. The exact finite-datatype and polynomial-storage guarantees follow from Appendix B.5. □

D.3 A one-layer clock

DeltaNet obtains the clock rotation from two alternating reflections, using a parity layer to select between them (Merrill et al., 2026a, Lem. 7). CKDA combines the two reflections within one transition: its signed diagonal supplies one of them.

Lemma 24 (One-step modular counter)**.** *For every integer $m \geq 1$ there is a* one*-layer CKDA head whose readout determines $i \bmod 2m$ at every position i.*

Proof. With $\beta = 2$, $\boldsymbol{\alpha} = (1, -1)$ and $\boldsymbol{k} = (-\sin\frac{\varphi}{2}, \cos\frac{\varphi}{2})^\top$ the transition $(\boldsymbol{I} - 2\boldsymbol{k}\boldsymbol{k}^\top)\operatorname{Diag}(\boldsymbol{\alpha})$ is a rotation by φ. Set $\varphi = \pi/m$ and $\boldsymbol{c}_0 = \boldsymbol{e}_1$. Then $\boldsymbol{c}_i = (\cos(\pi i/m), \sin(\pi i/m))^\top$ visits $2m$ distinct points. Choose a fixed unit query $\boldsymbol{q}$ such that $\boldsymbol{q}^\top(\boldsymbol{c}_i - \boldsymbol{c}_j) \neq 0$ for $0 \leq i < j < 2m$. Such a query exists because only finitely many lines are excluded. The single scalar readout $\boldsymbol{q}^\top \boldsymbol{c}_i$ then distinguishes all clock states, and a finite lookup returns the residue. □

This is similar to the DeltaProduct$_k$ $k \geq 2$ counter in Siems et al. (2025, Lem. 2(ii)). Its rotation parameters are algebraic; the fixed query can also be chosen algebraic by the argument following Lemma 11. The clock uses the fixed exact datatype of Appendix B.5.

D.4 Accumulator factorizations

The accumulator applies one matrix factor per recurrent step. We therefore bound the number of factors required for permutation, deterministic, and rational transitions. These bounds determine the block length $m \geq 1$ in Appendix D.1: any combined block transition must admit a factorization with at most m factors.

Proposition 25 (Factor-count upper bounds for the accumulator)**.** *Let $n \geq 1$ be the automaton-state dimension. A deterministic transition is an $n \times n$ matrix with one entry equal to 1 in each column and all other entries zero. The following numbers of recurrent matrix factors are sufficient; these*

construction-dependent bounds need not be optimal:

matrix family or primitive	*RWKV-7*	*DeltaNet/GDN*	*CKDA*
one transposition	1	1	1
arbitrary permutation	$n-1$	$n-1$	$n-1$
one DFA copy matrix	1	3	2
deterministic transition, explicit construction	n	$3n-3$	$2n-2$
arbitrary rational $n \times n$ matrix	$2n$	$(7n^2+5n+1)^\dagger$	$7n^2+3n$

The rational-matrix row permits scratch coordinates (auxiliary storage): RWKV-7 operates in dimension $2n$, and DeltaNet/GDN and CKDA in dimension $2n+1$. The dagger marks the DeltaNet/GDN budget with unrestricted unit-key learning rate $\beta \in \mathbb{R}$; the CKDA budget uses $\beta \geq 0$ and diagonal entries in $[-1,1]$. The explicit DeltaNet/GDN and CKDA deterministic constructions use $\beta \in [0,3]$; permutations use $\beta = 2$ reflections. A product with no factors is the identity; the swap and copy-matrix rows require $n \geq 2$.

Proof. Write $\boldsymbol{e}_j$ for the jth standard basis vector and $\boldsymbol{I}$ for the identity in the current state dimension. For a scalar γ and vector $\boldsymbol{u}$, write $\mathcal{H}_\gamma(\boldsymbol{u}) = \boldsymbol{I} - \gamma \boldsymbol{u}\boldsymbol{u}^\top$. A CKDA factor $\mathcal{H}_\gamma(\boldsymbol{u})\boldsymbol{D}$ applies a diagonal gate $\boldsymbol{D}$ first and a generalized Householder transformation second. For $\boldsymbol{u} \neq 0$, its unit key is $\boldsymbol{k} = \boldsymbol{u}/\|\boldsymbol{u}\|_2$ and its learning rate is $\beta = \gamma\|\boldsymbol{u}\|_2^2$. Rational transition entries do not imply rational normalized keys.

Transpositions and permutations. A transposition (a swap of coordinates $i \neq j$) is $\mathcal{H}_1(\boldsymbol{e}_i - \boldsymbol{e}_j)$, so a permutation costs at most $n-1$ factors (Siems et al., 2025, Thm. 3, proof). RWKV-7 also implements each swap in one factor (Peng et al., 2025, Lem. 4). For CKDA, this worst-case count is sharp by the n-cycle example in Proposition 10.

Deterministic transitions: explicit construction. For distinct coordinates s, d, a *copy matrix* is the identity with column d replaced by column s, following Peng et al. (2025, Lem. 3). Right-multiplying a matrix by it copies column s into column d. Acting on a column state $\boldsymbol{x}$, it instead sends (x_s, x_d) to $(x_s + x_d, 0)$, leaving other coordinates unchanged; thus it implements the DFA transition from state d to state s. Set $\boldsymbol{D}_d = \boldsymbol{I} - \boldsymbol{e}_d\boldsymbol{e}_d^\top$, which clears coordinate d. The copy matrix has the factorization

$$\boldsymbol{M}_{d\to s} := \boldsymbol{I} - \boldsymbol{e}_d\boldsymbol{e}_d^\top + \boldsymbol{e}_s\boldsymbol{e}_d^\top = \big[\mathcal{H}_3(\boldsymbol{e}_s)\boldsymbol{D}_d\big]\mathcal{H}_{1/6}(3\boldsymbol{e}_s + \boldsymbol{e}_d).$$

In the (s,d) coordinates, this identity is simply

$$\begin{pmatrix} -2 & 0 \\ 0 & 0 \end{pmatrix}\begin{pmatrix} -1/2 & -1/2 \\ -1/2 & 5/6 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix}.$$

The effective learning rates are 3 and $5/3$, so each copy matrix uses two CKDA factors. Separating $\boldsymbol{D}_d = \mathcal{H}_1(\boldsymbol{e}_d)$ gives three DeltaNet/GDN factors, with learning rates $3, 1, 5/3$. RWKV-7 uses one factor per identity, swap, or copy matrix and at most n such primitives per deterministic transition (Peng et al., 2025, Lems. 3–4), yielding the RWKV-7 bound.

For DeltaNet/GDN and CKDA, use one copy matrix per noncycle vertex and $\ell - 1$ swaps per cycle of length ℓ. To see this, let $f : \{1,\dots,n\} \to \{1,\dots,n\}$ be the deterministic state map of the underlying deterministic automaton, so $\boldsymbol{M}\boldsymbol{e}_i = \boldsymbol{e}_{f(i)}$. In the graph with arrows $i \to f(i)$, every component is a cycle with trees feeding into it. Let k count the vertices on cycles and c count the cycles, including fixed points as cycles of length one, which require no swaps. Right-multiplying by $\boldsymbol{M}_{d\to s}$ replaces column d by column s. Thus, starting from $\boldsymbol{I}$, right-multiply by $\boldsymbol{M}_{i\to f(i)}$ for each noncycle vertex i, processing vertices in decreasing distance from their cycle. This replaces column i by column $f(i)$, which is still the original $\boldsymbol{e}_{f(i)}$. The remaining cycle columns are then arranged by swaps within each cycle, leaving completed noncycle columns unchanged. The column-state recurrence applies the resulting factors from right to left. There are $n-k$ copy matrices and $k-c$ swaps. A copy matrix costs three DeltaNet/GDN factors or two CKDA factors, while a swap costs one, giving

$$\begin{aligned} \text{DeltaNet/GDN:} \quad & 3(n-k) + (k-c) = 3n - 2k - c \leq 3n - 3, \\ \text{CKDA:} \quad & 2(n-k) + (k-c) = 2n - k - c \leq 2n - 2, \end{aligned}$$

Here $k \geq c \geq 1$: following arrows in a finite graph must eventually revisit a vertex, so at least one cycle exists, with at least one vertex. For $n = 1$, the transition is the identity and the factorization is empty; when a recurrent update requires a factor, use the identity ($\beta = 0$ and unit gates).

Arbitrary rational matrices. Let $\boldsymbol{M} \in \mathbb{Q}^{n\times n}$ be the target matrix and $\boldsymbol{x} \in \mathbb{R}^n$ the input state. The RWKV-7 entry follows from the $2n$-factor construction in Merrill et al. (2026a, Lems. 5–6). In their row-state convention, the factor product is $\left(\begin{smallmatrix} \boldsymbol{M} & \boldsymbol{M} \\ 0 & 0 \end{smallmatrix}\right)$. Applying these same factors in reverse execution order to a column state maps $(\boldsymbol{x}, 0)$ to $(\boldsymbol{M}\boldsymbol{x}, 0)$, so scratch coordinates remain zero at block boundaries.

For DeltaNet/GDN and CKDA, refine the $8n^2 + 5n + 1$-factor program of Merrill et al. (2026a, Lems. 11–12), using n main coordinates $\boldsymbol{x}$, n scratch coordinates $\boldsymbol{s}$, and one temporary coordinate t. First clear $\boldsymbol{s}$ and t. A *unit coordinate addition* (transvection) adds coordinate s to coordinate $d \neq s$ of this augmented state, leaving all others unchanged. It uses three factors:

$$\boldsymbol{I} + \boldsymbol{e}_d\boldsymbol{e}_s^\top = \mathcal{H}_{1/3}(\boldsymbol{e}_s + 2\boldsymbol{e}_d)\mathcal{H}_{1/2}(\boldsymbol{e}_s)\mathcal{H}_2(\boldsymbol{e}_s + \boldsymbol{e}_d).$$

This is the column-state transpose of Merrill et al. (2026a, Lem. 10). Each scaled addition $s_i \leftarrow s_i + M_{ij}x_j$, for $1 \leq i, j \leq n$, uses $t \leftarrow t + x_j$, $t \leftarrow M_{ij}t$, $s_i \leftarrow s_i + t$, then $t \leftarrow 0$, where M_{ij} is the (i, j) entry of $\boldsymbol{M}$. For DeltaNet/GDN, combine the last addition and clear into the three-factor copy matrix $\boldsymbol{M}_{t\to s_i}$ above. Each scaled addition then costs $3 + 1 + 3 = 7$ factors, improving the eight-factor primitive of Merrill et al. (2026a, Lem. 11). Keeping their initial $n + 1$ clears, n main-coordinate clears, and $3n$ factors to copy scratch back gives

$$(n + 1) + 7n^2 + n + 3n = 7n^2 + 5n + 1.$$

This retains their $2n + 1$ coordinates and unrestricted learning rates. For CKDA, instead keep the three-factor final addition and absorb each temporary clear into the diagonal gate of the next Householder factor, also giving seven factors per scaled addition.

For a rational scaling coefficient $a > 0$, define the coordinate sign flip $\boldsymbol{S}_j = \boldsymbol{I} - 2\boldsymbol{e}_j\boldsymbol{e}_j^\top$ and use $\mathcal{H}_{1+a}(\boldsymbol{e}_j)\boldsymbol{S}_j$ to scale coordinate j by a. For $a < 0$, use $\mathcal{H}_{1+|a|}(\boldsymbol{e}_j)$, and for $a = 0$ use $\mathcal{H}_1(\boldsymbol{e}_j)$. Each is one admissible CKDA factor with nonnegative learning rate. The cited GDN scaling $\mathcal{H}_{1-a}(\boldsymbol{e}_j)$ instead needs a negative learning rate for $a > 1$ (Merrill et al., 2026a, Lem. 9).

The initial clear of scratch and temporary coordinates is absorbed into the first Householder factor. At the end, the final temporary clear and all main-coordinate clears are absorbed into the first of n three-factor unit additions that copy scratch back to main. The total is therefore

$$\underbrace{n^2(3 + 1 + 3)}_{\text{scaled additions; clears absorbed}} + \underbrace{3n}_{\text{copy back}} = 7n^2 + 3n.$$

The program maps $(\boldsymbol{x}, \boldsymbol{s}, t)$ to $(\boldsymbol{M}\boldsymbol{x}, \boldsymbol{M}\boldsymbol{x}, 0)$ for arbitrary initial scratch and temporary contents. □

The DFA simulation in Appendix D.2 uses this explicit CKDA construction and the fixed exact datatype of Appendix B.5.

D.5 Comparison with other transition families

Table 1 uses the decoder and arithmetic conventions of Appendices B.3 and B.5. Group inputs range over all elements, so tracking swaps alone does not establish tracking of arbitrary S_5 inputs. The reported depths are sufficient bounds, not claims of minimum depth. The S_n obstruction assumes non-expansive head transitions, bounded reachable states, and states separability for exact affine updates; for $n > 5$, it follows from Theorem 4 by restricting inputs to $S_5 \subset S_n$.

Spectral restrictions. Under the finite-precision convention of Grazzi et al. (2025, Thm. 2 and Appendix B.2), real transition spectra rule out one-layer counting modulo $q > 2$, and hence tracking any group containing an element of order greater than two. This applies to DeltaNet and scalar-gated GDN: $\boldsymbol{A} = \alpha(\boldsymbol{I} - \beta\boldsymbol{k}\boldsymbol{k}^\top)$ is symmetric, even for negative α. For RWKV-7 and GDN-2 under the table's nonnegative-gate assumptions, write $\boldsymbol{A} = \boldsymbol{D} - \boldsymbol{u}(\boldsymbol{v} \odot \boldsymbol{u})^\top$, $\boldsymbol{v} \geq 0$. If $\boldsymbol{v} > 0$, then $P = \mathrm{Diag}(\sqrt{\boldsymbol{v}})$ gives

$$P\boldsymbol{A}P^{-1} = \boldsymbol{D} - (P\boldsymbol{u})(P\boldsymbol{u})^\top,$$

which is symmetric; zero entries follow by continuity. Their WFA entries instead allow unrestricted gates. For DeltaProduct$_k$, the rank argument in Appendix C.5 gives at most $\lfloor \min(d, k)/2 \rfloor$ non-real pairs in dimension d.

RWKV-7 and GDN-2. The shared entries follow from transferring constructions between the two families under the parameter ranges in the table. Both contain DeltaNet when decay is one and the erase gate is constant, so they inherit the group constructions of Siems et al. (2025, Thms. 1 and 7). GDN-2 also implements the copy matrix: in Hatamizadeh et al. (2026, Eq. (10)), choose $\boldsymbol{D} = \boldsymbol{I}$, $\boldsymbol{k} = (\boldsymbol{e}_i - \boldsymbol{e}_j)/\sqrt{2}$, and erase gate $2\boldsymbol{e}_j$, obtaining $\boldsymbol{I} + (\boldsymbol{e}_i - \boldsymbol{e}_j)\boldsymbol{e}_j^\top$. This supplies the identity, swap, and copy matrices needed by the four-layer DFA construction of Peng et al. (2025, Thm. 3). The fixed-datatype entries use RWKV-7's $c = 2$ construction; its $c = 1$ adaptation uses normalization and growing exponent storage (Peng et al., 2025, Appendix D.3). The WFA entries use Merrill et al. (2026a, Thms. 6 and 11); GDN-2 inherits the DeltaNet construction through an unrestricted constant erase gate.

DeltaProduct and automata bounds. The one-layer DeltaProduct$_k$ clock for $k \geq 2$ (Siems et al., 2025, Lem. 2(ii)) replaces the two clock layers in Merrill's four-layer DeltaNet simulation (Merrill et al., 2026a, Thm. 11), giving three layers; for $k = 1$, the four-layer construction applies. With enough factors, a single DeltaProduct layer can instead implement each full automaton transition. Our copy-matrix factorizations in Proposition 25 give the thresholds $k \geq 3q - 3$ for a q-state DFA with $\beta \in [0, 3]$, and $k \geq B_q = 7q^2 + 5q + 1$ for rational WFAs with $2q + 1$ coordinates and unrestricted learning rates. At least one factor is used, padding with identities when needed. If learning rates stay in $[0, 2]$, Siems et al. (2025, Thm. 2) instead give a finite depth depending on the language, with the scalar gate supplying resets.

E Efficient Implementation of the Signed Gate

KDA stores decay magnitudes in log-space. To support signed gates, we absorb cumulative signs into the keys and queries, allowing the reuse of existing kernels in flash-linear-attention (Yang & Zhang, 2024). Fusing these sign flips into key and query normalization then avoids separate transformation passes.

Kimi K3's safe sigmoid (Kimi Team et al., 2026) uses $g = -5\sigma(a)$ and $\alpha = \exp(g)$ for the scaled, biased gate preactivation a, ensuring $\alpha \geq \epsilon := e^{-5}$ (indices suppressed). Our signed modification sets $r = \tanh(a/2) = 2\sigma(a) - 1$ and $\alpha = s[\epsilon + (1 - \epsilon)\,|r|]$, with $s = +1$ for $r \geq 0$ and $s = -1$ otherwise, and computes $g = \log|\alpha| \in [-5, 0]$ in FP32 without additional clamping. At $a = 0$, $\alpha = \epsilon$ and the signed gate is discontinuous; backpropagation detaches s and uses PyTorch's zero subgradient for $|r|$ at zero.

Write $\mathrm{Diag}(\boldsymbol{\alpha}_i) = \boldsymbol{S}_i\boldsymbol{D}_i$, where $\boldsymbol{S}_i = \mathrm{Diag}(\boldsymbol{\sigma}_i)$ with $\boldsymbol{\sigma}_i \in \{\pm 1\}^n$ and $\boldsymbol{D}_i = \mathrm{Diag}(|\boldsymbol{\alpha}_i|)$; the sign of a zero gate entry can be chosen arbitrarily. To cancel the signs accumulated by the state, let $\boldsymbol{P}_i := \boldsymbol{S}_1 \cdots \boldsymbol{S}_i$ with $\boldsymbol{P}_0 = \boldsymbol{I}$. These diagonal sign matrices satisfy $\boldsymbol{P}_i^\top = \boldsymbol{P}_i = \boldsymbol{P}_i^{-1}$.

Lemma 26 (Sign absorption)**.** *Let $\boldsymbol{H}_i = (\boldsymbol{I} - \beta_i\boldsymbol{k}_i\boldsymbol{k}_i^\top)\boldsymbol{S}_i\boldsymbol{D}_i\boldsymbol{H}_{i-1} + \beta_i\boldsymbol{k}_i\boldsymbol{v}_i^\top$ with readout $\boldsymbol{o}_i = \boldsymbol{H}_i^\top\boldsymbol{q}_i$. In the transformed coordinates $\tilde{\boldsymbol{H}}_i := \boldsymbol{P}_i\boldsymbol{H}_i$, $\tilde{\boldsymbol{k}}_i := \boldsymbol{P}_i\boldsymbol{k}_i$ and $\tilde{\boldsymbol{q}}_i := \boldsymbol{P}_i\boldsymbol{q}_i$, the same computation becomes*

$$\tilde{\boldsymbol{H}}_i = (\boldsymbol{I} - \beta_i\tilde{\boldsymbol{k}}_i\tilde{\boldsymbol{k}}_i^\top)\boldsymbol{D}_i\tilde{\boldsymbol{H}}_{i-1} + \beta_i\tilde{\boldsymbol{k}}_i\boldsymbol{v}_i^\top,$$
$$\boldsymbol{o}_i = \tilde{\boldsymbol{H}}_i^\top\tilde{\boldsymbol{q}}_i.$$

Moreover, $\tilde{\boldsymbol{H}}_0 = \boldsymbol{H}_0$, $\|\tilde{\boldsymbol{k}}_i\| = \|\boldsymbol{k}_i\|$, and $\boldsymbol{H}_i = \boldsymbol{P}_i\tilde{\boldsymbol{H}}_i$.

Proof. Left-multiply the recurrence by $\boldsymbol{P}_i$ and substitute $\boldsymbol{H}_{i-1} = \boldsymbol{P}_{i-1}\tilde{\boldsymbol{H}}_{i-1}$. Since diagonal matrices commute, $\boldsymbol{S}_i\boldsymbol{D}_i\boldsymbol{P}_{i-1} = \boldsymbol{P}_i\boldsymbol{D}_i$, so the transition becomes

$$\boldsymbol{P}_i(\boldsymbol{I} - \beta_i\boldsymbol{k}_i\boldsymbol{k}_i^\top)\boldsymbol{P}_i\boldsymbol{D}_i = (\boldsymbol{I} - \beta_i(\boldsymbol{P}_i\boldsymbol{k}_i)(\boldsymbol{P}_i\boldsymbol{k}_i)^\top)\boldsymbol{D}_i = (\boldsymbol{I} - \beta_i\tilde{\boldsymbol{k}}_i\tilde{\boldsymbol{k}}_i^\top)\boldsymbol{D}_i.$$

The additive term becomes $\beta_i\boldsymbol{P}_i\boldsymbol{k}_i\boldsymbol{v}_i^\top = \beta_i\tilde{\boldsymbol{k}}_i\boldsymbol{v}_i^\top$, and the readout satisfies $\boldsymbol{H}_i^\top\boldsymbol{q}_i = \tilde{\boldsymbol{H}}_i^\top\boldsymbol{P}_i\boldsymbol{q}_i = \tilde{\boldsymbol{H}}_i^\top\tilde{\boldsymbol{q}}_i$. Orthogonality gives the norm identity and the inverse transformation; $\boldsymbol{P}_0 = \boldsymbol{I}$ gives the initial condition. □

Relation to rotary embeddings. Absorbing cumulative transformations into keys and queries is familiar from RoPE and RetNet (Su et al., 2024; Sun et al., 2023), and from input-dependent rotary formulations of SSMs and Gated DeltaNet (Lahoti et al., 2026; Movahedi et al., 2026). Here, diagonal signs commute with arbitrary channel-wise decay, preserving all n independent decay magnitudes. The transformation introduces no parameters or additional recurrent state.

Three implementations of the sign transformation. The implementation labeled *gauge* uses compiled PyTorch operations, with no kernel changes. It computes cumulative signs and materializes $\tilde{\boldsymbol{q}}_i = \boldsymbol{P}_i \boldsymbol{q}_i$ and $\tilde{\boldsymbol{k}}_i = \boldsymbol{P}_i \boldsymbol{k}_i$ using `torch.compile`, then calls existing KDA kernels on $|\boldsymbol{\alpha}_i|$. It requires no kernel modification; the benchmark uses a TileLang-backed recurrence for this path. The Triton implementation computes signs by an integer parity scan and jointly normalizes and transforms keys and queries. This fusion uses $(\boldsymbol{P}_i \boldsymbol{x})/\|\boldsymbol{P}_i \boldsymbol{x}\| = \boldsymbol{P}_i(\boldsymbol{x}/\|\boldsymbol{x}\|)$; backward normalization and sign application are fused into the final query/key gradient writes. The TileLang-backed hybrid combines these Triton components with TileLang kernels for the WY backward computation; it is not an all-TileLang implementation. The algebraic recurrence is identical in all three cases. The final state is recovered as $\boldsymbol{H}_i = \boldsymbol{P}_i \tilde{\boldsymbol{H}}_i$, and its incoming gradient is transformed by the same signs. Our models use equal key and value head counts; grouped value attention with value-head-specific signs requires expanding keys and queries.

Recurrence-kernel benchmark. Figure 4 measures forward and backward on an H100 with BF16 inputs, 16 heads, and $d_k = d_v = 128$, holding the number of logical tokens per step at $32{,}768$. Projections, optimizer updates, and final-state output are excluded. Each implementation is wrapped in `torch.compile`; kernel compilation and autotuning precede 20 warm-up iterations per shape and 50 timed iterations per round. Bars report mean throughput over four rounds. The normal and signed kernel paths both use normalization fusion, and DeltaProduct$_2$ uses its tuned configuration without a forget gate at the same head dimensions; its two updates per token are not counted as extra logical tokens. The timed recurrence processes two separately projected value inputs per token, one for each delta-rule update, so the throughput includes the recurrence work associated with this additional value input relative to CKDA. Computing the projections is outside the timed region. Triton and hybrid measurements use separate H100 allocations with the same default dot-precision policy, without explicitly setting `TRITON_F32_DEFAULT`. The PyTorch-gauge series comes from an earlier allocation without normalization fusion and is therefore an implementation reference rather than a fully matched fusion ablation. Note that we do not benchmark FlashKDA (Chen et al., 2026) because it contains only inference kernels, while we benchmark the forward and backward pass.

F How the key activation affects learning group-word problems

The key path of DeltaNet (Yang et al., 2024b), inherited by KDA, is $\boldsymbol{x} \to \boldsymbol{W}_k \to \mathrm{SiLU} \to \ell_2$. Since $\mathrm{SiLU} \geq -0.2785$, normalization turns that floor into an angular prior: a unit key with a negative component of size c requires $\|\mathrm{SiLU}(\boldsymbol{z})\| \leq 0.2785/c$. Every unit-key direction remains reachable, so this is a possible optimization bias rather than a representational restriction. At $\beta = 2$, S_3 needs no negative component at all: $\boldsymbol{k} \in \{(\frac{\sqrt{3}}{2}, 0, \frac{1}{2}), (\frac{1}{2}, 0, \frac{\sqrt{3}}{2}), \boldsymbol{e}_3\}$ realizes all five non-identity elements, the two of order 3 reusing the first two keys. The cube realization of S_4 in Appendix C.3 uses mixed-sign keys; this does not establish that every higher-dimensional realization or decoded tracker must do so.

Dubinin et al. (2026) report that BD-LRU solves S_4 with much higher sample efficiency compared to DeltaProduct. But DeltaProduct shares the same key path as DeltaNet, while BD-LRU's entrywise gating has none to constrain. We hypothesize that the SiLU key activation contributes to optimization differences. The ablation in Figure 17 is group-dependent: removing SiLU improves the displayed A_5 result, whereas the displayed S_4 result is stronger with SiLU. It therefore does not establish SiLU as the cause of the cited S_4 gap.

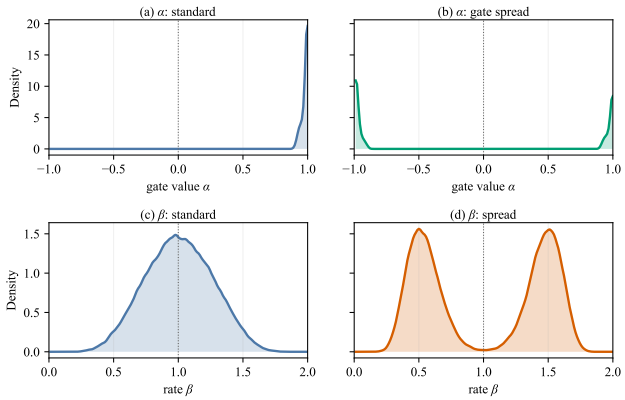

Figure 13: Standard and spread initializations for the signed extended-range layer.

G Experimental Details

G.1 Initialization

The gate starts from a log-uniform decay parameter d_t and applies the corresponding inverse parametrization, so that the initial magnitude is $|\alpha| = \exp(-d_t)$. For a signed gate, standard initialization uses $\alpha = +\exp(-d_t)$, whereas gate spread independently flips the sign of approximately half of the channels, $\alpha = s\exp(-d_t)$ with $s \in \{-1, +1\}$. The rate uses $\beta = c\,\sigma(b)$, where σ is the sigmoid, b is the rate preactivation, $c = 1$ for the standard range, and $c = 2$ for the extended range. Beta spread rescales the projection and adds opposite biases to different heads, $\beta = 2\sigma(\gamma z \pm b_0)$, where z is the unshifted projection output, producing two modes near 0.5 and 1.5. The resulting empirical distributions are shown in Figure 13.

G.2 State-Tracking

Standard models use one layer with 12 heads of dimension 16, trained for 60k steps with batch size 1024 and Muon (Jordan et al., 2024) at learning rate 5×10^{-3}. We use the length curriculum $4, 6, 8, 16, 32$ and report the best of three seeds. The curriculum aids learning, consistent with Beck et al. (2024); Siems et al. (2026), and Muon improves it further. For a group G, scaled accuracy $(a - 1/|G|)/(1 - 1/|G|)$ maps chance to 0 and perfection to 1; lengths above 32 test extrapolation.

For the theory-initialized A_5 experiment, we use $A_5 \cong 2I/\pm 1$ to assign each element a unit-quaternion lift $q_g \in \mathbb{R}^4$ following Appendix C.4.2. Two heads are initialized with $k_g = q_g$, $\boldsymbol{\alpha} = (-1, 1, 1, 1)$, and $\beta \approx 2$, yielding $T_g \approx (I - 2q_g q_g^\top)\,\mathrm{diag}(-1, 1, 1, 1)$. One head suffices theoretically; two improve empirical robustness. The model has four heads of dimension 4, hidden dimension 32, and a standard MLP readout. All parameters, including initialized embeddings and KDA projections, remain trainable. We train for 3k steps on all 60 elements using next-state cross-entropy and lengths $4, 6, 7, 8, 10, 12, 16, 32$, without auxiliary representation loss, restricted-generator curriculum, or maxout readout.

The robustness limitation discussed in Section 6 does not preclude exact infinite-horizon tracking in exact arithmetic (Chung et al., 2026). Complementarily, Dankowiakowski & Ronca (2025) show that linear-recurrence SSMs cannot robustly recognize all star-free languages, including FLIP-FLOP, whereas nonlinear xLSTM can. State-dependent nonlinear RNNs can implement error-correcting dynamics unavailable to affine recurrences (Beck et al., 2024; Pöppel et al., 2025; Danieli et al., 2026; Mishra et al., 2026).

G.3 Periodic waveform continuation

Data and task. We synthesize a fixed two-bar groove at 124 BPM with 32 sixteenth-note frames, render stereo audio at 4096 Hz, convert it to mono, and resample each frame to 64 waveform values. The resulting periodic sequence $\mathbf{Y} \in \mathbb{R}^{32\times 64}$ is normalized independently along each waveform coordinate. Each training example is circularly shifted by a phase $s \sim \mathrm{Uniform}\{0, \ldots, 31\}$, with

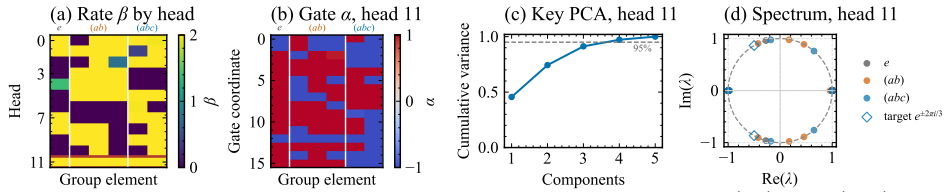

Figure 14: Learned CKDA transitions on the S_3 word problem. Here e, (ab), and (abc) denote the identity, transpositions, and 3-cycles, respectively. (a) Learned update strength β by head; head 11 approaches the reflection limit $\beta = 2$. (b) Its channel-wise gate becomes nearly sign-valued, with coordinates close to ± 1. (c) PCA of the keys: the first three principal components explain approximately 95% of the variance. (d) The resulting transition spectra show complex eigenvalues. The learned model recovers the mechanism predicted by our theory.

targets $\mathbf{y}_t = \mathbf{Y}_{(s+t) \bmod 32}$ and inputs

$$\mathbf{x}_t = \begin{cases} \mathbf{y}_t, & t < 8, \\ \mathbf{0}, & t \geq 8. \end{cases}$$

The model thus observes eight frames (one half-bar, approximately 0.97 seconds); inputs and targets have shape $B \times L \times 64$, and mean-squared error is computed only after the cue ($t \geq 8$). Using shifts of one fixed groove isolates phase inference and periodic extrapolation rather than general-purpose audio generation.

Architectures. All models use one sequence-mixing layer with hidden size 128, a bias-free $64 \to 127$ input projection followed by an appended constant coordinate, and a shared $128 \to 512 \to 64$ GELU readout. KDA models use eight heads with key and value dimensions 16, no short convolution, and no post-recurrent SiLU activation. We compare all four combinations of $\alpha \in [0, 1]$ or $[-1, 1]$ and $\beta \in [0, 1]$ or $[0, 2]$, using the corresponding spread initialization for extended ranges and default initialization for standard ranges. KDA models have approximately 182k parameters; the GRU baseline has one 128-dimensional recurrent layer and 206k parameters. The causal Transformer has one eight-head pre-norm self-attention block, causal masking, sinusoidal positional encodings, and a 128-dimensional feed-forward block, totaling 207k parameters.

Training. All models are trained from scratch for 2500 updates with batch size 64, seed 0, and the sequence-length curriculum $12, 16, 24, 40, 72, 136$. The curriculum occupies the first 40% of training, reaching length 136 at update 835 and retaining it thereafter. Muon optimizes two-dimensional matrices inside the sequence layer with learning rate 0.02, momentum 0.95, Nesterov momentum, and five Newton–Schulz iterations; AdamW optimizes the input projection, readout, biases, and other parameters with learning rate 0.003. Both learning rates use 250 warm-up updates followed by cosine decay to 10% of their initial values. Weight decay is 10^{-12}, and the global gradient norm is clipped to 1.

Evaluation. After undoing training normalization, we average error over all 32 cue phases, continuation steps, and waveform coordinates at sequence lengths $16, 24, 32, 40, 56, 72, 88, 104, 120, 136, 152, 168, 184, 200, 216, 232, 248, 264$. Waveform signal-to-noise ratio is $\mathrm{SNR} = 10 \log_{10}\big(\mathbb{E}[\mathbf{y}^2]/\mathbb{E}[(\hat{\mathbf{y}} - \mathbf{y})^2]\big)$, with both expectations taken over the continuation region. Qualitative waveforms in Figure 6 show phase 0 over steps 0–24 and the extrapolation window 232–264.

G.4 Language modeling.

Training. For the 1.3B parameter language modeling experiments, we follow the recipe of Yang et al. (2025); Hatamizadeh et al. (2026), training on 100B tokens of FineWeb-edu (Lozhkov et al., 2024), using the Llama-2 tokenizer (Touvron et al., 2023), a global batch size of $0.5M$ tokens, a linear warmup of $1B$ tokens to peak learning rate $4e-4$, followed by a cosine decay to 10% of the peak learning rate all optimized with the AdamW optimizer (Loshchilov & Hutter, 2019). For our CKDA variants we use the *standard* init with no SiLU on keys. Note that our hybrid variants

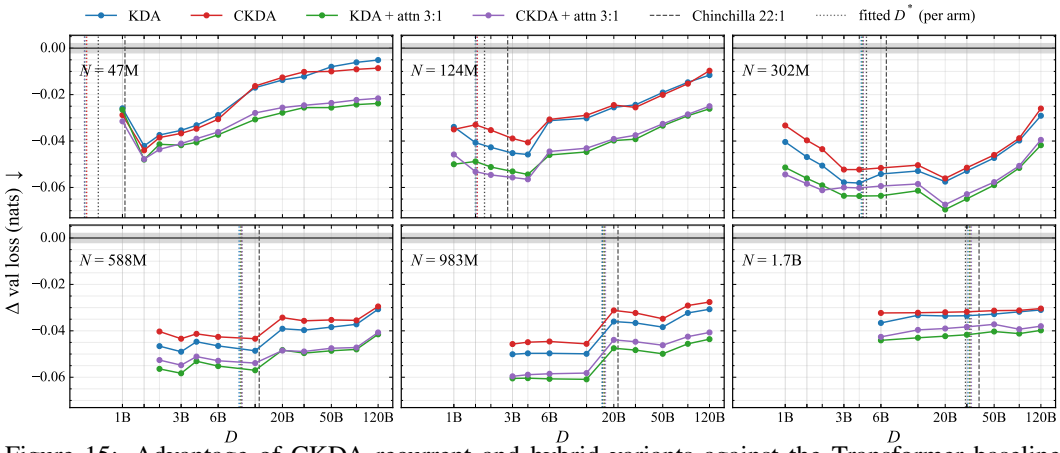

Figure 15: Advantage of CKDA recurrent and hybrid variants against the Transformer baseline (including QK-Norm, similar to Ajroldi et al. (2026)) in nats across scales. Grey area is the approximate noise floor. Compute optimal budgets are derived from fitted scaling laws, see Section H.1

are using full attention at a 3:1 recurrent-to-attention ratio compared to the 1:1 recurrent + sliding window comparison.

For additional ablations on the impact of $\boldsymbol{\alpha}$ and β ranges as well as the impact of SiLU activation for keys, we train on Nemotron-CC (Su et al., 2025) and a smaller version of FineWeb-edu. Also here, see Table 7, CKDA outperforms an attention baseline as well as DeltaProduct (Siems et al., 2025), GDN (Yang et al., 2025) and vanilla KDA (Kimi Team, 2025) on downstream evaluations. The small-model experiments use the GPT-2 tokenizer, with the model vocabulary padded to $65{,}536 = 2^{16}$ entries.

Table 4: Configurations used in the small-scale Nemotron-CC language modeling experiments. The intermediate size (the latent dimension of the SwiGLU MLP block) is reduced for GDN, DeltaProduct, and CKDA. We use AdamW with a global batch size of 64 and learning rate 5×10^{-4}, following a warmup–stable–decay (WSD) schedule: 2000 warmup steps and cooldown over the last 20% of the token budget. Weight decay is set to 0.1 and $(\beta_1, \beta_2) = (0.9, 0.95)$.

Config	**Dense Attention**	**GDN**	**DeltaProduct**	**KDA**	**CKDA**
Hidden size	768	768	768	768	768
Num. hidden layers	28	28	28	28	28
Num. attention heads	12	12	12	12	12
Intermediate size	3072	2816	2816	2816	2816
Hidden activation	SwiGLU	SwiGLU	SwiGLU	SwiGLU	SwiGLU
RoPE θ	10,000	N/A	N/A	N/A	N/A
Max. position embeddings	2048	2048	2048	2048	2048
Vocabulary size	65,536	65,536	65,536	65,536	65,536
Total parameters (millions)	341.55	342.33	342.07	344.87	344.87

Language-model evaluation. We use the Language Model Evaluation Harness (Gao et al., 2024). The evaluation tasks include WikiText (Merity et al., 2017), LAMBADA (OpenAI version) (Paperno et al., 2016), HellaSwag (Zellers et al., 2019), the Winograd Schema Challenge (Levesque et al., 2012), WinoGrande (Sakaguchi et al., 2021), COPA (Roemmele et al., 2011), PIQA (Bisk et al., 2020), ARC-Easy, ARC-Challenge (Clark et al., 2018), OpenBookQA (Mihaylov et al., 2018), SWDE (Lockard et al., 2019), FDA (Arora et al., 2023), and SQuADv2 (Rajpurkar et al., 2018). Table 9 expands the benchmark abbreviations. In Table 8 we also report RULER results. We use the standard methodology and prompts, but we found that slight variations of the prompt (e.g. adding a ":" to "The answer is") lead to largely different numbers and bad S-NIAH numbers can to a large degree be attributed to a lack of instruction following (no answer is given, the prompt is repeated). These should therefore be treated with caution.

Table 5: Architectures of the scaling ladder, with the 1.3B/100BT FineWeb-Edu replication beside it. Every arm is parameter-matched to its column's target: the intermediate size (the latent dimension of the SwiGLU MLP) absorbs the difference between each mixer's own parameter budget, so the MLP width differs between arms and the totals are very close. The ladder trains on Nemotron-CC under AdamW with $(\beta_1, \beta_2) = (0.9, 0.95)$ and weight decay 0.1; multiple global batch sizes / learning rates are possible for different total token budget. The FineWeb-Edu column follows the published Gated DeltaNet recipe instead, which is why its vocabulary, head dimension, tying and schedule differ.

Config	**47M**	**124M**	**302M**	**588M**	**983M**	**1.7B**	**1.3B (FineWeb-Edu)**
Hidden size	384	576	896	1280	1536	2048	2048
Num. hidden layers	12	18	20	20	24	24	24
Num. attention heads	6	9	14	20	24	32	16
Head dimension	64	64	64	64	64	64	128
Vocabulary size	50,304	50,304	50,304	50,304	50,304	50,304	32,000
Tied embeddings	yes	yes	yes	yes	yes	yes	no
Training corpus	Nemotron-CC	Nemotron-CC	Nemotron-CC	Nemotron-CC	Nemotron-CC	Nemotron-CC	FineWeb-Edu
LR schedule	WSD	WSD	WSD	WSD	WSD	WSD	cosine
Peak learning rate	0.002	0.001	0.001	0.0005 / 0.001	0.0005	0.0005	0.0004
Global batch size (sequences)	32 / 64	64	64 / 128	64 / 128 / 256	64 / 128 / 256 / 512	64 / 128 / 256 / 512	128
Warmup steps	2000	2000	2000	2000	2000	2000	1908
Cooldown fraction of budget	20%	20%	20%	20%	20%	20%	–
Intermediate size							
Transformer++	1536	2304	3584	5120	6144	8192	5632
KDA (safe sigmoid)	1472	2240	3520	4992	6016	8064	5440
CKDA	1472	2240	3520	4992	6016	8064	5440
KDA + attn 3:1	1408	2176	3456	4928	5952	7936	5312
CKDA + attn 3:1	1408	2176	3456	4928	5952	7936	5312
Total parameters (millions)							
Transformer++	47.64	124.55	302.01	588.73	983.31	1713.74	1364.30
KDA (safe sigmoid)	48.03	125.45	303.66	586.32	980.00	1709.71	1362.63
CKDA	48.03	125.45	303.66	586.32	980.00	1709.71	1362.63
KDA + attn 3:1	47.27	124.14	302.96	587.75	984.36	1712.29	1362.26
CKDA + attn 3:1	47.27	124.14	302.96	587.75	984.36	1712.29	1362.26
Constant across every column							
Hidden activation	SwiGLU						
Context length	4,096						

Table 6: Global batch size B (sequences of 4,096 tokens) and peak learning rate η at every cell of the scaling ladder, by model size and token budget. Every arm runs every cell at the same (B, η) – all runs: WSD schedule, linear warmup over $\min(2000, 9\%)$ of the steps, a linear cooldown over the final 20% to $\eta_{\min} = 10^{-5}$; AdamW with $\beta_2 = 0.95$, weight decay 0.1, gradient clipping at 1.

Model		1BT	1.5BT	2BT	3BT	4BT	6BT	12BT	20BT	30BT	50BT	80BT	120BT
47M	B	16	16	32	32	32	32	64	64	64	128	128	128
	η	10^{-3}	2×10^{-3}	2×10^{-3}	2×10^{-3}	2×10^{-3}	2×10^{-3}	2×10^{-3}	2×10^{-3}	2×10^{-3}	2×10^{-3}	2×10^{-3}	2×10^{-3}
124M	B	16	32	32	32	32	64	64	64	128	128	128	256
	η	10^{-3}	10^{-3}	10^{-3}	10^{-3}	10^{-3}	10^{-3}	10^{-3}	10^{-3}	10^{-3}	10^{-3}	10^{-3}	10^{-3}
302M	B	32	32	32	32	32	64	64	128	128	128	128	256
	η	5×10^{-4}	5×10^{-4}	5×10^{-4}	10^{-3}	10^{-3}	10^{-3}	10^{-3}	10^{-3}	10^{-3}	10^{-3}	10^{-3}	10^{-3}
588M	B	–	–	32	32	64	64	64	128	128	128	256	256
	η			5×10^{-4}	5×10^{-4}	5×10^{-4}	5×10^{-4}	5×10^{-4}	5×10^{-4}	5×10^{-4}	5×10^{-4}	10^{-3}	10^{-3}
983M	B	–	–	–	64	64	64	64	128	128	128	256	256
	η				5×10^{-4}	5×10^{-4}	5×10^{-4}	5×10^{-4}	5×10^{-4}	5×10^{-4}	5×10^{-4}	5×10^{-4}	5×10^{-4}
1.7B	B	–	–	–	–	–	64	128	128	128	128	256	256
	η						5×10^{-4}	5×10^{-4}	5×10^{-4}	5×10^{-4}	5×10^{-4}	5×10^{-4}	5×10^{-4}

H SCALING LAW RESULTS

In addition to language modeling performance at a fixed scale, we also test for scaling behavior for varying model size and training data size (Kaplan et al., 2020; Hoffmann et al., 2022). We follow the recipe of Ajroldi et al. (2026) on varying the model and data size of training with a linear warmup-stable-decay (WSD) schedule with AdamW (Loshchilov & Hutter, 2019), using model sizes of $47M$, $124M$, $302M$, $588M$, $983M$ and $1.71B$ parameters and data scales of $1B$, $1.5B$, $2B$, $3B$, $4B$, $6B$, $12B$, $20B$, $30B$, $50B$, $80B$, $120B$ tokens (leaving out the largest data scales $200B$ and $300B$ here and adding some at the low end). To save compute, we rely on their optimal batch size and learning rates scaling laws, where we round their optimal scaling laws to a grid, assuming invariance of these optimal points towards a change of sequence mixing backbone or hybrid model architecture. We follow the scaling model of Videau et al. (2026); Busbridge et al. (2025) in the parametrization of the loss for a parametric fit, which also worked well in Ajroldi et al. (2026). We

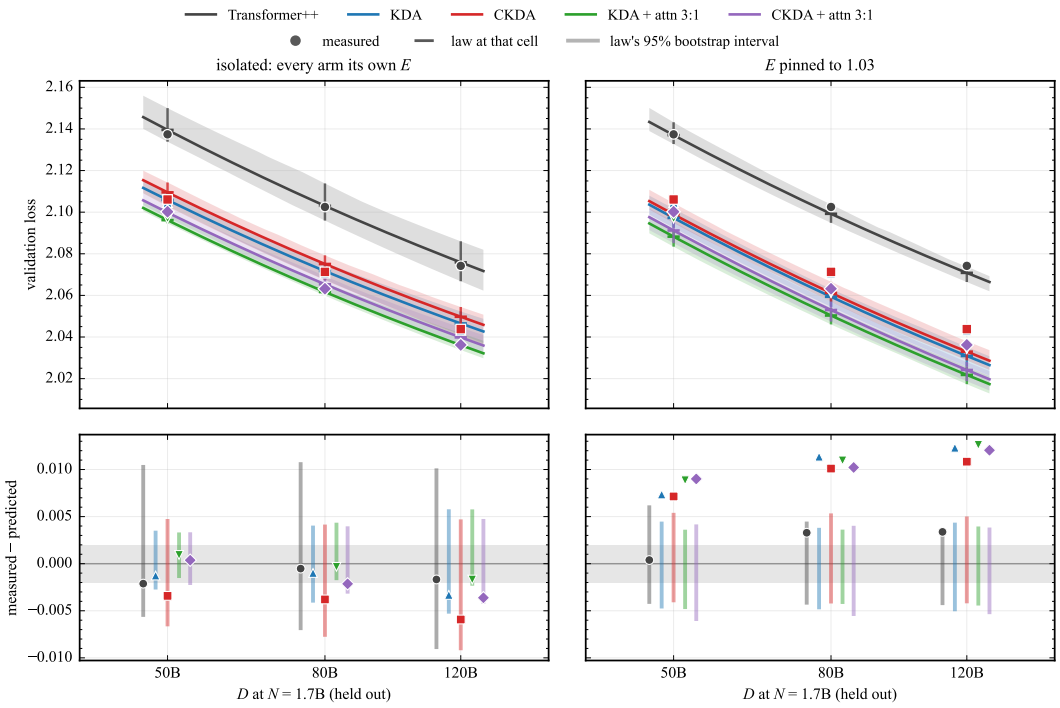

Figure 16: Errors of the $1.7B$ model size holdout points for different data budgets. We compare both the isolated (per model arch) fit and the one with loss noise offset E pinned to the number from Ajroldi et al. (2026). Measured losses are within the confidence bounds of the scaling fits using per-model isolated parameters, with a slight systematic prediction above the measured validation loss. The fit with pinned irreducible loss from Ajroldi et al. (2026) does not extrapolate well here, massively predicting below the measured loss. There might be some systematic inconsistencies also given the fitted values for E in isolation are well above the value from the reference paper.

Table 7: Language modeling results. The best-performing configuration for each evaluation is highlighted in **bold**, and the second best is underlined; markings consider only the Nemotron-CC rows. Nemotron-CC runs use 15B training tokens and FineWeb runs use 45B tokens; both blocks average the same ten accuracy tasks. Validation perplexity is measured on each block's own training distribution, and differences in training data and token budget prevent the cross-block comparison from isolating architecture effects. Abbreviations are expanded in Table 9.

Architecture	Perplexity (↓)			Accuracy (↑)										
	Val.	Wiki.	Lamb.	Hella.	W.grad	W.grande	Lamb.	COPA	PIQA	ARC-e.	ARC-c.	BBQA-Wiki	OBQA	Avg.
Dense Attention	17.99	**24.15**	24.54	42.36	63.37	**53.75**	**39.92**	70.00	68.44	59.81	29.61	48.79	33.00	50.90
DeltaProduct $k = 2$	18.87	28.07	31.90	39.50	62.64	52.09	32.00	65.00	68.17	56.61	27.39	46.79	34.20	48.44
DeltaProduct $k = 2$, $\beta \in [0, 2]$	19.04	28.38	30.43	39.40	61.54	50.67	33.20	64.00	66.65	57.45	27.47	46.85	33.60	48.08
GDN	18.29	27.18	25.28	42.30	66.67	51.14	36.13	65.00	68.28	57.87	28.41	47.47	34.60	49.79
GDN $\beta \in [0, 2]$	18.10	27.22	25.21	42.15	66.30	51.54	36.00	63.00	67.79	58.12	29.10	47.40	35.00	49.64
KDA $\boldsymbol{\alpha} \in [0, 1]$ $\beta \in [0, 1]$	**17.45**	25.29	19.67	43.87	66.30	52.72	39.71	67.00	68.55	59.81	30.03	**51.21**	34.00	51.32
KDA $\boldsymbol{\alpha} \in [0, 1]$ $\beta \in [0, 2]$	17.55	25.51	20.53	43.79	63.74	52.57	39.10	66.00	68.34	59.05	30.12	49.48	33.60	50.58
KDA $\boldsymbol{\alpha} \in [-1, 1]$ $\beta \in [0, 2]$ (*with ablations*)														
standard	17.74	25.86	22.50	43.22	**69.96**	51.38	38.13	62.00	**69.75**	60.40	28.75	51.20	**36.20**	51.10
spread	17.49	25.34	20.21	**44.04**	68.86	53.67	39.67	68.00	69.53	61.32	**31.23**	51.10	35.60	**52.30**
spread, no SiLU on keys	17.49	25.30	**19.64**	43.65	65.93	53.12	39.74	**72.00**	68.55	61.07	30.12	50.48	34.00	51.87
spread, $\beta \in [0, 1]$	**17.45**	25.26	20.17	43.91	64.10	52.41	39.28	67.00	68.82	**61.91**	30.29	50.34	35.80	51.39
FineWeb, 45B tokens														
KDA $\boldsymbol{\alpha} \in [0, 1]$ $\beta \in [0, 2]$	17.02	23.46	13.66	47.08	64.84	54.22	45.29	68.00	70.84	52.82	26.02	54.32	32.60	51.60
KDA $\boldsymbol{\alpha} \in [0, 1]$ $\beta \in [0, 2]$, no SiLU on keys	17.03	23.43	13.67	47.19	68.13	55.09	45.57	67.00	69.80	52.02	26.11	55.79	31.80	51.85
CKDA $\boldsymbol{\alpha} \in [-1, 1]$ $\beta \in [0, 2]$ (*with ablations*)														
standard	17.00	23.29	14.18	46.76	68.86	54.22	44.42	66.00	70.08	52.57	27.30	56.12	31.80	51.81
standard, no SiLU on keys	17.01	23.20	14.21	46.72	70.70	56.04	44.11	65.00	70.73	52.31	25.26	56.71	31.40	51.90
spread	17.10	23.92	13.83	46.98	67.77	54.70	44.75	68.00	70.13	52.31	26.02	53.49	31.80	51.60
spread, no SiLU on keys	17.12	23.76	14.34	46.40	67.77	52.33	44.58	71.00	70.84	53.07	27.39	54.22	31.80	51.94
spread, β spread init	17.16	23.80	14.33	46.70	68.86	54.62	44.23	70.00	70.95	51.85	26.88	54.99	33.00	52.21

Table 8: RULER needle retrieval at 1.3B parameters / 100B FineWeb-Edu tokens, 4K training sequences; accuracy (%) over 500 samples per cell. It remains to be investigated why KDA without extended gates is much worse here.

Model	S-NIAH-1				S-NIAH-2				S-NIAH-3			MK-NIAH-1		
	1K	2K	4K	8K	1K	2K	4K	8K	1K	2K	4K	1K	2K	4K
KDA, bounded gate (ours)	32.2	65.8	94.0	68.8	73.6	100.0	97.6	21.0	7.4	46.6	9.8	48.6	26.2	26.0
CKDA (ours)	100.0	100.0	100.0	87.8	100.0	99.8	88.6	33.2	94.4	88.2	51.0	56.0	46.6	37.0
ours, hybrid - 3:1 full NoPE gated attention														
KDA + attn 3:1 (ours)	100.0	100.0	100.0	100.0	99.8	99.2	99.2	70.2	99.4	98.0	95.8	73.8	60.2	63.8
CKDA + attn 3:1 (ours)	99.8	96.8	81.2	53.4	93.2	98.6	83.2	62.0	97.6	91.0	86.8	95.0	94.4	94.4
Hatamizadeh et al. (2026), recurrent														
Mamba-2	100.0	100.0	97.0	55.8	99.6	99.6	62.6	21.0	59.2	38.6	14.4	29.0	21.2	21.4
Gated DeltaNet	99.8	100.0	100.0	97.6	100.0	100.0	87.2	32.0	89.8	54.2	60.6	58.0	37.0	27.8
KDA	100.0	100.0	99.2	70.6	100.0	100.0	89.0	30.6	77.4	63.2	26.2	54.0	44.2	28.0
Mamba-3 (SISO)	100.0	99.0	63.4	27.8	99.8	99.0	59.4	25.2	60.2	35.6	12.2	44.8	27.4	20.2
Mamba-3 (MIMO)	100.0	99.8	93.0	35.6	99.8	98.8	64.2	27.2	89.2	72.4	29.2	49.4	19.2	18.0
Gated DeltaNet-2	100.0	100.0	100.0	97.8	100.0	100.0	93.0	39.2	92.0	89.8	31.8	72.6	51.4	37.8

compare KDA and CKDA, with their hybrids against a Transformer++ (SwiGLU, QK-norm, RoPE) baseline.

H.1 Scaling-law methodology

The grid. Each architecture is trained on a shared (N, D) grid of six model sizes (47M–1.7B parameters, N from 4.73×10^7 to 1.71×10^9) crossed with at least five token budgets ($D \in \{1, 1.5, 2, 3, 4, 6, 12, 20, 30, 50, 80, 120\}$ BT or a reduced central subset for KDA baseline), on the high-quality subset of Nemotron-CC tokenized with the GPT-NeoX-20B tokenizer at a sequence length of 4096. Every non-attention arm is parameter-matched to the attention baseline at its rung by adjusting the FFN dimension d_{ffn} - depth, d_{model}, head count and head dimension are held fixed across arms. Batch size $b^\star$ and peak learning rate $\eta^\star$ derived from to the hyperparameter scaling laws derived in Ajroldi et al. (2026) rounded to a logarithmic grid, see Table 6. We depart from that reference in exactly two respects, both applied uniformly across the grid: $\beta_2 = 0.95$ at every rung, where the reference switches from 0.99 below 300M and so would place a hyperparameter change inside our N axis; and no bias terms anywhere outside the (C)KDA layers. We hold out the $1.7B$ points at $50B$, $80B$ and $120B$ tokens for measuring extrapolation capabilities of our fits.

Table 9: Full names of abbreviated evaluation benchmarks.

Abbreviation	Full Name
Wiki.	WikiText
Lamb.	LAMBADA OpenAI
Hella.	HellaSwag
W.grad	Winograd Schema Challenge
W.grande, Wino.	WinoGrande
COPA	Choice of Plausible Alternatives
PIQA	Physical Interaction QA
ARC-e.	AI2 Reasoning Challenge (Easy)
ARC-c.	AI2 Reasoning Challenge (Challenge)
BBQA	BigBench QA
OBQA	OpenBook QA
SIQA	Social IQa (Social Interaction Question Answering)
BoolQ	Boolean Questions
SQuAD	Stanford Question Answering Dataset (version 2.0)
SWDE	Structured Web Data Extraction
FDA	Food and Drug Administration 510(k) document extraction
S-NIAH-1/2/3	Single Needle-in-a-Haystack (variants 1–3)
MK-NIAH-1	Multi-Key Needle-in-a-Haystack (variant 1)
Val.	Validation set
Avg.	Average across accuracy tasks

Endpoints. Every cell is trained under a warmup–stable–decay schedule, and the loss we fit is the validation loss (over 0.838 B tokens) of the *annealed* endpoint. Within a rung, cells sharing $b^\star$ and $\eta^\star$ are run as a single constant-rate trunk with a cooldown branched from it at $0.8D$ and annealed linearly over the remaining 20%, rather than as independent runs to save compute.

Functional forms. We fit the non-separable Skaling form of Videau et al. (2026); Busbridge et al. (2025), the one Ajroldi et al. (2026) find extrapolates reliably on this corpus:

$$L_{\text{Skaling}}(N, D) = E + \left(AN^{-\alpha} + BD^{-\beta}\right)^{k}. \tag{25}$$

Two treatments of the irreducible loss. Every fit below gives each architecture its own A, α, B, β, k and chained-cell offset, and the three differ in how the irreducible loss E is modeled. Table 10 fits each arm in complete isolation, so E is free and separate per arm; Table 11 pins E to the value of Ajroldi et al. (2026), as they have a more extensive, compute heavy grid. Since the irreducible loss should be independent of the model - assuming each can converge fully at infinite scale - this is a valid assumption.

Estimation. Following Ajroldi et al. (2026), the residual is taken in log space, $\log \hat{L} - \log L$, and A and B are fitted as $\log A$ and $\log B$. The residuals enter a Huber loss with scale 10^{-3} — a relative 0.1%, about 0.002–0.003 nats at the losses here, which is the measured run-to-run reproducibility of the ladder (two bit-identical configurations differ by up to 0.002 nats): cells agreeing to within noise enter quadratically, and a single outlier cannot dominate. Because the form is strongly multi-modal in (A, α, B, β, k), each fit is started from 600 randomized points, log-uniform over the prefactors and uniform over the exponents. All starts are screened by a batched Levenberg–Marquardt descent of 60 iterations; the best eight are then polished by trust-region least squares on the full Huber objective, and the lowest-cost optimum is kept.

Uncertainty. Intervals are obtained by bootstrap rather than from a covariance matrix: the loss is Huber, the objective is multi-modal. We resample the endpoints with replacement, stratified within each arm so that every replicate keeps the design of the data, and refit 200 times; each replicate is warm-started from the full-data optimum and given 6 further random starts. We report the 2.5th and 97.5th percentiles as a subscript on each estimate. The same replicates give the prediction intervals at the held-out cells and the band drawn around each fitted law in the held-out figure. These intervals

describe the sampling variability of the fit; they do not include the run-to-run noise of an individual endpoint (about 0.002–0.004 nats) or any systematic effect of the fixed (B, η) grid.

Compute-optimal budgets. Where the figures mark a compute-optimal budget, it is $D^\star(N)$, the budget at which a model of size N is compute-optimal under $C = 6ND$: the inverse of the compute-optimal frontier of the fitted law. It is computed from the isolated fit made without the held-out cells, and Hoffmann et al. (2022) found optimal ratio of 22 tokens per parameter is drawn beside it.

H.2 Scaling Law Results

All trained model architectures can be well fitted to the chosen parametric form (Busbridge et al., 2025; Videau et al., 2026), see Tables 10 and 11. A held-out point on the largest scale can be predicted with low error for each model, see Figure 16. Re-using the irreducible loss found in Ajroldi et al. (2026) (which is smaller than our fitted values) leads to a slight shift of fitted parameters, namely the data scaling exponent and coupling exponent are smaller. This means that larger scale experiments could lead to a non-negligible change still in the fitted parameters. While the validation loss slightly favors KDA over CKDA, the downstream evaluations show a mixed picture here, see Table 14. Given the confidence bounds, CKDA and KDA perform on par in vanilla language modeling. All measured validation losses are shown in Tables 12 and 13.

The held out points fit well within the uncertainty bounds of the isolated fit. Reusing the irreducible loss from Ajroldi et al. (2026) leads to a worse extrapolation capability - the values for E in the isolated fits also don't match very well with the values from Ajroldi et al. (2026). This discrepancy hints at a systematic error, potentially due to an incomplete parametrization of the loss and should be investigated in future work.

The derived compute-optimal scaling for the model architectures is shown in Figure 18. The observed decline of the advantage of the recurrent / hybrid models over the Transformer going to large over-training regimes at small scales (see 15) is not clearly visible in the scaling exponents, though there are differences. A detailed exploration of this remains to be investigated in future work.

Table 10: Fitted parameters with $1.7B$ (for $50B$, $80B$, $120B$ tokens) held out, all parameters fit per architecture in an isolated way. RMSE is in-sample, over that arm's remaining cells; the held-out error is in Figure 16. Note that our data is missing the regime of about $3\times$ more compute ($300B$ tokens) - leading to a slightly different offset compared to Ajroldi et al. (2026). Interestingly, KDA and CKDA variants have a higher irreducible loss (although it should be architecture independent), hinting at a potential systematic error in the parametrization of the loss.

Arm	E	A	α	B	β	k	RMSE
Transformer++	$1.1439_{[0.9108,\,1.3533]}$	$1.20 \times 10^4_{[4.62\times10^3,\,2.85\times10^4]}$	$0.473_{[0.423,\,0.519]}$	$2.37 \times 10^4_{[5.96\times10^3,\,5.63\times10^4]}$	$0.443_{[0.381,\,0.472]}$	$0.422_{[0.360,\,0.489]}$	0.00800
KDA	$1.3071_{[1.2175,\,1.3813]}$	$3.56 \times 10^4_{[2.42\times10^4,\,6.36\times10^4]}$	$0.548_{[0.531,\,0.573]}$	$3.34 \times 10^4_{[1.45\times10^4,\,7.00\times10^4]}$	$0.474_{[0.435,\,0.513]}$	$0.438_{[0.399,\,0.462]}$	0.00577
CKDA	$1.3146_{[1.1956,\,1.3710]}$	$3.39 \times 10^4_{[2.33\times10^4,\,5.99\times10^4]}$	$0.547_{[0.529,\,0.569]}$	$3.27 \times 10^4_{[1.57\times10^4,\,9.94\times10^4]}$	$0.473_{[0.442,\,0.520]}$	$0.440_{[0.396,\,0.465]}$	0.00578
KDA + attn 3:1	$1.3016_{[1.2452,\,1.3933]}$	$2.62 \times 10^4_{[1.63\times10^4,\,4.04\times10^4]}$	$0.534_{[0.511,\,0.556]}$	$2.42 \times 10^4_{[1.33\times10^4,\,4.14\times10^4]}$	$0.460_{[0.430,\,0.491]}$	$0.451_{[0.426,\,0.490]}$	0.00596
CKDA + attn 3:1	$1.3228_{[1.2354,\,1.3904]}$	$2.71 \times 10^4_{[1.87\times10^4,\,5.20\times10^4]}$	$0.538_{[0.522,\,0.562]}$	$2.36 \times 10^4_{[1.03\times10^4,\,4.17\times10^4]}$	$0.461_{[0.420,\,0.485]}$	$0.455_{[0.413,\,0.487]}$	0.00564
Ajroldi et al. (2026)	$1.03_{\pm0.087}$	$2.34 \times 10^4_{\pm1.40\times10^4}$	$0.491_{\pm0.030}$	$4.63 \times 10^3_{\pm2.90\times10^3}$	$0.350_{\pm0.022}$	$0.411_{\pm0.033}$	–

Table 11: E pinned to 1.03, the value of Ajroldi et al. (2026), fitted on the same corpus and held-out split ($1.7B$ models at $50B$, $80B$, $120B$ tokens held out) over a ladder reaching $300B$ tokens (ours is up to $120BT$).

Arm	E	A	α	B	β	k	RMSE
Transformer++	1.0300	$1.29 \times 10^4_{[2.71\times10^3,\,2.62\times10^4]}$	$0.463_{[0.389,\,0.500]}$	$1.95 \times 10^4_{[4.39\times10^3,\,4.67\times10^4]}$	$0.422_{[0.362,\,0.458]}$	$0.399_{[0.376,\,0.466]}$	0.00813
KDA	1.0300	$6.53 \times 10^4_{[2.77\times10^4,\,9.28\times10^4]}$	$0.546_{[0.503,\,0.563]}$	$3.54 \times 10^4_{[1.50\times10^4,\,6.61\times10^4]}$	$0.446_{[0.407,\,0.474]}$	$0.364_{[0.352,\,0.392]}$	0.00887
CKDA	1.0300	$5.72 \times 10^4_{[2.73\times10^4,\,8.87\times10^4]}$	$0.539_{[0.503,\,0.559]}$	$4.20 \times 10^4_{[1.69\times10^4,\,1.01\times10^5]}$	$0.454_{[0.416,\,0.493]}$	$0.364_{[0.349,\,0.389]}$	0.00812
KDA + attn 3:1	1.0300	$3.36 \times 10^4_{[1.16\times10^4,\,6.81\times10^4]}$	$0.516_{[0.464,\,0.549]}$	$2.16 \times 10^4_{[8.80\times10^3,\,6.37\times10^4]}$	$0.428_{[0.390,\,0.472]}$	$0.385_{[0.355,\,0.420]}$	0.00886
CKDA + attn 3:1	1.0300	$3.94 \times 10^4_{[1.40\times10^4,\,7.87\times10^4]}$	$0.524_{[0.472,\,0.559]}$	$2.21 \times 10^4_{[8.03\times10^3,\,5.42\times10^4]}$	$0.428_{[0.385,\,0.466]}$	$0.381_{[0.354,\,0.412]}$	0.00853
Ajroldi et al. (2026)	$1.03_{\pm0.087}$	$2.34 \times 10^4_{\pm1.40\times10^4}$	$0.491_{\pm0.030}$	$4.63 \times 10^3_{\pm2.90\times10^3}$	$0.350_{\pm0.022}$	$0.411_{\pm0.033}$	–

Table 12: Validation loss at every cell of the ladder, 47M–302M (588M–1.7B in Table 13): annealed endpoints on our held-out Nemotron-CC split. Best per row in bold. These compare the arms with each other and with no published table.

Cell	Transformer++	KDA	CKDA	KDA + attn 3:1	CKDA + attn 3:1
47M/1BT	3.1802	3.1543	3.1514	3.1537	**3.1487**
47M/1.5BT	3.1252	3.0831	3.0812	**3.0772**	3.0774
47M/2BT	3.0630	3.0256	3.0245	3.0216	**3.0194**
47M/3BT	3.0178	2.9824	2.9811	**2.9760**	2.9767
47M/4BT	2.9894	2.9562	2.9547	**2.9488**	2.9504
47M/6BT	2.9528	2.9240	2.9222	**2.9155**	2.9167
47M/12BT	2.8828	2.8658	2.8665	**2.8521**	2.8549
47M/20BT	2.8519	2.8382	2.8393	**2.8241**	2.8263
47M/30BT	2.8319	2.8197	2.8217	**2.8063**	2.8073
47M/50BT	2.8053	2.7973	2.7953	**2.7797**	2.7817
47M/80BT	2.7883	2.7822	2.7792	**2.7640**	2.7660
47M/120BT	2.7764	2.7713	2.7678	**2.7526**	2.7548
124M/1BT	2.9801	2.9462	2.9451	**2.9301**	2.9343
124M/1.5BT	2.9051	2.8644	2.8721	2.8563	**2.8519**
124M/2BT	2.8605	2.8178	2.8252	2.8093	**2.8059**
124M/3BT	2.8067	2.7615	2.7678	2.7536	**2.7510**
124M/4BT	2.7739	2.7281	2.7333	2.7195	**2.7174**
124M/6BT	2.7185	2.6873	2.6878	**2.6725**	2.6740
124M/12BT	2.6522	2.6220	2.6233	**2.6075**	2.6091
124M/20BT	2.6104	2.5849	2.5859	**2.5706**	2.5713
124M/30BT	2.5833	2.5589	2.5578	**2.5441**	2.5458
124M/50BT	2.5492	2.5301	2.5291	**2.5158**	2.5166
124M/80BT	2.5232	2.5085	2.5079	**2.4941**	2.4947
124M/120BT	2.5046	2.4930	2.4949	**2.4785**	2.4796
302M/1BT	2.8916	2.8512	2.8583	2.8402	**2.8372**
302M/1.5BT	2.8039	2.7570	2.7642	2.7478	**2.7455**
302M/2BT	2.7500	2.6994	2.7065	2.6909	**2.6888**
302M/3BT	2.6716	2.6138	2.6193	**2.6080**	2.6116
302M/4BT	2.6342	2.5761	2.5819	**2.5705**	2.5740
302M/6BT	2.5779	2.5237	2.5263	**2.5143**	2.5185
302M/12BT	2.5053	2.4524	2.4549	**2.4439**	2.4468
302M/20BT	2.4655	2.4080	2.4094	**2.3960**	2.3981
302M/30BT	2.4299	2.3770	2.3784	**2.3650**	2.3670
302M/50BT	2.3914	2.3441	2.3454	**2.3324**	2.3337
302M/80BT	2.3591	2.3193	2.3203	**2.3075**	2.3084
302M/120BT	2.3269	2.2978	2.3009	**2.2851**	2.2874

Table 13: Validation loss at every cell of the ladder, continued from Table 12: 588M–1.7B. Same conventions.

Cell	Transformer++	KDA	CKDA	KDA + attn 3:1	CKDA + attn 3:1
588M/2BT	2.6685	2.6219	2.6282	**2.6121**	2.6159
588M/3BT	2.5962	2.5472	2.5528	**2.5379**	2.5414
588M/4BT	2.5562	2.5115	2.5149	**2.5031**	2.5051
588M/6BT	2.4943	2.4478	2.4517	**2.4391**	2.4414
588M/12BT	2.4098	2.3612	2.3664	**2.3528**	2.3559
588M/20BT	2.3534	2.3143	2.3191	2.3051	**2.3049**
588M/30BT	2.3172	2.2775	2.2815	**2.2676**	2.2683
588M/50BT	2.2770	2.2386	2.2417	**2.2284**	2.2295
588M/80BT	2.2425	2.2053	2.2070	**2.1945**	2.1953
588M/120BT	2.2144	2.1837	2.1849	**2.1729**	2.1737
983M/3BT	2.5565	2.5064	2.5108	**2.4960**	2.4969
983M/4BT	2.5030	2.4533	2.4581	**2.4426**	2.4441
983M/6BT	2.4382	2.3885	2.3936	**2.3775**	2.3797
983M/12BT	2.3496	2.2997	2.3040	**2.2887**	2.2914
983M/20BT	2.2844	2.2484	2.2532	**2.2369**	2.2405
983M/30BT	2.2464	2.2098	2.2141	**2.1981**	2.2017
983M/50BT	2.2070	2.1686	2.1722	**2.1571**	2.1608
983M/80BT	2.1715	2.1392	2.1424	**2.1259**	2.1290
983M/120BT	2.1440	2.1133	2.1164	**2.1004**	2.1033
1.7B/6BT	2.3725	2.3359	2.3402	**2.3284**	2.3299
1.7B/12BT	2.2829	2.2496	2.2507	**2.2399**	2.2433
1.7B/20BT	2.2218	2.1882	2.1898	**2.1795**	2.1828
1.7B/30BT	2.1812	2.1477	2.1494	**2.1395**	2.1429
1.7B/50BT	2.1374	2.1046	2.1061	**2.0971**	2.1002
1.7B/80BT	2.1025	2.0707	2.0713	**2.0613**	2.0632
1.7B/120BT	2.0742	2.0432	2.0438	**2.0344**	2.0362

Table 14: Downstream accuracy of models of the scaling ladder (from 6B tokens on), averaged over the nine tasks of Table 2. Highest per row in bold. Below 302M most of the suite is at chance – HellaSwag near 27, ARC-c near 23, OpenBookQA near 25 – so the average at the lower rungs is largely majority-class behaviour on BoolQ and PIQA.

Cell	Transformer++	CKDA	CKDA + attn 3:1	KDA	KDA + attn 3:1
47M/6BT	37.75	38.49	**39.12**	39.08	37.70
47M/12BT	37.62	38.43	**38.60**	38.04	38.11
47M/20BT	38.22	37.84	**39.42**	39.06	37.72
47M/30BT	37.49	39.36	39.46	**40.16**	39.22
47M/50BT	38.94	38.96	38.49	**39.42**	38.96
47M/80BT	39.16	38.09	39.69	38.91	**40.38**
47M/120BT	39.99	39.96	**40.32**	39.83	39.71
124M/6BT	40.06	**42.25**	40.69	41.22	41.06
124M/12BT	41.74	41.29	41.68	**42.41**	41.96
124M/20BT	42.16	**43.67**	42.88	43.41	43.11
124M/30BT	41.52	43.64	43.48	43.53	**43.68**
124M/50BT	44.01	**44.23**	43.51	44.02	43.59
124M/80BT	43.75	44.49	**44.56**	44.29	44.54
302M/6BT	42.39	44.56	44.22	**44.76**	44.50
302M/12BT	44.15	45.90	45.53	45.89	**46.77**
302M/20BT	45.24	**47.19**	47.02	46.12	47.01
302M/30BT	45.19	47.99	47.97	47.73	**48.20**
302M/50BT	46.21	48.25	**49.14**	47.69	48.82
302M/80BT	47.99	49.77	**50.02**	49.86	49.74
588M/6BT	44.15	45.35	45.46	45.75	**45.96**
588M/12BT	46.67	47.67	47.87	47.82	**48.53**
588M/20BT	46.79	48.66	**49.54**	48.57	48.75
588M/30BT	47.98	49.45	**50.79**	50.18	49.50
588M/50BT	49.83	50.44	51.30	**51.43**	51.24
588M/80BT	50.39	52.10	**52.94**	52.70	52.73
588M/120BT	51.73	53.03	53.79	52.97	**53.98**
983M/6BT	46.19	47.14	46.86	46.80	**47.78**
983M/12BT	48.36	49.48	49.33	49.29	**50.40**
983M/20BT	49.12	51.22	50.95	50.51	**51.49**
983M/30BT	50.42	51.95	52.32	51.51	**52.60**
983M/50BT	51.57	54.10	**54.25**	53.13	54.10
983M/80BT	52.36	54.76	**55.68**	54.08	54.50
983M/120BT	53.82	56.07	**56.13**	55.51	55.57
1.7B/6BT	47.61	**48.84**	48.29	48.25	48.35
1.7B/12BT	49.59	**51.34**	50.95	51.10	51.18
1.7B/20BT	51.75	**53.38**	52.53	52.80	52.60
1.7B/30BT	53.29	54.59	54.35	**55.13**	54.91
1.7B/50BT	55.10	**57.21**	55.82	56.87	56.47
1.7B/80BT	55.30	57.07	57.51	**58.06**	57.54
1.7B/120BT	56.81	57.65	**59.08**	58.77	59.00

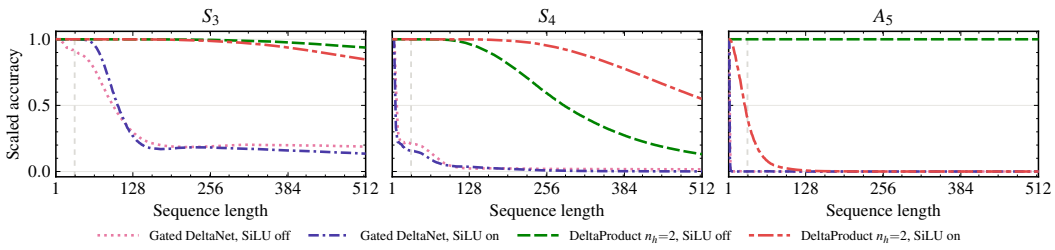

Figure 17: Baseline results for Figure 5

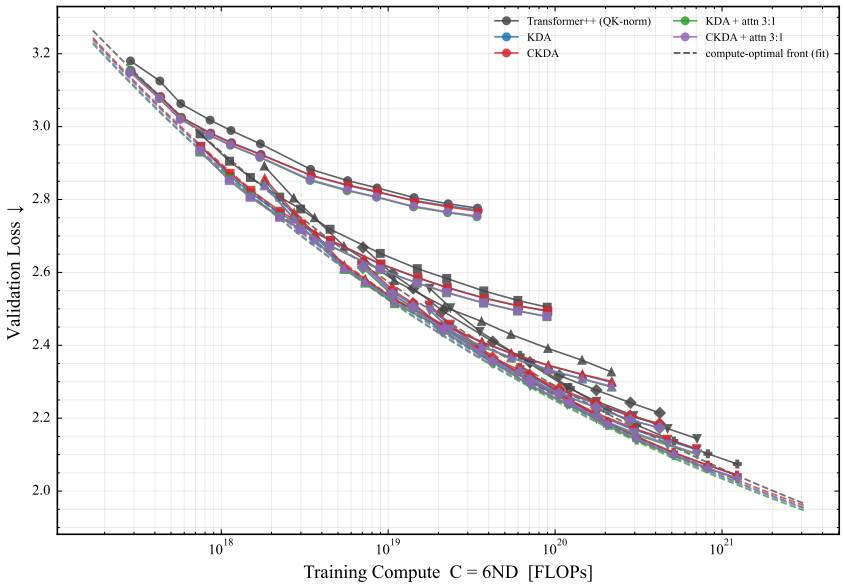

Figure 18: Validation loss vs. Compute for the language model scaling experiments on Nemotron-CC.

I Additional Experimental Results

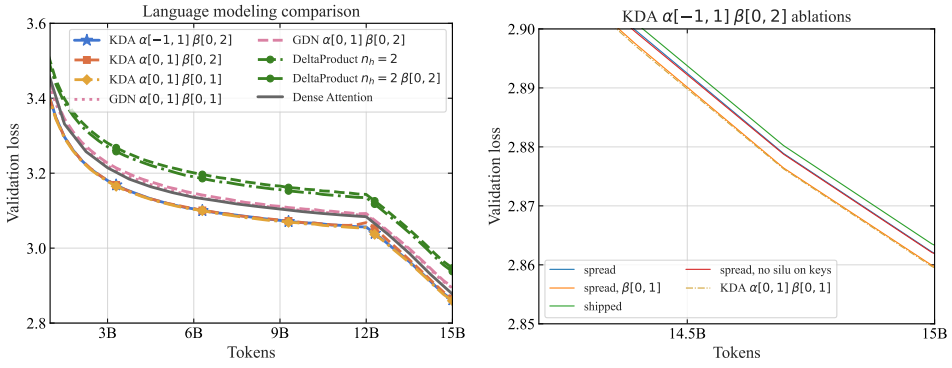

Figure 19: Language modeling ablation results at 340M scale trained on Nemotron-CC: validation loss versus tokens. (Left) Comparison across baselines. (Right) Comparison across ablations shown for the end of the cooldown phase, as loss values are quite close together.

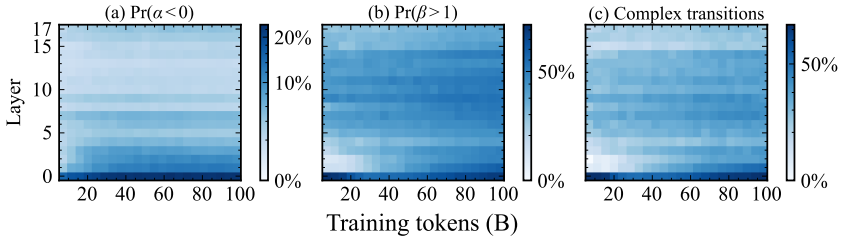

Figure 20: Hybrid CKDA 1.3B, see Figure 7.

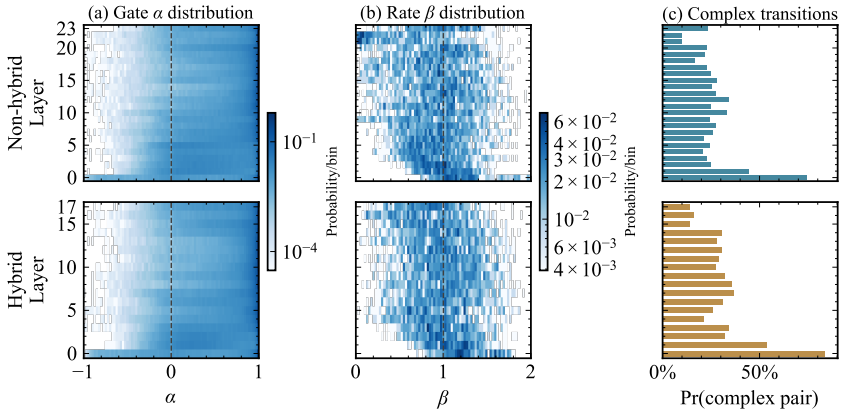

Figure 21: Final layer profiles of gate and β for the CKDA 1.3B non-hybrid and hybrid models after training on 100B tokens.