NON-ABELIAN STATE TRACKING WITH QUATERNIONIC STATE SPACE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent developments in state-space model (SSM) design have improved the state tracking and language modeling capabilities of these models through increasingly expressive transition dynamics. In particular, complex-valued diagonal transitions have been shown to be highly effective and computationally efficient. However, their expressivity remains fundamentally limited by the commutativity of diagonal matrices. In this work, we introduce Icosian, a quaternionic SSM with non-commutative state transitions through input-dependent 4D rotations. Icosian exploits the *exceptional* structure of the 4D rotation group $\mathrm{SO}(4)$, parametrized through pairs of unit quaternions. This parametrization enables fast and numerically stable recurrent computation through quaternion arithmetic, without explicitly materializing and composing rotation matrices. We evaluate Icosian on language modeling across scales from 500M to 3B parameters. Across multiple seeds, Icosian consistently outperforms Mamba-3, Gated DeltaNet, and Gated DeltaProduct, while its additional computational cost decreases with scale to only 3% at 3B parameters. We then characterize the state tracking and recall capabilities of Icosian through synthetic experiments. In contrast to complex diagonal SSMs, a single Icosian layer exhibits length generalization on the non-solvable A_5 group, while deeper models also track rotations drawn uniformly from continuous groups $\mathrm{SO}(3)$ and $\mathrm{SO}(4)$. Overall, our results show that Icosian provides expressive non-commutative state dynamics and achieves notable gains in language modeling, while retaining the computational efficiency required for SSMs at scale.

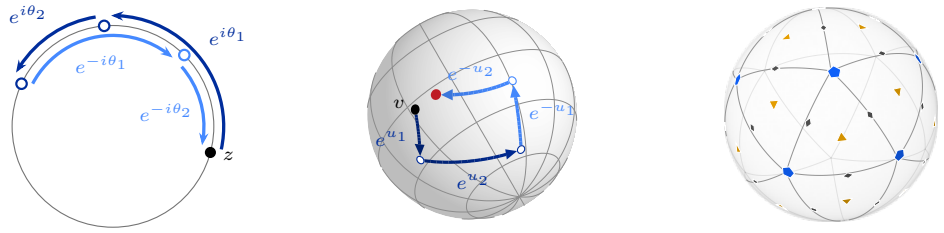
1 INTRODUCTION

The defining characteristic of a state-space model (SSM) is the compression of an entire sequence history into a fixed-size hidden state (Gu et al., 2022; Gu & Dao, 2024; Yang et al., 2024; Siems et al., 2025a; Peng et al., 2025). This compression enables linear-time training and constant-memory inference, making SSMs notably more efficient than self-attention at inference time. At the same time, compression creates a fundamental challenge: all information about the past must be represented and manipulated through the recurrent state dynamics. Indeed, the expressiveness of these dynamics has emerged as a key bottleneck for SSMs, with limitations in state manipulation often translating into weaker state tracking and language modeling performance (Merrill et al., 2024).

Modern SSMs span transition structures ranging from highly efficient scalar-times-identity matrices (Dao & Gu, 2024) over Householder-based transformations (Schlag et al., 2021; Yang et al., 2024; 2025; Siems et al., 2025a) to the most recent complex-diagonal SSMs (Lahoti et al., 2026), the last of which we build upon in this work. While the diagonal complex-valued transitions have substantially improved state tracking capabilities compared to previous real-valued models, these transitions are commutative, limiting the model's expressivity. In particular, no finite-depth diagonal SSM can accumulate the actions of a non-solvable finite group over arbitrary sequence lengths (Merrill et al., 2024; Sarrof et al., 2024; Shakerinava et al., 2026). This motivates transition structures that provide richer, non-commutative dynamics without abandoning efficient computation.

In this work, we introduce **Icosian**, a quaternion SSM that performs selective, non-commutative state transitions through input-dependent 4D rotations (Figure 1). Icosian exploits the *exceptional* (or *accidental*) structure of $\mathrm{SO}(4)$, whose rotations can be represented through pairs of unit quaternions

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.



(a) Commutative composition of complex-valued 2D rotations.

(b) Non-commutative composition of quaternionic 4D rotations.

(c) Extracted representation of A_5 from a trained quaternionic SSM.

Figure 1: **Quaternionic transitions enable non-commutative state dynamics.** Complex-valued 2D rotations compose commutatively (left; Lahoti et al. 2026), whereas quaternionic 4D rotations are non-commutative (center). This structure enables Icosian to represent non-solvable dynamics; when trained to track A_5, the learned transitions directly encode a representation of the group (right).

and efficiently composed through quaternion arithmetic rather than generic matrix operations (Conway & Smith, 2003). As a result, Icosian provides substantially richer recurrent dynamics while retaining the numerical stability and computational efficiency of quaternion arithmetic.

We first evaluate Icosian at language-model scales ranging from 500M to 3B parameters, where it consistently improves upon Mamba-3 (Lahoti et al., 2026), Gated DeltaNet (Yang et al., 2025), and Gated DeltaProduct (Siems et al., 2025a), with an additional computational cost on Mamba-3 of only 3% at 3B parameters. We then characterize Icosian's state dynamics through synthetic experiments. A single Icosian layer length-generalizes on the non-solvable group A_5 (Carter, 2009) and directly learns a representation of the group, while deeper models track continuous rotations drawn from $\mathrm{SO}(3)$ and $\mathrm{SO}(4)$. These results demonstrate that Icosian combines strong language modeling performance with non-commutative state tracking capabilities and efficient recurrent computation. Our main contributions are as follows:

- **Quaternion State-Space Model.** We introduce **Icosian**, a novel selective SSM with transition matrices parameterized as quaternionic representations of $\mathrm{SO}(4)$ rotations. By exploiting the factorization of 4D rotations as unit quaternions, we develop an optimized implementation of Icosian that achieves richer recurrent dynamics while preserving efficient computation.
- **Language Modeling.** We show that Icosian achieves improved language modeling performance over strong contemporary baselines, including Mamba-3, Gated DeltaNet, and Gated DeltaProduct. Our thorough analysis at 500M, 1B, and 3B scales establishes notable perplexity improvements over all considered alternatives.
- **State Tracking and Mechanistic Analysis.** We establish the position of quaternion transitions within the hierarchy of modern SSM architectures through both theoretical and experimental means. We demonstrate length generalization on discrete and continuous non-commutative dynamics and find that a single Icosian layer trained on A_5 directly learns a representation of the group.

2 Preliminaries

We first recapitulate the relevant details on SSMs and then provide details on the quaternionic algebra which we use to define our model.

2.1 State-Space Models

State-space models (SSMs) operate on an underlying continuous system generally defined as:

$$\dot{h}(t) = \boldsymbol{A}(t)h(t) + \boldsymbol{B}(t)x(t) \tag{1}$$

$$y(t) = \boldsymbol{C}(t)h(t) \tag{2}$$

with $\boldsymbol{A}$ mapping to $\mathbb{R}^{N\times N}$, $\boldsymbol{B}$ to $\mathbb{R}^{N}$ and $\boldsymbol{C}$ to $\mathbb{R}^{1\times N}$, and N the state size.

Complex-valued diagonal SSMs. Recent work (Lahoti et al., 2026) has shown that complex-valued diagonal SSMs are a particularly efficient and expressive class of SSMs. Specifically, the Mamba-3 state-space dynamic is defined as

$$\dot{h}(t) = (a(t) + i\theta(t))h(t) + (b(t) + i\hat{b}(t))x(t) \tag{3}$$
$$y(t) = \mathrm{Re}((c(t) - i\hat{c}(t))h(t)) \tag{4}$$

with $h(t) \in \mathbb{C}$ and all other functions mapping to $\mathbb{R}$[1].

In practice, we have access to a sequence of sampled inputs $x_k := x(t_k) \in \mathbb{R}$ with $t_k = t_{k-1} + \Delta_k$ the discretized time-step and $\Delta_k := t_k - t_{k-1} \in \mathbb{R}_{>0}$ the index-dependent step size. We wish to obtain a discrete sequence of states $h_k := h(t_k) \in \mathbb{C}$. Given h_{k-1}, defined analogously to h_k, the closed-form solution to Equation 3 at time t_k is:

$$h_k = e^{\int_{t_{k-1}}^{t_k}(a(s)+i\theta(s))ds}h_{k-1} + \int_{t_{k-1}}^{t_k} e^{\int_{\tau}^{t_k}(a(s)+i\theta(s))ds}(b(\tau) + i\hat{b}(\tau))x(\tau)d\tau \tag{5}$$

Discretization. The state-space equation is then discretized as follows, following Lahoti et al. (2026). First, factors of the form $e^{\int_{t_1}^{t_2} a(s)ds}$ are approximated by the right-hand rule as $e^{(t_2-t_1)a(t_2)}$. Afterwards, the remaining integral term is treated using generalized trapezoidal quadrature with an input-dependent coefficient $\lambda_k \in [0, 1]$. With a_k, θ_k defined analogously to x_k, h_k, this results in:

$$h_k = e^{\Delta_k(a_k+i\theta_k)}h_{k-1} + (1-\lambda_k)\Delta_k e^{\Delta_k(a_k+i\theta_k)}(b_{k-1} + i\hat{b}_{k-1})x_{k-1} + \lambda_k\Delta_k(b_k + i\hat{b}_k)x_k \tag{6}$$

2.2 Quaternions and Four-Dimensional Rotations

Our first goal is to define an SSM tbat generalizes Equation 3 by inducing a non-commutative rotation of the state. We do this using *quaternions*.

Definition 1 (Quaternions)**.** *The set of quaternions, denoted by $\mathbb{H}$, forms a four-dimensional vector space over $\mathbb{R}$ with a basis $\{1, i, j, k\}$. Writing $q \in \mathbb{H}$ as $q = a + bi + cj + dk$ with $a, b, c, d \in \mathbb{R}$ and imposing the relation $i^2 = j^2 = k^2 = ijk = -1$ associates a normed division algebra with $\mathbb{H}$.*

The definition implies an identification of $\mathbb{H}$ with $\mathbb{R}^4$ augmented by an algebra that provides associative multiplication and well-defined division. Associativity enables efficient parallelism (Blelloch, 1990), while division enables simple renormalization, useful for remedying numerical drifts.

Writing $q = a + \mathbf{v}$ and $p = e + \mathbf{w}$ with $\mathbf{v} = (b, c, d)$, $\mathbf{w} \in \mathbb{R}^3$ the imaginary parts, the imposed relation on i, j, k determines quaternion multiplication as:

$$qp = \big(ae - \mathbf{v}\cdot\mathbf{w}\big) + \big(a\mathbf{w} + e\mathbf{v} + \mathbf{v}\times\mathbf{w}\big). \tag{7}$$

This operation is associative but *does not commute* due to the cross-product term $\mathbf{v} \times \mathbf{w}$.

For $q = a + bi + cj + dk$, define the conjugate as $\bar{q} : q \mapsto a - bi - cj - dk$. All non-zero quaternions invert as $q^{-1} = \bar{q}/\|q\|^2$. With the multiplicative norm, $\|pq\| = \|p\|\|q\|$, the unit quaternions $S^3 := \{q \in \mathbb{H} : \|q\| = 1\}$ form a group.

We denote the set of all imaginary quaternions, i.e. those with null real part ($a = 0$), as $\mathrm{Im}\,\mathbb{H}$. In the same manner as the exponential of an imaginary number ($e^{i\phi}$, $\phi \in \mathbb{R}$) traverses the complex unit circle $S^1 := \{z \in \mathbb{C} : |z| = 1\}$, the exponential of an imaginary quaternion traverses the quaternion unit hypersphere S^3. For $\eta \in \mathrm{Im}\,\mathbb{H}$, the Taylor series determines $\exp\colon \mathrm{Im}\,\mathbb{H} \to S^3$ as:

$$\exp\eta = \cos\|\eta\| + \sin\|\eta\|\,\frac{\eta}{\|\eta\|}. \tag{8}$$

Unit quaternions S^3 describe 4D rotations $\mathrm{SO}(4)$ as per Proposition 1 (Conway & Smith, 2003).

[1]For exposition purposes, we only consider a single dimension of a diagonal complex-valued SSM.

Proposition 1 (Quaternionic Double Cover of $\mathrm{SO}(4)$)**.** *For unit quaternions $l, r \in S^3$, the map $\mathcal{R}_{l,r} \colon v \mapsto l\, v\, \bar{r}$ is, under the identification $\mathbb{H} \cong \mathbb{R}^4$, an element of $\mathrm{SO}(4)$. Every element of $\mathrm{SO}(4)$ arises in this way, and $\mathcal{R}_{l,r} = \mathcal{R}_{l',r'}$ if and only if $(l', r') = \pm(l, r)$. Consequently, the direct product $S^3 \times S^3$ of unit quaternions forms a double cover of $\mathrm{SO}(4)$ as $\mathrm{SO}(4) \cong (S^3 \times S^3)/\{\pm(1,1)\}$.*

Briefly, every 4D rotation matrix $\mathbf{Q} \in \mathrm{SO}(4)$ admits two representations in form of two pairs of unit quaternions $\pm(l, r) \in S^3 \times S^3$, with (l, r) defined in terms of $\mathbf{Q}$. The action of $\mathbf{Q} \in \mathrm{SO}(4)$ on $\mathbf{x} \in \mathbb{R}^4$ is obtained by interpreting $\mathbf{x} = (x_1, x_2, x_3, x_4)^\top$ as $x = x_1 + x_2 i + x_3 j + x_4 k \in \mathbb{H}$ and acting on it using either quaternion pair as $lx\bar{r} = (-l)x(-\bar{r})$. We discuss further properties of rotational algebras in Appendix C. In particular, dimension four is the largest in which rotations compose through an associative division algebra, not requiring naïve matrix-matrix product. This property makes it a natural limit for efficient computation.

3 The Quaternionic State-Space Model

We start by specifying a quaternionic dynamical system and then demonstrate that the resulting discrete recurrence can be implemented using existing efficient parallel algorithms (Dao & Gu, 2024). All proofs and derivations are in Appendix D.

3.1 Defining the Quaternionic Continuous-Time System

The core idea of our approach is to generalize the transition function of a diagonal SSM such that it expresses rotations in 4D space. We claim that the following SSM is an appropriate generalization.

Definition 2 (Quaternionic State-Space Model)**.** *Let $a, x \colon [0,T] \to \mathbb{R}$, $b, c \colon [0,T] \to \mathbb{H}$ and $l, r \colon [0,T] \to \operatorname{Im}\mathbb{H}$ be continuous. The quaternionic state-space model is given by*

$$\dot{h}(t) = a(t)h(t) + l(t)h(t) - h(t)r(t) + b(t)x(t), \qquad h(0) = h_0 \in \mathbb{H}, \tag{9}$$

$$y(t) = \operatorname{Re}\big(\bar{c}(t)\, h(t)\big), \tag{10}$$

with state $h(t) \in \mathbb{H}$ and output $y(t) \in \mathbb{R}$.

We show that this system is well-grounded, as it induces the desired action on the state; it rotates it using general elements of $\mathrm{SO}(4)$. We base our argumentation on the system executing with $a(t) = b(t) = 0$. In this case, the only allowed evolution of the state is a time-dependent 4D rotation of the initial condition, and Proposition 2 shows that this is indeed the form the solution takes. In fact, for every sufficiently regular mapping $[0,T] \to \mathrm{SO}(4)$, there exists a choice of l and r in Equation 9 that induces it.

Proposition 2 (Rotational Flow of the Quaternionic ODE)**.** *Let $l, r \colon [0,T] \to \operatorname{Im}\mathbb{H}$ be continuous, and consider Equation 9 with $a \equiv b \equiv 0$. Its solution is*

$$h(t) = U_L(t)\, h(0)\, \overline{U_R(t)},$$

where $U_L, U_R \colon [0,T] \to S^3$ solve $\dot{U}_L = l\, U_L$ and $\dot{U}_R = r\, U_R$ with $U_L(0) = U_R(0) = 1$. Conversely, for every continuously differentiable curve $R \colon [0,T] \to \mathrm{SO}(4)$ with $R(0) = \mathrm{id}$ there exist l, r for which the solution satisfies $h(t) = R(t)\, h(0)$.

3.2 Discretizing the Quaternion ODE

We apply variation of constants to Equation 9 to obtain a closed-form solution in $[t_{k-1}, t_k]$.

Proposition 3 (Closed-Form Expression for the Quaternionic System)**.** *Let $a(\cdot), x(\cdot) \in \mathbb{R}$, $b(\cdot) \in \mathbb{H}$ and $l, r \in \operatorname{Im}\mathbb{H}$ be continuous on $[t_{k-1}, t_k]$. Given h_{k-1}, the analytical expression for h_k in Equation 9 is*

$$\begin{aligned} h_k = {} & \exp\left(\int_{t_{k-1}}^{t_k} a(s)\, ds\right) \mathcal{T}\exp\Big(\int_{t_{k-1}}^{t_k} l(s)\, ds\Big)\, h_{k-1}\, \overline{\mathcal{T}}\exp\Big(\int_{t_{k-1}}^{t_k} \bar{r}(s)\, ds\Big) \\ & + \int_{t_{k-1}}^{t_k} \exp\left(\int_{\tau}^{t_k} a(s)\, ds\right) \mathcal{T}\exp\Big(\int_{\tau}^{t_k} l(s)\, ds\Big)\, b(\tau)\, x(\tau)\, \overline{\mathcal{T}}\exp\Big(\int_{\tau}^{t_k} \bar{r}(s)\, ds\Big)\, d\tau \end{aligned} \tag{11}$$

where $\mathcal{T}\exp(\cdot)$, $\overline{\mathcal{T}}\exp(\cdot)$ denote the time-ordered exponential in forward and backward order.

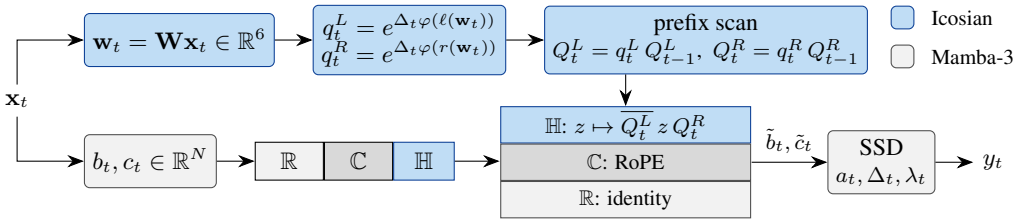


Figure 2: **Quaternionic state-space.** We implement Icosian as a generalization of Mamba-3 (Lahoti et al., 2026) by replacing a portion of its state-space by quaternions. Blue components are added by Icosian; grey components, including the SSD kernel, are inherited from Mamba-3.

With the time-ordered exponential as in Agrachev & Sachkov (2013); Heo et al. (2026). Equation 11 is the quaternionic analogue of the complex Equation 5. We discretize it in the same manner to obtain the discrete form of our system:

$$h_k = e^{a_k \Delta_k} e^{l_k \Delta_k}\, h_{k-1}\, e^{\bar{r}_k \Delta_k} + (1-\lambda_k)\,\Delta_k\, e^{a_k \Delta_k} e^{l_k \Delta_k}\, b_{k-1} x_{k-1}\, e^{\bar{r}_k \Delta_k} + \lambda_k\,\Delta_k\, b_k x_k \quad (12)$$

$$y_k = \mathrm{Re}(\bar{c}_k h_k) \quad (13)$$

3.3 Efficient Computation via the State-Space Duality

We now demonstrate that the quaternion actions on the state h_{k-1} in Equation 12 can equivalently be subsumed into the input and output transformations b_k, c_k, resulting in a recurrence whose only action on the state is multiplication by the scalar $e^{a_k \Delta_k}$. Such a reformulation of the recurrence has been referred to as the *RoPE trick* in the complex-valued case (Lahoti et al., 2026; Su et al., 2024). It enables the application of an efficient parallel algorithm for computing the discrete sequence of outputs, denoted as the *state-space duality* (SSD; Dao & Gu 2024).

Writing $q_k^L := \exp(l_k \Delta_k)$, $q_k^R := \exp(r_k \Delta_k)$ for the per-token quaternions of Equation 12 and defining $Q_k^L := q_k^L\, Q_{k-1}^L$, $Q_k^R := q_k^R\, Q_{k-1}^R$, with $Q_0^L = Q_0^R = 1$ for the accumulated quaternion prefixes, the scalar reformulation of the recurrence is given in Proposition 4.

Proposition 4 (Reformulated Scalar Recurrence)**.** *Define* $\tilde{h}_k := \overline{Q_k^L}\, h_k\, Q_k^R$, $\tilde{b}_k = \overline{Q_k^L}\, b_k\, Q_k^R$ *and* $\tilde{c}_k = \overline{Q_k^L}\, c_k\, Q_k^R$. *Then Equations 12 and 13 are equivalent to*

$$\tilde{h}_k = e^{a_k \Delta_k}\, \tilde{h}_{k-1} + (1-\lambda_k)\Delta_k\, e^{a_k \Delta_k}\, \tilde{b}_{k-1} x_{k-1} + \lambda_k \Delta_k\, \tilde{b}_k x_k, \quad (14)$$

$$y_k = \mathrm{Re}(\bar{\tilde{c}}_k\, \tilde{h}_k) = \tilde{\mathbf{c}}_k^\top \tilde{\mathbf{h}}_k. \quad (15)$$

Therefore, in the computational path, we first compute the cumulative quaternion products Q_k^L, Q_k^R, appropriately apply them to b_k, c_k, then execute the SSD algorithm on the scalar recurrence (14, 15).

3.4 Instantiation in a Neural Network – Icosian

Parametrizing the Quaternions. The quaternions l_k, r_k are generated as a function of the input $\mathbf{x}_k \in \mathbb{R}^d$. Firstly, a linear projection of the input generates 6 scalars per quaternionic block, the 6 degrees of freedom of $\mathrm{SO}(4)$ rotations, by $\mathbf{w}_k = \mathbf{W}\mathbf{x}_k \in \mathbb{R}^6$. We describe one block and write $\mathbf{w}_k \in \mathbb{R}^6$ for its six coordinates. We split $\mathbf{w}_k = (u_k, v_k) \in \mathbb{R}^3 \times \mathbb{R}^3$ and write $\ell(\mathbf{w}_k) = \frac{1}{2}(v_k - u_k)$ and $r(\mathbf{w}_k) = \frac{1}{2}(v_k + u_k)$, each identified with an imaginary quaternion.

A map φ below bounds the norm of each imaginary quaternion to $[0, \pi)$, the injectivity radius of the exponential (Equation 8; Grassia 1998; Lezcano-Casado & Martınez-Rubio 2019). The step size Δ_k then scales the result. Concretely, $\ell(\mathbf{w}_k)$ and $r(\mathbf{w}_k)$ are first processed by $\varphi(\eta) = \pi \tanh(\|\eta\|)\, \eta / \|\eta\|$. Thereafter, the quaternions are generated as:

$$q_k^L = \exp\big(\Delta_k\, \varphi(\ell(\mathbf{w}_k))\big), \quad q_k^R = \exp\big(\Delta_k\, \varphi(r(\mathbf{w}_k))\big), \quad \exp \eta = \cos\|\eta\| + \sin\|\eta\|\, \eta/\|\eta\| \quad (16)$$

Partitioning the State-Space. Concretely, we implement Icosian through the existing efficient approach of Mamba-3 (Lahoti et al., 2026). In particular, we partition the state size N as $N =$

$n_{\mathbb{R}} + 2n_{\mathbb{C}} + 4n_{\mathbb{H}}$, where $n_{\mathbb{F}}$ stands for the number of $\mathbb{F}$-valued blocks, $\mathbb{F} \in \{\mathbb{R}, \mathbb{C}, \mathbb{H}\}$. We compactly write the partition as $(f_{\mathbb{C}}, f_{\mathbb{H}}) = (2n_{\mathbb{C}}/N, 4n_{\mathbb{H}}/N)$, from which the fraction of the real-valued state is immediately inferred. For example, with $N = 128$, Mamba-3 uses its default $(\frac{1}{2}, 0)$ layout, i.e. 64 real, 32 complex, and 0 quaternionic elements. For $N = 128$, the usual value in our larger experiments, the default state partition of the Icosian is $(\frac{1}{2}, \frac{3}{32})$, i.e. 52 real dimensions, 32 complex dimensions, and 3 quaternionic ones, though we ablate this choice in Appendix B. Figure 2 shows the described partition. The remaining elements of the SSM are computed as usual (Appendix A).

3.5 Expressivity Bounds of One Icosian Layer

We now bound, from above and below, the expressivity of a single Icosian layer. We consider the setting of Icosian learning the structure of a finite group, following Liu et al. (2023); Merrill et al. (2024); Sarrof et al. (2024); Grazzi et al. (2025); Siems et al. (2025a); Walker et al. (2025); Shakerinava et al. (2026) among others. Detailed proofs are given in Appendix D.5.

Consider a finite group G. Throughout, a sequence of tokens is called a *word* $\omega = (\kappa_1, \dots, \kappa_T)$ with $\kappa_i \in \Sigma$, and Σ^* the (infinitely large) set of all finite-length words. The input alphabet is an encoding of a generating set of the group, i.e. $\Sigma \subseteq G$ with $\langle \Sigma \rangle = G$. The word ω is processed to obtain a sequence of embeddings $x(\kappa_1), \dots, x(\kappa_T)$ with $x(\kappa) \in \mathbb{R}^d$, forming the input of the SSM.

Without loss of generality, we consider an orthogonal discrete SSM[2]. recurrently processing words as $h_k = \rho(x(\kappa_k))\, h_{k-1}$. For Icosian, $\rho(x) \in \mathrm{SO}(4)^{n_{\mathbb{H}}}$ is the tuple of the block rotations $\mathcal{R}_{q^L, q^R}$ of Proposition 1. We write $\rho(\kappa) := \rho(x(\kappa))$ for the rotation induced by $\kappa \in \Sigma$. For a word $\omega = (\kappa_1, \dots, \kappa_T)$ we write the accumulated rotations as $\rho(\omega) := \rho(\kappa_T) \cdots \rho(\kappa_1)$ and the accumulated group elements as $g(\omega) := \kappa_T \cdots \kappa_1$. For our notion of *state tracking*, we require that they be consistent as per Definition 3:

Definition 3 (Tracking a Finite Group)**.** *Given a finite group G and a set of generators $\Sigma \subseteq G$, a discrete SSM $h_k = \rho(x(\kappa_k))\, h_{k-1}$ can track G if there exists a choice of weights and a mapping $\psi : \{\rho(\omega) : \omega \in \Sigma^*\} \to G$ such that $\{\rho(\omega) : \omega \in \Sigma^*\}$ is finite and $g(\omega) = \psi(\rho(\omega))$ for all $\omega \in \Sigma^*$.*

With this definition, our main results are summarized in Theorem 1, which places the expressivity of Icosian strictly beyond complex diagonal SSMs that cannot track non-solvable groups with any finite depth (Merrill et al., 2024; Shakerinava et al., 2026).

Theorem 1 (Expressivity Bounds of One Icosian Layer)**.** *Let a layer of Icosian have $n_{\mathbb{H}} \geq 1$ quaternionic blocks and width $d \geq 6n_{\mathbb{H}} + 1$. (i) The layer can track every finite subgroup of $\mathrm{SO}(4)$, including the non-solvable A_5. (ii) The layer cannot track S_5.*

4 Experimental Results

4.1 Language Modeling

We pretrain all baselines under the FineWeb data-ablation protocol (Penedo et al., 2024) at three scales, 0.5B, 1.1B and 3B, on 10.1B, 22.2B and 60.0B tokens respectively. For statistical robustness, the evaluations at the 0.5B and 1.1B parameter scales are performed with 5 seeds, while at the 0.5B multiple-input multiple-output (MIMO) and 3B scales they are are performed with 3 seeds. Additional details on the language experiments can be found in Appendix B.7.

Perplexity and QA tasks. Table 1 reports the validation perplexity and zero-shot accuracy across a standard range of 6 QA tasks: LAMBADA, HellaSwag, PIQA, ARC-e, ARC-c and Winogrande. At every scale Icosian's mean perplexity is below all baselines. Icosian also shows consistent improvement in the zero-shot suite, and its MIMO version ties Mamba-3.

Recall tasks. We further evaluate all trained models on 6 recall tasks (Arora et al., 2024): SWDE, SQuAD, FDA, TriviaQA, Natural Questions and DROP. We find that Icosian surpasses the baselines on these tasks as well (see Table 9).

[2]Orthogonality is fully general for groups: finite matrix groups are orthogonal in some basis.

Table 1: **Language Modeling.** Validation perplexity and zero-shot accuracy of Icosian trained on a subset of FineWeb, mean over seeds with the standard deviation over seeds in parentheses. Five seeds at each scale except 3B. Best results in bold.

Model	PPL ↓	LAMB. PPL ↓	LAMB. ↑	PIQA ↑	Hella. ↑	Wino. ↑	ARC-e ↑	ARC-c ↑	Avg. ↑
0.5B									
GDN $[-1,1]$	21.77(0.05)	30.70(0.87)	32.6(0.7)	**68.2(0.8)**	37.4(0.2)	51.1(0.6)	46.7(1.0)	**24.2(0.7)**	43.4(0.4)
GDP $[-1,1]$	21.83(0.01)	30.56(1.55)	33.4(0.8)	67.6(0.5)	37.3(0.2)	**51.6(1.1)**	46.1(0.7)	24.1(0.8)	43.4(0.5)
Mamba-3	21.76(0.05)	26.92(1.70)	35.5(1.2)	67.5(0.3)	37.2(0.1)	50.5(0.7)	**47.0(0.3)**	23.9(0.5)	43.6(0.2)
Icosian	**21.73(0.02)**	**26.75(0.38)**	**35.6(0.4)**	67.7(0.3)	**37.5(0.2)**	51.1(0.8)	**47.0(0.5)**	**24.2(0.2)**	**43.9(0.2)**
0.5B MIMO									
Mamba-3	21.36(0.03)	**24.40(0.98)**	**37.0(0.5)**	67.7(0.9)	**38.0(0.3)**	**51.8(1.3)**	47.1(1.0)	**24.3(0.3)**	**44.3(0.2)**
Icosian	**21.31(0.04)**	24.89(1.22)	36.7(0.7)	**68.4(0.5)**	37.8(0.2)	51.4(0.7)	**47.2(0.8)**	24.1(0.5)	**44.3(0.2)**
1.1B									
GDN $[-1,1]$	16.10(0.05)	13.24(0.66)	45.2(0.8)	71.9(0.3)	49.4(0.2)	52.8(1.1)	54.5(1.0)	26.2(0.6)	50.0(0.4)
GDP $[-1,1]$	16.18(0.05)	13.37(0.42)	44.6(0.9)	71.5(0.9)	48.9(0.3)	52.6(0.8)	53.9(0.7)	25.8(0.8)	49.5(0.3)
Mamba-3	16.07(0.03)	12.37(0.38)	46.2(0.8)	**72.0(0.4)**	**49.7(0.2)**	**52.9(1.3)**	**54.9(1.2)**	26.0(0.4)	50.3(0.3)
Icosian	**16.03(0.03)**	**12.10(0.25)**	**46.9(0.8)**	**72.0(0.3)**	49.6(0.2)	52.8(0.6)	54.8(0.6)	**27.1(0.8)**	**50.5(0.1)**
3B									
Mamba-3	12.27(0.02)	6.95(0.20)	56.6(0.6)	75.6(0.4)	63.8(0.1)	58.6(1.2)	**65.3(0.1)**	**32.6(0.3)**	58.7(0.2)
Icosian	**12.25(0.04)**	**6.80(0.07)**	**57.0(0.5)**	**75.9(0.2)**	**64.2(0.4)**	**59.0(0.4)**	64.9(0.6)	31.7(0.6)	**58.8(0.1)**

4.2 State Tracking and Mechanistic Evaluation

Synthetic State Tracking. We consider the finite groups $Q_8, D_8, A_5,$ and S_5 (Carter, 2009). The first two are non-commutative solvable groups, whereas the last two are additionally non-solvable. The experimental setup follows prior work (Terzic et al., 2025) and mimics the setup of Definition 3; the input sequence consists of one-hot embeddings indexing group elements, and the models are trained to predict the composition of the selected elements. The setup adopts the standard length-generalization protocol (Walker et al., 2025), i.e. single-layer models are trained to predict the composition of group actions on sequences up to length 40, and are evaluated on lengths $40 - 256$. This protocol excludes length-dependent *shortcut* solutions (Liu et al., 2023). Section 3.5 predicts that Icosian should be able to model each group except S_5. As we can see in Table 2, the theoretical prediction is accurately reflected by experimental data. One Icosian layer solves A_5 and D_8 and reaches 92% on Q_8, while Mamba-3 (Lahoti et al., 2026) and Gated DeltaNet (GDN) $[-1,1]$ (Yang et al., 2025; Grazzi et al., 2025) fail to accurately model any of the four groups. Gated DeltaProduct (GDP) (Siems et al., 2025a) reaches 95% on A_5. Flash PD-SSM (Terzić et al., 2026) solves Q_8, D_8 and A_5 and partly S_5, which no other model solves. In Appendix B.2 we extract from the trained weights on A_5 the representation of Figure 1c, corresponding to the construction of Theorem 1.

Table 2: **Results of synthetic state tracking.** Best validation accuracy (%) on lengths 40 to 256 after training up to 40 (Terzic et al., 2025), mean $\pm$ sd over five seeds. Best results in bold.

Model	$Q_8 \uparrow$	$D_8 \uparrow$	$A_5 \uparrow$	$S_5 \uparrow$
Chance (%)	12.5	12.5	1.67	0.83
Discrete SSMs				
Flash PD-SSM	$\mathbf{100.0 \pm 0.1}$	$\mathbf{100.0 \pm 0.0}$	$\mathbf{100.0 \pm 0.0}$	$\mathbf{67.1 \pm 44.7}$
Continuous SSMs				
GDN $[-1,1]$	52.4 ± 0.4	59.0 ± 2.7	15.9 ± 0.6	8.2 ± 0.7
GDP $[-1,1]$, $n_h = 2$	54.2 ± 0.7	88.7 ± 14.0	94.7 ± 7.1	25.3 ± 4.1
Mamba-3	37.0 ± 1.6	39.1 ± 4.8	7.9 ± 0.0	6.1 ± 0.4
Icosian	92.1 ± 15.9	99.6 ± 0.5	$\mathbf{100.0 \pm 0.0}$	12.3 ± 2.8

State Tracking in Natural Language. We now test the capabilities of Icosian on a variant of the previous task in which the generators of A_5 are mapped to disjoint sets of english sentences (Terzic et al., 2025), with the goal of emulating a more realistic world-model scenario. Concretely, while a group element was previously associated with a one-hot encoding thereof, each element is now instead associated with a semantically related group of sentences in English. After processing a sentence, the model must internally execute the action of the corresponding group element on the hidden state (details in Appendix B.4). First, the input sequence is processed by a frozen Qwen 2.5-1.5B

model (Qwen et al., 2025), and the resulting embeddings are fed through a linear map into a single SSM layer, which is trained to predict the composed group element. The models are trained with a length curriculum to 25 sentences and evaluated at lengths up to 80. As shown in Figure 3 Icosian matches Mamba-3 up to ten sentences but maintains the high accuracy for longer (Appendix B.4). We also evaluate both models on a natural language version of the parity automaton (Terzic et al., 2025). On this task, both models correctly generalize, as expected.

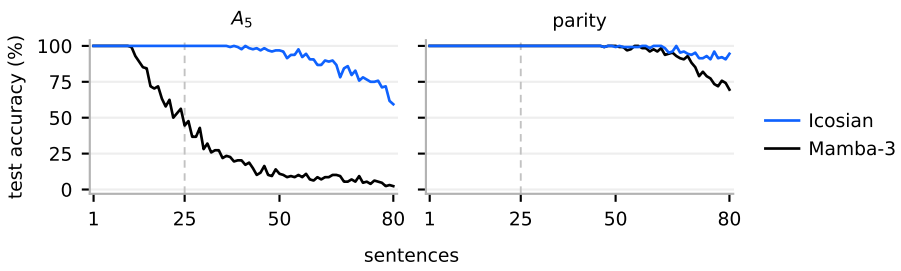


Figure 3: **State Tracking in Natural Language.** Test accuracy against length on A_5 and on the parity control (Terzic et al., 2025), maximum over five seeds. The dashed line is the training maximum.

Tracking Continuous Rotations. Finally, we evaluate Icosian on an extended version of the rigid-body rotation task (Heo et al., 2026). In the original task, the continuous embedding of A_5 in $\mathrm{SO}(3)$ is used to define actions on a random unit vector in $\mathbb{R}^3$. The models are trained to compose A_5 rotations, interpreted as elements of $\mathrm{SO}(3)$, on the randomly drawn unit vector. We extend the task by adding continuous rotations drawn uniformly from $\mathrm{SO}(3)$ and from $\mathrm{SO}(4)$ (see Appendix B.3). Figure 4 reports all three versions, with every model in its best configuration (see Figure 8 for all the configurations). Icosian achieves the best performance on all benchmarks, with GDP the second best on the continuous rotations. Interestingly, while GDN is second best on the discrete task, it underperforms on the continuous ones, corroborating the different nature of the two settings.

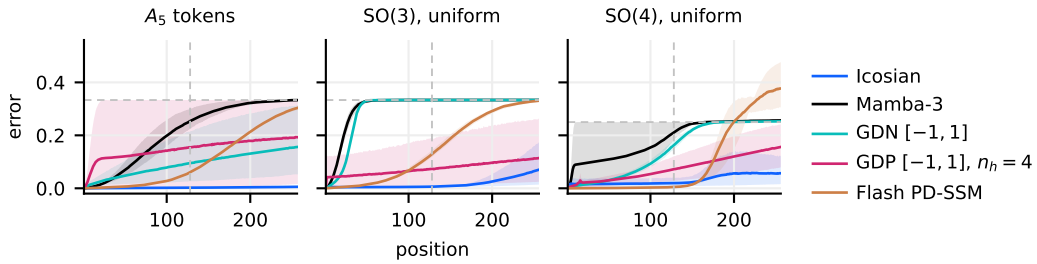


Figure 4: **Rigid-Body Rotations.** Error against position on the A_5 task (Heo et al., 2026) and its continuous extensions, at length 256 after training at 128, mean over seeds with range shaded, each model at its best configuration. Dashed lines: chance and training length.

Mechanistic Architecture Design (MAD). The Mechanistic Architecture Design (MAD) benchmark (Poli et al., 2024) defines 6 small-scale synthetic tasks, high performance on which has been correlated with high performance of compute-optimal trained hybrid models. The benchmark mainly probes the models' capabilities to perform in-context recall, memorization, and other related tasks. The performance on the six tasks is reported in Table 6 in Appendix B.5, with Icosian exhibiting the highest average performance against all baselines. This indicates that Icosian is a strong candidate layer for language modeling, which we validate in Section 4.1.

Chomsky Hierarchy. Finally, we compare Icosian with the baselines on the Chomsky-Hierarchy finite state automata tasks (Walker et al., 2025). Here, Icosian notably outperforms Mamba-3, and matches or outperforms GDN and GDP in some configurations (see Table 7).

5 Related Work

SSMs. Gu et al. (2022) reparametrize the linear RNN as a discretized continuous-time system, giving the structured SSM its long-range memory without attention. Orvieto et al. (2023); Gupta et al. (2022) demonstrate the effectiveness of real and complex-valued diagonal SSMs with fixed weights. Gu & Dao (2024) make the SSM weights data-dependent at every step, yielding Mamba's selective recurrence; Dao & Gu (2024) recast this recurrence as a form of masked attention, the state-space duality, and derive from it the chunked, hardware-efficient algorithm that both Mamba-3 (Lahoti et al., 2026) and Icosian reuse. In parallel, Schlag et al. (2021) recover the delta rule as an associative fast-weight update within linear attention, Yang et al. (2024) give it a chunkwise-parallel form (DeltaNet) that scales to modern hardware, and Yang et al. (2025) add an input-dependent forget gate to the same recurrence (Gated DeltaNet).

State Tracking with SSMs. Merrill et al. (2024) show that diagonal SSMs fall in the TC^0 complexity class and thus cannot compose arbitrary permutations. Sarrof et al. (2024) characterize the regular languages representable by diagonal SSMs as the star-free ones. Grazzi et al. (2025) recover parity and further state tracking in linear recurrent neural networks (i.e. SSMs) by allowing negative eigenvalues in the recurrence. Siems et al. (2025b) build on the DeltaNet approach and extend its single Householder transition matrix to a product of Householder reflections per token to increase state-tracking capability. Shakerinava et al. (2026) prove that a diagonal complex recurrence tracks exactly the finite abelian groups. Icosian keeps the diagonal scan and places the non-abelian part in the quaternionic blocks.

Unitary, Orthogonal and Quaternion Recurrences. Unitary evolution RNNs (Arjovsky et al., 2016) and their full-capacity (Wisdom et al., 2016), tunable (Jing et al., 2017), Householder (Mhammedi et al., 2017), Cayley (Helfrich et al., 2018) and exponential-map (Lezcano-Casado & Martınez-Rubio, 2019) parametrizations keep the recurrence matrix in a compact group to control the gradient norm. Their transition matrix is one learned weight shared across time steps, and the hidden state passes through a nonlinearity. Quaternion RNNs (Parcollet et al., 2019) give the weights and the hidden state quaternion values and multiply them with the Hamilton product, again with a weight shared across time steps. In Icosian the rotation is chosen by the input at every step, and the recurrence stays linear so that it remains amenable to parallel computation.

6 Conclusion

State-space models increasingly rely on richer transition dynamics to improve state tracking and language modeling without sacrificing their computational advantages. In this work, we introduce Icosian, a quaternionic SSM that leverages the *exceptional* (or *accidental*) structure of $\mathrm{SO}(4)$ to efficiently implement input-dependent, non-commutative 4D rotations. Importantly, 4D rotations represent a natural limit of this approach: $\mathrm{SO}(4)$ is the highest-dimensional rotation group whose rotations can be accumulated through a compact algebraic representation more efficiently than through generic rotation matrices. Across language models ranging from 500M to 3B parameters, Icosian consistently improves upon Mamba-3, Gated DeltaNet, and Gated DeltaProduct. In addition, we develop an optimized implementation that accumulates the quaternionic rotations through efficient prefix scans before the adapted SSD kernel, limiting the additional computational cost over Mamba-3 to only 3% at the 3B scale. We further characterize the state dynamics of Icosian theoretically and experimentally. A single Icosian layer length-generalizes on the non-solvable group A_5 and directly learns a representation of the group, while deeper models accurately track continuous rotations in $\mathrm{SO}(3)$ and $\mathrm{SO}(4)$. Together, these results show that quaternionic transitions provide a practical approach to incorporating richer, non-commutative dynamics into SSMs while retaining the efficiency required for language modeling at scale.

Limitations. Our theoretical analysis primarily characterizes a single Icosian layer and does not establish how its expressivity evolves with depth. In particular, while a single layer cannot track S_5 and two-layer models fail to do so empirically, whether deeper architectures can represent such dynamics remains open.

AI USE STATEMENT

In this work, we used generative AI tools to edit and polish the text of the paper, to obtain initial drafts of the proofs, and to help write the efficient kernel code and evaluation suites. We did not use generative AI tools to generate the core method or the trained models, which were designed and implemented by the authors. We have reviewed all AI-assisted work: the code was run and its outputs checked by the authors, and the text and figures were revised by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

All research conducted conforms with the Code of Ethics. There are many potential consequences of work regarding LLMs, however, there is no specific ethics impact deriving from this work.

REPRODUCIBILITY STATEMENT

All datasets used are publicly available. The architecture, the training setup and the hyperparameters of every benchmark are in Appendix B. As indicated by ICLR's author guidelines, we are available to share the code anonymously during the rebuttal period.

REFERENCES

Andrei A Agrachev and Yuri Sachkov. *Control theory from the geometric viewpoint*. Springer Science & Business Media, 2013.

Martin Arjovsky, Amar Shah, and Yoshua Bengio. Unitary evolution recurrent neural networks. In *International Conference on Machine Learning*, pp. 1120–1128, 2016.

Simran Arora, Aman Timalsina, Aaryan Singhal, Sabri Eyuboglu, Xinyi Zhao, Ashish Rao, Atri Rudra, and Christopher Re. Just read twice: closing the recall gap for recurrent language models. In *Workshop on Efficient Systems for Foundation Models II @ ICML2024*, 2024. URL https://openreview.net/forum?id=RsqCqziLAt.

George M Bergman. On semigroups that are prime in the sense of tarski, and groups prime in the senses of tarski and of rhodes. *Communications in Algebra*, 54(7):2773–2795, 2026.

Guy E Blelloch. Prefix Sums and Their Applications, 1990.

Nathan C. Carter. *Visual group theory*. Classroom resource materials. Mathematical Association of America, Washington, D.C., 2009.

Bilal Chughtai, Lawrence Chan, and Neel Nanda. A toy model of universality: Reverse engineering how networks learn group operations. In *International Conference on Machine Learning*, pp. 6243–6267. PMLR, 2023.

John H Conway and Derek A Smith. *On quaternions and octonions*. CRC Press, 2003.

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. URL https://openreview.net/forum?id=ztn8FCR1td.

Grégoire Delétang, Anian Ruoss, Jordi Grau-Moya, Tim Genewein, Li Kevin Wenliang, Elliot Catt, Chris Cundy, Marcus Hutter, Shane Legg, Joel Veness, et al. Neural networks and the chomsky hierarchy. In *International Conference on Learning Representations*, 2023.

David S. Dummit and Richard M. Foote. *Abstract Algebra*. Wiley, 3 edition, 2004.

F. Sebastin Grassia. Practical parameterization of rotations using the exponential map. *J. Graph. Tools*, 3(3):29–48, March 1998. ISSN 1086-7651. doi: 10.1080/10867651.1998.10487493. URL https://doi.org/10.1080/10867651.1998.10487493.

Riccardo Grazzi, Julien Siems, Arber Zela, Jörg Franke, Frank Hutter, et al. Unlocking state-tracking in linear rnns through negative eigenvalues. In *International Conference on Learning Representations*, volume 2025, pp. 36565–36597, 2025.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=tEYskw1VY2.

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.

Ankit Gupta, Albert Gu, and Jonathan Berant. Diagonal state spaces are as effective as structured state spaces. *Advances in neural information processing systems*, 35:22982–22994, 2022.

A. Hatcher. *Algebraic Topology*. Algebraic Topology. Cambridge University Press, 2002. ISBN 9780521795401. URL https://books.google.ch/books?id=BjKs86kosqgC.

Kyle Helfrich, Devin Willmott, and Qiang Ye. Orthogonal recurrent neural networks with scaled Cayley transform. In *International Conference on Machine Learning*, pp. 1969–1978, 2018.

Gyuryang Heo, Timothy Ngotiaoco, Kazuki Irie, Samuel J Gershman, and Bernardo L Sabatini. Why depth matters in parallelizable sequence models: A lie algebraic view. *arXiv preprint arXiv:2603.05573*, 2026.

Du Q. Huynh. Metrics for 3D rotations: comparison and analysis. *Journal of Mathematical Imaging and Vision*, 35(2):155–164, 2009.

Li Jing, Yichen Shen, Tena Dubček, John Peurifoy, Scott Skirlo, Yann LeCun, Max Tegmark, and Marin Soljačić. Tunable efficient unitary neural networks (EUNN) and their application to RNNs. In *International Conference on Machine Learning*, pp. 1733–1741, 2017.

Kenneth Krohn and John Rhodes. Algebraic theory of machines. i. prime decomposition theorem for finite semigroups and machines. *Transactions of the American Mathematical Society*, 116: 450–464, 1965.

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. URL https://openreview.net/forum?id=HwCvaJOiCj.

Mario Lezcano-Casado and David Martınez-Rubio. Cheap orthogonal constraints in neural networks: A simple parametrization of the orthogonal and unitary group. In *International conference on machine learning*, pp. 3794–3803. PMLR, 2019.

Bingbin Liu, Jordan T. Ash, Surbhi Goel, Akshay Krishnamurthy, and Cyril Zhang. Transformers learn shortcuts to automata. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=De4FYqjFueZ.

Pertti Lounesto. *Clifford Algebras and Spinors*. London Mathematical Society Lecture Note Series. Cambridge University Press, 2 edition, 2001.

Henry B Mann and Donald R Whitney. On a test of whether one of two random variables is stochastically larger than the other. *The annals of mathematical statistics*, pp. 50–60, 1947.

Giovanni Luca Marchetti, Daniel Kunin, Adele Myers, Francisco Acosta, and Nina Miolane. Sequential group composition: a window into the mechanics of deep learning. In *International Conference on Machine Learning*, 2026.

John Meier. *Groups, Graphs and Trees: An Introduction to the Geometry of Infinite Groups*. Cambridge University Press, 2008.

William Merrill, Jackson Petty, and Ashish Sabharwal. The illusion of state in state-space models. In *International Conference on Machine Learning (ICML)*, 2024. URL https://openreview.net/forum?id=QZgo9JZpLq.

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, 2017.

Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. Progress measures for grokking via mechanistic interpretability. In *International Conference on Learning Representations*, 2023.

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*, pp. 26670–26698. PMLR, 2023.

Titouan Parcollet, Mirco Ravanelli, Mohamed Morchid, Georges Linarès, Chiheb Trabelsi, Renato De Mori, and Yoshua Bengio. Quaternion recurrent neural networks. In *International Conference on Learning Representations*, 2019.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. The fineweb datasets: Decanting the web for the finest text data at scale. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. URL https://openreview.net/forum?id=n6SCkn2QaG.

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. URL https://openreview.net/forum?id=ayB1PACN5j.

Michael Poli, Armin W Thomas, Eric Nguyen, Pragaash Ponnusamy, Björn Deiseroth, Kristian Kersting, Taiji Suzuki, Brian Hie, Stefano Ermon, Christopher Re, Ce Zhang, and Stefano Massaroli. Mechanistic design and scaling of hybrid architectures. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=GDp7Gyd9nf.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

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*, volume 37, 2024.

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.

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 *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025a. URL https://openreview.net/forum?id=SoRiaijTGr.

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*, volume 38, pp. 153738–153782, 2025b.

Dashiell Stander, Qinan Yu, Honglu Fan, and Stella Biderman. Grokking group multiplication with cosets. In *International Conference on Machine Learning*, 2024.

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

Aleksandar Terzic, Nicolas Menet, Michael Hersche, Thomas Hofmann, and Abbas Rahimi. Structured sparse transition matrices to enable state tracking in state-space models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=RDbuSCWhad.

Aleksandar Terzić, Francesco Carzaniga, Nicolas Menet, Yannick Biehl, Michael Hersche, Thomas Hofmann, and Abbas Rahimi. Flash pd-ssm: Memory-optimized structured sparse state-space models. *arXiv preprint arXiv:2605.19150*, 2026.

Benjamin Walker, Lingyi Yang, Nicola Muca Cirone, Cristopher Salvi, and Terry Lyons. Structured linear CDEs: Maximally expressive and parallel-in-time sequence models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=HKDyRDzy1E.

Frank Wilcoxon. Individual comparisons by ranking methods. In *Breakthroughs in statistics: Methodology and distribution*, pp. 196–202. Springer, 1992.

Scott Wisdom, Thomas Powers, John Hershey, Jonathan Le Roux, and Les Atlas. Full-capacity unitary recurrent neural networks. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

Songlin Yang, Bailin Wang, Yu Zhang, Yikang Shen, and Yoon Kim. Parallelizing linear transformers with the delta rule over sequence length. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2024. URL https://openreview.net/forum?id=y8Rm4VNRPH.

Songlin Yang, Jan Kautz, and Ali Hatamizadeh. Gated delta networks: Improving mamba2 with delta rule. In *International Conference on Learning Representations*, volume 2025, pp. 29687–29707, 2025.

A Further Details of the Parametrization

Aside from the added quaternion parameters, the remaining functions of the state-space model are given by:

$$a_k = -\text{softplus}\big(\mathbf{w}_a^\top \mathbf{x}_k\big) \in \mathbb{R}_{<0}, \qquad \theta_k = \mathbf{w}_\theta^\top \mathbf{x}_k \in \mathbb{R},$$
$$\Delta_k = \text{softplus}\big(\mathbf{w}_\Delta^\top \mathbf{x}_k + \beta_\Delta\big) \in \mathbb{R}_{>0}, \qquad \lambda_k = \sigma\big(\mathbf{w}_\lambda^\top \mathbf{x}_k\big) \in [0,1],$$
$$x_k = \mathbf{w}_x^\top \mathbf{x}_k \in \mathbb{R}, \qquad \mathbf{b}_k, \mathbf{c}_k = \text{RMSNorm}(\mathbf{W}_{b,c}\mathbf{x}_k) + \beta_{b,c} \in \mathbb{R}^N.$$

Furthermore, we reuse the standard layer construction of Mamba-3, with details provided in (Lahoti et al., 2026).

B Experimental Details and Additional Results

We describe one benchmark per subsection, in the presented order in Section 4.

B.1 Conventions

State Layouts. As detailed in Section 3.4, the *layout* is the pair $(f_{\mathbb{C}}, f_{\mathbb{H}})$ that determines the fraction of the default $N = 128$ rows of the state in complex planes and the fraction in quaternion blocks. The remaining rows are real. Concretely, $(\frac{1}{2}, 0)$ is Mamba-3 with thirty-two complex planes and no block. The default $(\frac{1}{2}, \frac{3}{32})$ adds three quaternion blocks and is written Icosian wherever a table shows one layout. The layout $(\frac{1}{2}, \frac{1}{4})$ adds eight quaternion blocks, $(\frac{1}{4}, \frac{1}{4})$ has sixteen complex planes and eight quaternion blocks, and $(0, \frac{1}{2})$ has sixteen quaternion blocks and no complex plane. The MAD benchmark (Poli et al., 2024) runs at $N = 16$, where one block is a quarter of the state (Appendix B.5).

Tables and Statistical Test. In each table, every entry is the mean over seeds with the between-seed standard deviation, with the seed count given in the caption or in the section of the benchmark. Cited rows, marked in the caption or by the header *Cited*, are numbers published by a benchmark's authors. For testing statistical significance in Table 3, we use the exact two-sided Wilcoxon rank-sum test (Wilcoxon, 1992; Mann & Whitney, 1947). It compares a row's per-seed values with Mamba-3, taken as the baseline. Its smallest attainable p is $2/\binom{n+m}{n}$, which is 0.0079 at five seeds a side. With three seeds against three or four it cannot reach 0.05, and the table shows a dash. Bold marks the best mean of a group among our runs.

Flash PD-SSM. The model has a gated block and a simplified architecture without gates and convolution, each real- or complex-valued (Terzić et al., 2026). We report the best form we ran per benchmark. On the group word problems and the Chomsky suite this is the complex simplified architecture. On the rotation tasks it is the gated form with the identity dictionary initialization. On MAD it is the real-valued gated form of the model's own MAD experiments (Terzić et al., 2026). There is no language row, since the model is proposed as a hybrid component (Terzić et al., 2026).

B.2 Group Word Problems

We follow the PD-SSM protocol (Terzic et al., 2025) on the group automata (Liu et al., 2023; Walker et al., 2025) Concretely, we train one layer of width 1024 and state 128 with Adam without weight decay, at batch 256 for 100k steps on lengths 2 to 40. We validate on $8{,}192$ words of lengths 40 to 256. The learning rate is selected per model and group on seeds 0 to 2, by the three-seed mean, over the grid $\{1, 5, 10, 50, 100\}\times 10^{-4}$ along with the suite's default value (Walker et al., 2025), 2×10^{-3}. Seeds 3 to 7 are then trained at the selected rate and reported as mean $\pm$ sd of the best validation accuracy. GDN and GDP are parameter-matched within one percent, and Flash PD-SSM runs at 8 heads of state 128 and dictionary size 32. Table 4 repeats Table 2 for every Icosian state layout.

The Defect of the Token Quaternions. Theorem 1 says that the token quaternions of one quaternion block can realize a homomorphism of A_5 into $\text{SO}(3)$. Table 2 reports that an Icosian layer can

Table 3: **Seed-Level Tests.** Exact two-sided Wilcoxon rank-sum p of each row's per-seed values against Mamba-3's (Appendix B.1) for Tables 1, 8 and 10. A dash marks three seeds a side, where $p < 0.05$ is unattainable.

$(f_{\mathbb{C}}, f_{\mathbb{H}})$	Model	Perplexity	Zero-shot Avg.	Recall Avg.
0.5*B*				
	GDN $[-1, 1]$	0.84	0.42	0.0079
	GDP $[-1, 1]$, 2 refl.	0.0079	0.55	0.0079
$(\frac{1}{2}, \frac{3}{32})$	Icosian	0.31	0.15	0.22
$(\frac{1}{2}, \frac{1}{4})$	Icosian	0.15	0.22	0.84
$(\frac{1}{4}, \frac{1}{4})$	Icosian	0.31	0.0079	0.69
0.5*B MIMO*				
$(\frac{1}{2}, \frac{3}{32})$	Icosian, MIMO	0.095	0.84	0.31
$(\frac{1}{2}, \frac{1}{4})$	Icosian, MIMO	0.095	0.42	1.00
$(\frac{1}{4}, \frac{1}{4})$	Icosian, MIMO	0.0079	0.22	0.84
1.1*B*				
	GDN $[-1, 1]$	0.55	0.15	0.0079
	GDP $[-1, 1]$, 2 refl.	0.0079	0.0079	0.016
$(\frac{1}{2}, \frac{3}{32})$	Icosian	0.032	0.42	0.55
$(\frac{1}{2}, \frac{1}{4})$	Icosian	0.095	0.55	0.42
$(\frac{1}{4}, \frac{1}{4})$	Icosian	0.016	0.15	0.31
3*B*				
$(\frac{1}{2}, \frac{3}{32})$	Icosian	–	–	–

FineWeb loss by position bucket, Icosian$(\frac{1}{2}, \frac{3}{32})$ *against Mamba-3*

	0–2048	2048–4096	4096–8192	8192–16384	16384–32768
0.5B	0.69	0.69	0.31	1.00	0.42
1.1B	0.016	0.056	0.056	0.0079	0.032
3B	–	–	–	–	–

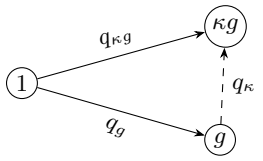


Figure 5: **The Defect on One Edge.** The tree reaches g and κg from the identity (solid), and the token κ leads from g to κg (dashed). The defect compares the two ways of arriving at κg: $q_\kappa q_g$ along the edge and $q_{\kappa g}$ along the tree. The number δ is the largest such distance over all g and κ.

learn to solve the task. We ask how distant the quaternions the trained layer assigns to the tokens are from a homomorphism of the group into $\mathrm{SO}(3)$. We require a single number that is zero exactly for homomorphisms and increases with the distance from a homomorphism. We call this number the *defect*. Extracting a homomorphism of the group from trained weights has precedents in the mechanistic studies of group composition in feed-forward networks. There, the learned features have been read as the irreducible representations of the group (Nanda et al., 2023; Chughtai et al., 2023; Marchetti et al., 2026) or as cosets of its subgroups (Stander et al., 2024). Here, the representation is a rotation representation carried by one quaternion block, and we measure it directly.

Conventions. We reuse the notions of tokens, words and the element $g(\omega)$ of a word from Section 3.5. Each token κ is assigned the pair $q^L(\kappa), q^R(\kappa)$ (Equation 16) with the step size $\Delta(\mathbf{x}(\kappa))$. We write q_κ for either of the left or right quaternion, as the construction only requires one of them. Two unit quaternions are compared by

$$d(q, q') := \tfrac{1}{2} \min_{\pm} \|q \mp q'\|,$$

known as the chordal metric on $S^3/\{\pm 1\} \cong \mathrm{SO}(3)$. It is one half of the standard quaternion distance (Huynh, 2009). It is a sine of a quarter angle: for unit quaternions $\|q \mp q'\|^2 = 2 \mp 2\langle q, q'\rangle$, and the rotation $\mathcal{R}_{q',q'}^{-1} \mathcal{R}_{q,q}$ has angle $\alpha \in [0, \pi]$ with $\cos(\alpha/2) = |\langle q, q'\rangle|$ (Conway & Smith, 2003), so $d(q, q') = \sin(\alpha/4)$. We report every distance as this angle $4 \arcsin d$.

Table 4: **Every Icosian layout on the group word problems.** Extended version of Table 2 showing all considered Icosian state layouts.

$(f_{\mathbb{C}}, f_{\mathbb{H}})$	Model chance (%)	$Q_8 \uparrow$ 12.5	$D_8 \uparrow$ 12.5	$A_5 \uparrow$ 1.67	$S_5 \uparrow$ 0.83
Discrete SSMs					
	Flash PD-SSM	$\mathbf{100.0 \pm 0.1}$	$\mathbf{100.0 \pm 0.0}$	$\mathbf{100.0 \pm 0.0}$	$\mathbf{67.1 \pm 44.7}$
Continuous SSMs					
	GDN $[-1, 1]$	52.4 ± 0.4	59.0 ± 2.7	15.9 ± 0.6	8.2 ± 0.7
	GDP $[-1, 1]$, 2 refl.	54.2 ± 0.7	88.7 ± 14.0	94.7 ± 7.1	25.3 ± 4.1
$(\frac{1}{2}, 0)$	Mamba-3	37.0 ± 1.6	39.1 ± 4.8	7.9 ± 0.0	6.1 ± 0.4
$(\frac{1}{2}, \frac{3}{32})$	Icosian	92.1 ± 15.9	99.6 ± 0.5	$\mathbf{100.0 \pm 0.0}$	12.3 ± 2.8
$(\frac{1}{2}, \frac{1}{4})$	Icosian	86.4 ± 16.1	$\mathbf{100.0 \pm 0.0}$	$\mathbf{100.0 \pm 0.0}$	15.0 ± 2.9
$(\frac{1}{4}, \frac{1}{4})$	Icosian	93.3 ± 13.5	99.8 ± 0.5	$\mathbf{100.0 \pm 0.0}$	17.7 ± 4.0
$(0, \frac{1}{2})$	Icosian	98.2 ± 3.9	$\mathbf{100.0 \pm 0.0}$	$\mathbf{100.0 \pm 0.0}$	16.7 ± 3.9

Tree Words and the Defect. For a word let $q(\omega) := q_{\kappa_k} \cdots q_{\kappa_1}$, the product in the order of the scan. A breadth-first search of the Cayley graph of G over Σ (Meier, 2008, Chapter 1) fixes one shortest word ω_g for every element, the *tree word* of g. We set $q_g := q(\omega_g)$. Then $q_g = 1$ for the identity, and $q_{\kappa g} = q_\kappa q_g$ along the tree.

Definition 4 (Defect of the Token Quaternions)**.** *Let $q_\kappa \in S^3$ be given for every token $\kappa \in \Sigma$, and let $q_g := q(\omega_g)$ for every $g \in G$ along the tree words. We call the largest distance, over all elements $g \in G$ and tokens $\kappa \in \Sigma$, between the product $q_\kappa q_g$ and the quaternion $q_{\kappa g}$ assigned to the product element the **defect** of the token quaternions (Figure 5),*

$$\delta \;:=\; \max_{g \in G,\ \kappa \in \Sigma} d(q_\kappa\, q_g,\ q_{\kappa g}). \tag{17}$$

Proposition 5 (The Defect Vanishes Exactly on Homomorphisms)**.** *The defect is zero if and only if some homomorphism $G \to \mathrm{SO}(3)$ sends every token κ to the rotation $\mathcal{R}_{q_\kappa, q_\kappa}$. That homomorphism is $g \mapsto \mathcal{R}_{q_g, q_g}$.*

Proof. Such a homomorphism sends g to $\mathcal{R}_{q_g,q_g}$, since q_g is the product of the token quaternions along the tree word of g, so it satisfies every edge. Conversely let $\delta = 0$, so $q_{\kappa g} = \pm q_\kappa q_g$ for every element g and token κ. Let $a, b \in G$ and write $a = \kappa_k \cdots \kappa_1$ along its tree word. Applying the edge relation k times to b and then to the identity gives $q_{ab} = \pm\, q_{\kappa_k} \cdots q_{\kappa_1}\, q_b = \pm\, q_a\, q_b$. The tree word of a token $\kappa \neq 1$ is κ itself, and for $\kappa = 1$ the edge at $g = 1$ gives $q_\kappa = \pm 1$, so $g \mapsto \mathcal{R}_{q_g,q_g}$ is a homomorphism that sends every token κ to $\mathcal{R}_{q_\kappa,q_\kappa}$. □

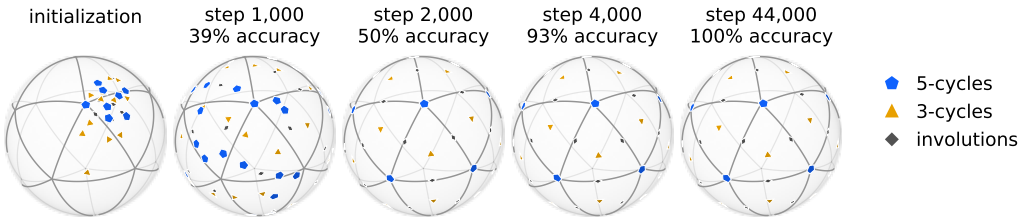
A map of defect zero is thus a homomorphism, and a map of small defect satisfies every edge of the Cayley graph up to that angle. Random unit quaternions give a defect near $178°$.

The Measurement Procedure. For every quaternion block of the layer, and for the left and the right quaternions, we form q_g along the tree words and compute δ. We report the block whose $|G|$ quaternions are pairwise distinct modulo sign, no two within $23°$ of each other, and whose defect is smallest. For Q_8, where -1 must differ from 1, d is taken without the minimum over $\pm$ and the homomorphism is into S^3. Figures 1c and 7 draw, for each element, the axis and angle of $\mathcal{R}_{q_g,q_g}$, in a frame with a five-fold axis at the pole.

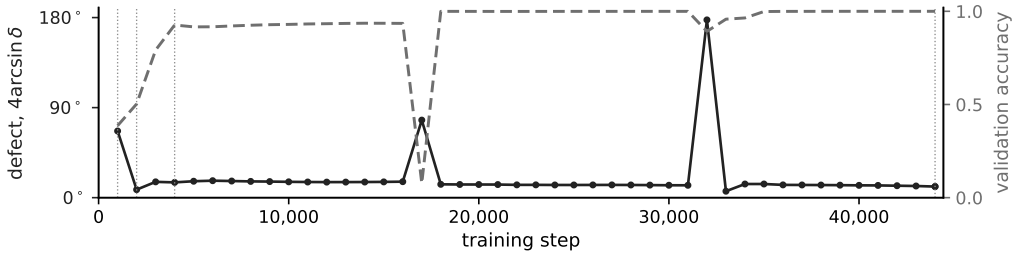
Measured Values. Over the reported seeds on A_5 the sixty quaternions are distinct, and the defect is at most $14.7°$. Over all words of length at most six, the product the scan forms lies within $14.9°$ of its element's quaternion. On S_5 the smallest defect over the blocks of any seed is $152°$. Figure 6 shows the token quaternions at successive training steps in one seed, one snapshot every $1{,}000$ steps.

B.3 Composing Continuous Rotations

A sequence of rotations enters the model one rotation per position. The target at position k is a fixed unit vector $\mathbf{v}_0$ rotated by the product of the rotations seen so far, $\mathbf{v}_k = \mathbf{R}_k \cdots \mathbf{R}_1 \mathbf{v}_0$. The model outputs a vector at every position, and the loss is the mean squared error over positions and



(a) Rotation axes of the sixty elements on a quaternion block. Shown at initialization, at a tenth, a quarter, a half and nine tenths of validation accuracy, and finally at the best checkpoint.



(b) The defect (Equation 17) in degrees (left axis) and validation accuracy (right axis) against training step. The vertical dotted lines are the training steps of the snapshots in (a).

Figure 6: **The Homomorphism of A_5 Forming Along Training.** Layout $(\frac{1}{2}, \frac{3}{32})$, seed 3, one snapshot per 1,000 steps.

coordinates. This is the rigid-body rotation task (Heo et al., 2026), whose released code we run unmodified with our layer in place of their Mamba layer. In their task every rotation is one of the sixty elements of A_5, embedded in $\mathrm{SO}(3)$ as the rotation group of the icosahedron. The model receives the element as one of sixty tokens through a learned embedding, and $\mathbf{v}_0 = (1,1,1)/\sqrt{3}$.

We keep the target, the loss, the sequence lengths and the training loop, and change where the rotations come from and how the model receives them. On the task $\mathrm{SO}(3)$, *uniform* the rotation at every position is drawn from the uniform distribution on $\mathrm{SO}(3)$. The model receives its rotation vector, the rotation angle in $[0,\pi]$ times the unit axis, as three real numbers through a linear map in place of the token embedding. On the task $\mathrm{SO}(4)$, *uniform* the rotation is drawn from the uniform distribution on $\mathrm{SO}(4)$ and acts on the fixed unit vector $\mathbf{v}_0 = (1,1,1,1)/2$ of $\mathbb{R}^4$. Every rotation of $\mathbb{R}^4$ is a two-sided product $z \mapsto q^L z \overline{q^R}$ of two unit quaternions (Section 3). The model receives the pair of half-angle vectors $(\mathbf{u}, \mathbf{v})$ with $q^L = \exp(\mathbf{u})$ and $q^R = \exp(\mathbf{v})$, six real numbers, where the sign of the pair is fixed by a nonnegative real part of q^L so that every rotation has one input. A fourth task, $\mathrm{SO}(3)$, *small angles*, draws the axis uniformly and the angle uniformly on $[0, 0.3]$ radians. It appears in Table 5 only, where every model has low error within the trained length. We call the error of the constant prediction $\mathbf{0}$ *chance*. Since the squared coordinates of a unit vector sum to one, it is $1/3$ on the tasks in $\mathbb{R}^3$ and $1/4$ on $\mathrm{SO}(4)$.

We train on 495,000 sequences of length 128 for 50 epochs at batch size 256, with AdamW at learning rate 10^{-3}, cosine decay and weight decay 10^{-6}, i.e. the settings as originally released by Heo et al. (2026), and evaluate on 5,000 held-out sequences of length 256. The statistic is the error of the last epoch by position, averaged over positions 1 to 128 (within the trained length) and over positions 129 to 256 (beyond it), and then over seeds 0 to 2. Every model has width 128. The depths are 1, 2 and 4 layers on the tasks in $\mathbb{R}^3$, and 2, 4, 8 and 12 layers on $\mathrm{SO}(4)$.

Mamba-3 and Icosian run at state size 128. GDN runs at eight heads. GDP runs with two Householder reflections per token at eight heads. With four reflections it runs at eight heads on the two uniform tasks and, on the A_5 task, in the configuration of the DeltaProduct paper (Siems et al., 2025a) with four heads. Flash PD-SSM runs in the gated form with four heads, state size 128 and a dictionary of 64 matrices initialized to the identity (Appendix B.1). Table 5 gives the errors within

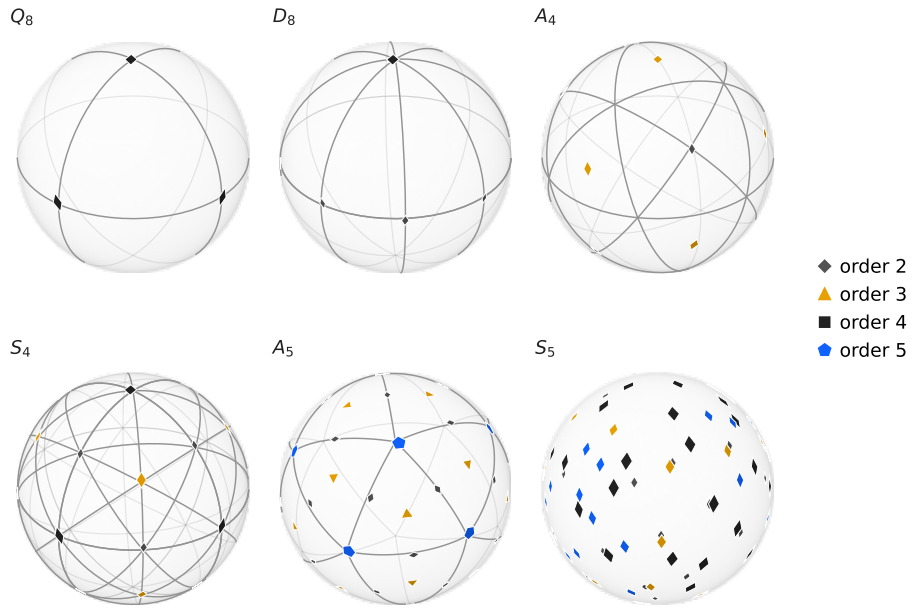


Figure 7: **Learned Homomorphisms.** The same measurement as in Figure 1c, repeated for all groups in Table 2, including also A_4. Marker shape gives the element order (square four, diamond two, pentagon five, triangle three), drawn on the axes of the group as a subgroup of $\mathrm{SO}(3)$. The defects are Q_8 10.1°, D_8 5.9°, A_4 10.9°, S_4 5.7°, A_5 12.2°, and S_5 157°. On S_5, no homomorphism forms.

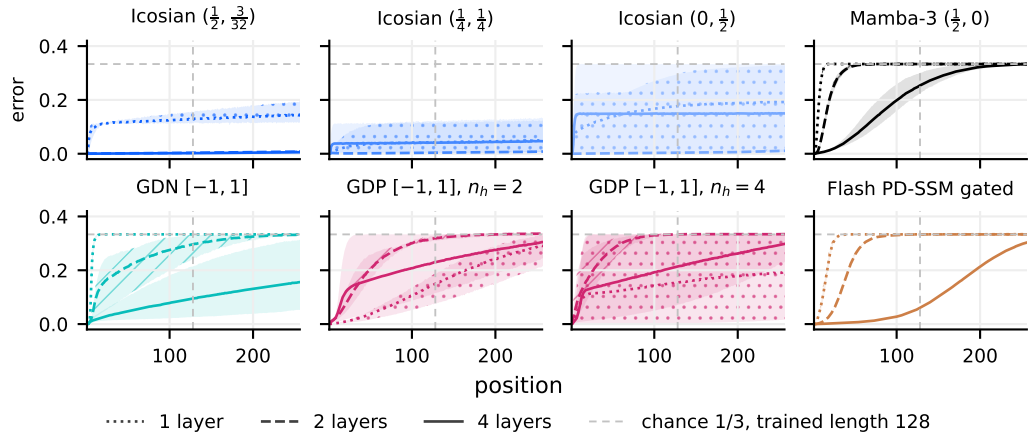
and beyond the trained length, and Figure 8 shows the error against position for every model and depth combination.

Table 5: **Continuous Rotation Tasks.** Error (↓) IID (before training maximum) and OOD (beyond training maximum). Mean over three seeds. Four layers in general, except twelve on $\mathrm{SO}(4)$. Chance is $1/3$ in general, excepting $1/4$ on $\mathrm{SO}(4)$. Bold marks the lowest mean per column.

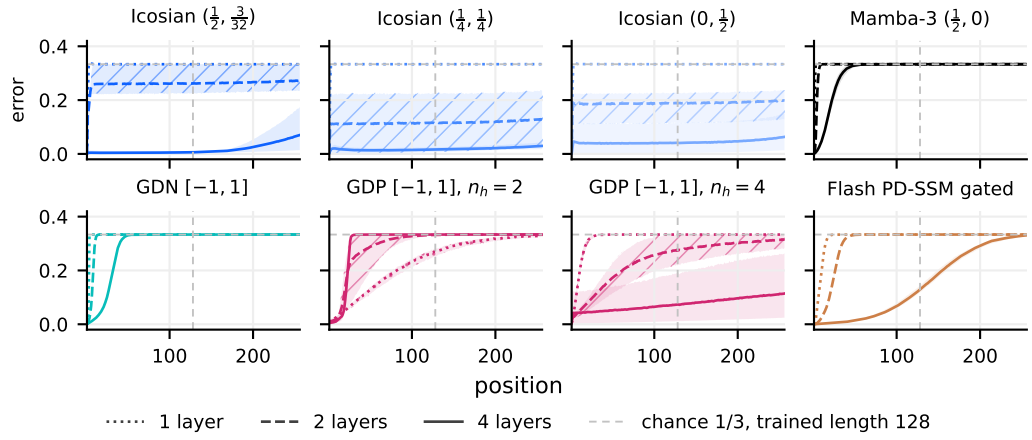
	alphabet of A_5		$\mathrm{SO}(3)$, uniform		$\mathrm{SO}(3)$, small angles		$\mathrm{SO}(4)$, uniform	
model	IID	OOD	IID	OOD	IID	OOD	IID	OOD
Discrete SSMs								
Flash PD-SSM	0.0162	0.1925	0.0374	0.2625	**0.0000**	**0.0036**	**0.0016**	0.1914
Continuous SSMs								
GDN $[-1, 1]$	0.0552	0.1269	0.2601	0.3333	0.0010	0.0261	0.0565	0.2390
GDP $[-1, 1]$, 2 refl.	0.1636	0.2702	0.2790	0.3334	0.0004	0.0170	0.0219	0.2511
GDP $[-1, 1]$, 4 refl.	0.1554	0.2577	0.0555	0.0937	–	–	0.0377	0.1147
Mamba-3	0.1131	0.3107	0.2798	0.3334	0.0020	0.0610	0.1240	0.2489
Icosian	**0.0013**	**0.0037**	**0.0046**	**0.0272**	0.0012	0.0465	0.0170	**0.0443**

B.4 State Tracking in Natural Language

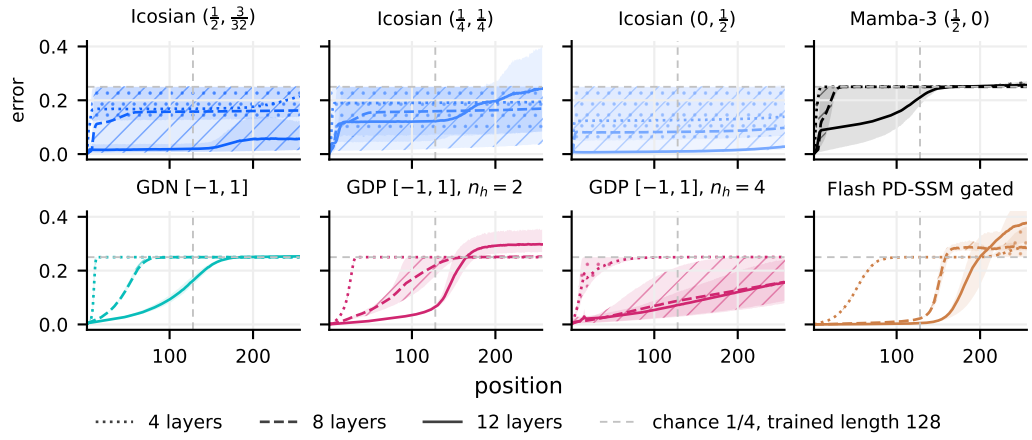
Defined in (Terzic et al., 2025, Section 4.5). One trainable layer of width 128 and state 128 reads the hidden states of a frozen Qwen 2.5-1.5B (Qwen et al., 2025) through a linear map and a normalization. A readout at the last position predicts the state. We train with Adam at 10^{-3} for 100,000 steps at batch 128, with a length curriculum to 25 sentences, on seeds 0 to 4. We report the best-validation checkpoint evaluated at lengths up to 80 and the maximum over seeds per length. The mean $\pm$ sd of the best validation accuracy is 58.4 ± 2.1 for Mamba-3 and 85.6 ± 16.8 for Icosian.



(a) The A_5 task (Heo et al., 2026), sixty tokens. Depths 1, 2 and 4.



(b) Rotations drawn uniformly from $\mathrm{SO}(3)$. Depths 1, 2 and 4.



(c) Rotations drawn uniformly from $\mathrm{SO}(4)$. Depths 4, 8 and 12.

Figure 8: **Full overview of results on continuous rotation tasks.** Figure 4 showed every model at its best depth.

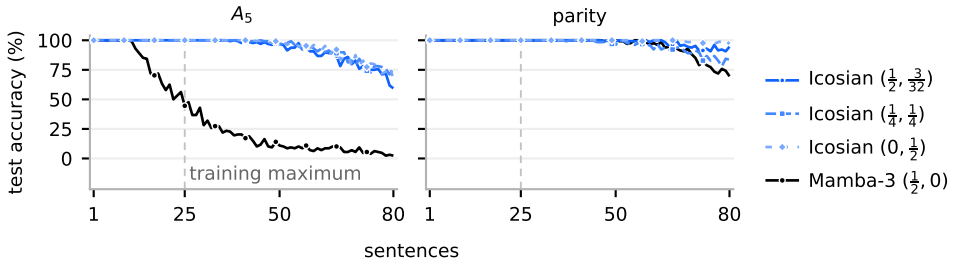


Figure 9: **Every Icosian Layout on State-Tracking in Natural Language.** An extension of Figure 3 for all three considered Icosian state layouts, reporting maximum over five seeds at each length.

Table 6: **Mechanistic Architecture Design.** Accuracy (%) on the MAD benchmark Poli et al. (2024). Results marked with † are as originally reported by their authors. Best results are in bold.

Model	Compress ↑	Fuzzy ↑	IC Rec. ↑	Memorize ↑	Noisy ↑	Sel. Copy ↑	Avg. ↑
Transformer†	51.6	29.8	94.1	85.2	86.8	99.6	74.5
Flash PD-SSM	47.8	16.7	97.4	71.9	96.5	92.2	70.4
GDN $[-1, 1]$	49.5	**26.6**	99.9	39.2	99.9	90.8	67.7
GDP $[-1, 1]$	51.7	23.6	99.8	31.1	99.9	93.6	66.6
Mamba†	52.7	6.7	90.4	89.5	90.1	86.3	69.3
Mamba-2	54.1	10.7	98.8	84.5	98.4	78.4	70.8
Mamba-3	52.9	19.5	**100.0**	**88.5**	**100.0**	96.6	76.2
Icosian ($\frac{1}{2}, \frac{1}{4}$)	**56.0**	22.0	**100.0**	87.5	**100.0**	**96.9**	**77.0**

B.5 Mechanistic Architecture Design

Every model has two layers of width 128 and a 4096-dimensional recurrent state, trained with AdamW ($\beta_2 = 0.98$) over learning rates $\{1, 5, 10\}\times10^{-4}$ and weight decays $\{0, 0.1\}$ for 200 epochs. GDN and GDP run with and without the suite's SwiGLU layer and at three head configurations, and we report the best mean. Flash PD-SSM is the real-valued gated form (Terzić et al., 2026).

B.6 Chomsky-Hierarchy Suite

Based on (Walker et al., 2025). Training runs for 100,001 steps with batch size 256 and AdamW at 2×10^{-3}. The LR schedule consists of warm-up followed by cosine decay, and a weight decay of 10^{-2} is applied. The training covers lengths 3 to 40. Validation covers lengths 40 to 256. We report the best validation accuracy as mean ± sd over ten seeds, with the best width (128 or 512). Flash PD-SSM is the simplified architecture with 8 heads of state 128 and dictionary size 8.

Table 7: **Chomsky-Hierarchy Tasks.** Validation accuracy (%) on the regular language tasks from (Delétang et al., 2023; Walker et al., 2025). Reporting mean ± sd over ten seeds. Results marked with † are as originally reported by their authors. Best results in bold.

$(f_{\mathbb{C}}, f_{\mathbb{H}})$	Model	Parity ↑	Even Pairs ↑	Cycle Nav. ↑	Mod. Arith. ↑	Avg. ↑
	LSTM†	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	99.9 ± 0.1	100.0
	BD-SLiCE† (128 wide, 4 blocks)	95.3 ± 3.9	85.9 ± 11.3	99.8 ± 0.2	54.0 ± 12.5	83.8
	Flash PD-SSM	97.5 ± 5.7	98.1 ± 3.2	$\mathbf{99.4 \pm 1.3}$	93.4 ± 7.8	**97.1**
	GDN $[-1, 1]$	$\mathbf{99.8 \pm 0.3}$	$\mathbf{100.0 \pm 0.0}$	49.5 ± 3.3	88.8 ± 15.3	84.5
	GDP $[-1, 1]$, 2 refl.	$\mathbf{99.8 \pm 0.2}$	$\mathbf{100.0 \pm 0.0}$	49.7 ± 18.2	$\mathbf{99.2 \pm 0.8}$	87.2
$(0, 0)$	Mamba-2	55.0 ± 6.9	$\mathbf{100.0 \pm 0.0}$	46.9 ± 7.0	25.6 ± 3.7	56.9
$(\frac{1}{2}, 0)$	Mamba-3	73.2 ± 14.2	$\mathbf{100.0 \pm 0.0}$	54.2 ± 12.2	55.8 ± 15.6	70.8
$(\frac{1}{2}, \frac{3}{32})$	Icosian	90.4 ± 14.7	$\mathbf{100.0 \pm 0.0}$	53.4 ± 8.6	72.0 ± 17.1	78.9
$(\frac{1}{4}, \frac{1}{4})$	Icosian	92.0 ± 17.3	$\mathbf{100.0 \pm 0.0}$	61.4 ± 22.5	78.0 ± 21.6	82.8
$(0, \frac{1}{2})$	Icosian	86.3 ± 21.0	$\mathbf{100.0 \pm 0.0}$	84.1 ± 15.4	71.0 ± 17.4	85.3

Table 8: **Every Icosian Layout on Language Modeling.** An extended version of Table 1 for every considered Icosian state layout, with the average recall accuracy over the six tasks of Arora et al. (2024).

$(f_{\mathbb{C}}, f_{\mathbb{H}})$	Model	PPL ↓	LAMBADA PPL ↓	LAMBADA ↑	PIQA ↑	HellaSwag ↑	Winogrande ↑	ARC-e ↑	ARC-c ↑	Avg. ↑	Recall ↑
0.5*B*											
	GDN $[-1, 1]$	21.77(0.05)	30.70(0.87)	32.6(0.7)	**68.2(0.8)**	37.4(0.2)	51.1(0.6)	46.7(1.0)	24.2(0.7)	43.4(0.4)	14.9(0.4)
	GDP $[-1, 1]$, 2 refl.	21.83(0.01)	30.56(1.55)	33.4(0.8)	67.6(0.5)	37.3(0.2)	51.6(1.1)	46.1(0.7)	24.1(0.8)	43.4(0.5)	14.7(0.5)
$(\frac{1}{2}, 0)$	Mamba-3	21.76(0.05)	26.92(1.70)	35.5(1.2)	67.5(0.3)	37.2(0.1)	50.5(0.7)	47.0(0.3)	23.9(0.5)	43.6(0.2)	16.5(0.5)
$(\frac{1}{2}, \frac{3}{32})$	**Icosian**	21.73(0.02)	26.75(0.38)	35.6(0.4)	67.7(0.3)	**37.5(0.2)**	51.1(0.8)	47.0(0.5)	24.2(0.2)	43.9(0.2)	**17.0(0.4)**
$(\frac{1}{2}, \frac{1}{4})$	**Icosian**	**21.71(0.02)**	25.98(0.46)	36.2(0.4)	67.4(0.5)	37.3(0.3)	51.1(0.7)	**47.1(0.7)**	**24.4(0.8)**	43.9(0.3)	16.4(0.5)
$(\frac{1}{4}, \frac{1}{4})$	**Icosian**	21.72(0.06)	**25.65(0.97)**	**36.3(0.6)**	67.7(0.4)	37.2(0.3)	**52.8(1.6)**	46.8(0.6)	24.3(1.0)	**44.2(0.3)**	16.7(0.2)
0.5*B MIMO*											
$(\frac{1}{2}, 0)$	Mamba-3, MIMO	21.36(0.03)	24.40(0.98)	37.0(0.5)	67.7(0.9)	38.0(0.3)	**51.8(1.3)**	47.1(1.0)	**24.3(0.3)**	**44.3(0.2)**	17.3(0.2)
$(\frac{1}{2}, \frac{3}{32})$	**Icosian**, MIMO	21.31(0.04)	24.89(1.22)	36.7(0.7)	**68.4(0.5)**	37.8(0.2)	51.4(0.7)	47.2(0.8)	24.1(0.5)	**44.3(0.2)**	**17.5(0.3)**
$(\frac{1}{2}, \frac{1}{4})$	**Icosian**, MIMO	21.31(0.04)	24.59(0.49)	36.9(0.7)	67.9(0.3)	37.8(0.4)	51.2(2.0)	47.1(0.7)	24.2(0.8)	44.2(0.3)	17.4(0.5)
$(\frac{1}{4}, \frac{1}{4})$	**Icosian**, MIMO	**21.27(0.03)**	**23.86(0.24)**	**37.2(0.4)**	67.8(0.3)	**38.1(0.2)**	50.3(1.0)	**47.4(0.6)**	**24.3(0.8)**	44.2(0.3)	17.3(0.5)
1.1*B*											
	GDN $[-1, 1]$	16.10(0.05)	13.24(0.66)	45.2(0.8)	71.9(0.3)	49.4(0.2)	52.8(1.1)	54.5(1.0)	26.2(0.6)	50.0(0.4)	21.9(0.4)
	GDP $[-1, 1]$, 2 refl.	16.18(0.05)	13.37(0.42)	44.6(0.9)	71.5(0.9)	48.9(0.3)	52.6(0.8)	53.9(0.7)	25.8(0.8)	49.5(0.3)	22.0(0.9)
$(\frac{1}{2}, 0)$	Mamba-3	16.07(0.03)	12.37(0.38)	46.2(0.8)	72.0(0.4)	**49.7(0.2)**	52.9(1.3)	54.9(1.2)	26.0(0.4)	50.3(0.3)	24.3(0.5)
$(\frac{1}{2}, \frac{3}{32})$	**Icosian**	16.03(0.03)	**12.10(0.25)**	**46.9(0.8)**	72.0(0.3)	49.6(0.2)	52.8(0.6)	54.8(0.6)	**27.1(0.8)**	50.5(0.1)	**24.6(0.4)**
$(\frac{1}{2}, \frac{1}{4})$	**Icosian**	**16.02(0.03)**	12.23(0.45)	46.8(0.6)	**72.2(0.6)**	**49.7(0.1)**	**53.4(0.5)**	54.7(0.7)	26.2(0.3)	50.5(0.1)	24.1(0.8)
$(\frac{1}{4}, \frac{1}{4})$	**Icosian**	**16.02(0.02)**	12.25(0.14)	46.7(0.4)	72.1(0.4)	**49.7(0.2)**	53.1(1.1)	**55.7(0.6)**	26.5(1.3)	**50.6(0.2)**	23.8(0.6)
3*B*											
$(\frac{1}{2}, 0)$	Mamba-3	12.27(0.02)	6.95(0.20)	56.6(0.6)	75.6(0.4)	63.8(0.1)	58.6(1.2)	**65.3(0.1)**	**32.6(0.3)**	58.7(0.2)	**31.5(1.3)**
$(\frac{1}{2}, \frac{3}{32})$	**Icosian**	**12.25(0.04)**	**6.80(0.07)**	**57.0(0.5)**	**75.9(0.2)**	**64.2(0.4)**	**59.0(0.4)**	64.9(0.6)	31.7(0.6)	**58.8(0.1)**	30.8(0.4)

B.7 Language Modeling

We pretrain under the training setup of the FineWeb ablations (Penedo et al., 2024): the GPT-2 tokenizer on the 100-billion-token sample of FineWeb, sequences of $2{,}048$ tokens, 2^{21} tokens per step and 24 layers. The optimizer is AdamW $(0.9, 0.95)$ with weight decay 0.1, a learning rate of $3{\times}10^{-4}$, 500 warm-up steps and cosine decay to $3{\times}10^{-5}$. Each model sees about twenty tokens per parameter, namely 10.1B tokens at 0.5B, 22.2B at 1.1B and 60.0B at 3B.

GDN and GDP (2 reflections, head width 128) are sized to 505M and 527M parameters against Mamba-3's 514M. Both allow eigenvalues in $[-1, 1]$ (Grazzi et al., 2025; Siems et al., 2025a).

Mamba-3 and every layout of Icosian are also further trained in Mamba-3's MIMO form of rank 4 at 0.5B, with the MIMO kernel of Mamba-3 correspondingly adapted for Icosian.

Zero-shot evaluation uses lm-eval-harness $0.4.9$ on LAMBADA, HellaSwag, PIQA, ARC-e, ARC-c and Winogrande.

Recall covers the six retrieval tasks of the recall benchmark (Arora et al., 2024): SWDE, SQuAD, FDA, TriviaQA, Natural Questions and DROP, with inputs cut to $2{,}000$ tokens. The recall average is in Table 8 and the six tasks are in Table 9. Table 10 gives the loss by position, inside and beyond the training context. Icosian's per-task means lie within the range of Mamba-3's per-seed values at every scale. The gap of GDN and GDP to Mamba-3 lies on SQuAD, TriviaQA and DROP.

Seed-Level Tests. We apply the test from Appendix B.1 to the perplexity and to the zero-shot and recall averages, with five seeds a side unless stated otherwise. Every p is in Table 3. The test rejects in Icosian's favour in four cases: the perplexity of the default layout at 1.1B ($p = 0.032$), the perplexity of $(\frac{1}{4}, \frac{1}{4})$ at 1.1B ($p = 0.016$), the perplexity of $(\frac{1}{4}, \frac{1}{4})$ in the MIMO form at 0.5B ($p = 0.0079$), and the zero-shot average of $(\frac{1}{4}, \frac{1}{4})$ at 0.5B ($p = 0.0079$). It rejects for no recall average. It rejects in Mamba-3's favour for the perplexity of GDP at 0.5B and at 1.1B ($p = 0.0079$ both), for the zero-shot average of GDP at 1.1B ($p = 0.0079$), and for the recall averages of GDN and GDP at both scales ($p = 0.0079$ and 0.0079 at 0.5B, 0.0079 and 0.016 at 1.1B). GDN's perplexity and zero-shot average are not rejected at either scale.

B.8 Training Time, Inference Latency and Peak Memory

A training step consists of the forward pass, the backward pass and the optimizer step of a 24-layer model on one H100 in bf16. It is measured at sequence length 2048 and the per-GPU batch used in the language modeling experiments, averaged over 5 steps. The scan and its adjoint take 6.4 to 2.4 percent of a step from 510M to 3.0B.

Table 9: **Retrieval tasks.** Accuracy (%) on the retrieval tasks Arora et al. (2024) at 2,000 tokens, mean over seeds with the standard deviation over seeds in parentheses. Bold marks the best mean per column within a size.

$(f_{\mathbb{C}}, f_{\mathbb{H}})$	Model	SWDE ↑	SQuAD ↑	FDA ↑	TriviaQA ↑	NQ ↑	DROP ↑	Avg. ↑
0.5*B*								
	GDN $[-1, 1]$	**11.3(0.8)**	16.4(0.6)	**3.9(0.6)**	34.4(1.3)	**9.8(0.9)**	13.4(0.5)	14.9(0.4)
	GDP $[-1, 1]$, 2 refl.	10.5(0.8)	16.0(0.6)	3.5(1.4)	35.5(0.5)	9.4(0.5)	13.6(0.9)	14.7(0.5)
$(\frac{1}{2}, 0)$	Mamba-3	10.6(0.7)	20.8(0.8)	2.9(1.0)	39.1(0.7)	9.1(0.3)	16.5(1.0)	16.5(0.5)
$(\frac{1}{2}, \frac{3}{32})$	**Icosian**	11.2(1.0)	**21.7(0.8)**	3.5(0.6)	**39.3(1.1)**	9.7(0.9)	16.5(0.3)	**17.0(0.4)**
$(\frac{1}{2}, \frac{1}{4})$	**Icosian**	10.7(0.8)	21.3(1.2)	2.5(0.5)	38.7(0.4)	9.0(0.8)	16.4(1.0)	16.4(0.5)
$(\frac{1}{4}, \frac{1}{4})$	**Icosian**	11.0(0.6)	21.4(0.8)	2.8(0.7)	38.9(0.7)	9.2(0.6)	**16.9(0.4)**	16.7(0.2)
0.5*B MIMO*								
$(\frac{1}{2}, 0)$	Mamba-3, MIMO	**12.4(1.4)**	22.2(0.7)	3.2(0.2)	39.3(0.8)	9.3(1.0)	**17.3(0.8)**	17.3(0.2)
$(\frac{1}{2}, \frac{3}{32})$	**Icosian**, MIMO	11.9(1.4)	**22.8(0.5)**	3.5(0.7)	**40.2(1.1)**	9.5(0.3)	17.1(0.7)	**17.5(0.3)**
$(\frac{1}{2}, \frac{1}{4})$	**Icosian**, MIMO	**12.4(0.7)**	21.6(0.3)	**3.8(1.5)**	39.7(1.0)	**10.1(1.4)**	16.8(0.7)	17.4(0.5)
$(\frac{1}{4}, \frac{1}{4})$	**Icosian**, MIMO	11.4(0.8)	22.2(0.7)	3.7(1.2)	39.8(1.1)	9.5(0.5)	17.1(1.4)	17.3(0.5)
1.1*B*								
	GDN $[-1, 1]$	18.2(2.9)	23.3(1.1)	9.8(1.6)	46.8(1.1)	15.1(1.0)	18.2(0.7)	21.9(0.4)
	GDP $[-1, 1]$, 2 refl.	17.1(2.4)	23.7(1.0)	10.2(2.5)	46.4(1.3)	16.1(1.3)	18.4(2.0)	22.0(0.9)
$(\frac{1}{2}, 0)$	Mamba-3	**19.6(1.6)**	29.1(0.7)	9.2(2.4)	**51.1(0.5)**	**16.5(1.1)**	20.4(0.8)	24.3(0.5)
$(\frac{1}{2}, \frac{3}{32})$	**Icosian**	19.4(1.7)	**29.5(0.8)**	**11.0(2.1)**	50.9(0.8)	16.1(0.7)	20.4(0.2)	**24.6(0.4)**
$(\frac{1}{2}, \frac{1}{4})$	**Icosian**	18.2(2.0)	29.4(0.6)	9.8(2.7)	50.9(0.8)	**16.5(1.4)**	19.6(1.0)	24.1(0.8)
$(\frac{1}{4}, \frac{1}{4})$	**Icosian**	17.7(0.2)	28.9(0.8)	9.0(2.1)	50.3(1.0)	16.4(0.8)	**20.7(1.1)**	23.8(0.6)
3*B*								
$(\frac{1}{2}, 0)$	Mamba-3	26.0(0.8)	**34.8(0.7)**	**22.5(6.9)**	**60.6(0.7)**	23.5(1.2)	**21.4(0.7)**	**31.5(1.3)**
$(\frac{1}{2}, \frac{3}{32})$	**Icosian**	**26.3(3.8)**	**34.8(1.0)**	18.7(2.1)	60.3(0.4)	**24.0(0.7)**	20.8(0.6)	30.8(0.4)

Table 10: **FineWeb Loss by Position.** Token loss (↓) inside and beyond the 2,048-token training context, mean ± sd over seeds. Bold marks the lower mean; tests in Table 3.

Model	0–2048	2048–4096	4096–8192	8192–16384	16384–32768
0.5*B*, 5 *seeds*					
Mamba-3	$\mathbf{3.015 \pm 0.003}$	$\mathbf{3.088 \pm 0.004}$	3.162 ± 0.004	$\mathbf{3.042 \pm 0.002}$	3.042 ± 0.003
Icosian	$\mathbf{3.015 \pm 0.003}$	$\mathbf{3.088 \pm 0.002}$	$\mathbf{3.160 \pm 0.002}$	$\mathbf{3.042 \pm 0.002}$	$\mathbf{3.041 \pm 0.002}$
1.1*B*, 5 *seeds*					
Mamba-3	2.720 ± 0.002	2.804 ± 0.004	2.861 ± 0.003	2.748 ± 0.002	2.743 ± 0.002
Icosian	$\mathbf{2.715 \pm 0.002}$	$\mathbf{2.800 \pm 0.003}$	$\mathbf{2.857 \pm 0.003}$	$\mathbf{2.744 \pm 0.001}$	$\mathbf{2.740 \pm 0.001}$
3*B*, 3 *and* 3 *seeds*					
Mamba-3	2.453 ± 0.003	2.507 ± 0.002	2.577 ± 0.002	2.482 ± 0.003	2.475 ± 0.002
Icosian	$\mathbf{2.451 \pm 0.002}$	$\mathbf{2.504 \pm 0.004}$	$\mathbf{2.575 \pm 0.004}$	$\mathbf{2.481 \pm 0.003}$	$\mathbf{2.473 \pm 0.003}$

Inference is measured as per Mamba-3 (Lahoti et al., 2026): namely, we measure the prefill plus decode time of as many tokens as were prefilled. Batch size is 128. The results are shown in Figure 10.

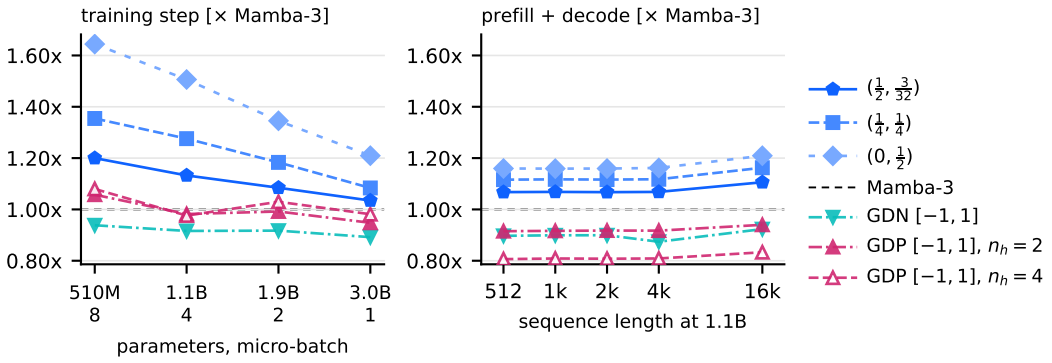


Figure 10: **Training and Inference Cost.** Cost relative to the parameter-matched Mamba-3 (the line at 1). Left: training step time of 24-layer models over batch size; Right: prefill plus decode over sequence length at 1.1B and batch 128.

C Clifford Algebras, Division Algebras, and Dimension 4

In addition to the usual representation in form of $d \times d$ orthogonal matrices, all $\mathbb{R}^d$ rotations admit a *rotor* representation: unit elements R of the even *Clifford subalgebra* $\mathrm{Cl}^+(d)$ act on $\mathbb{R}^d$ as $x \mapsto RxR^{-1}$. The even Clifford subalgebra $\mathrm{Cl}^+(d)$ forms the double cover $\mathrm{Spin}(d)$ of $\mathrm{SO}(d)$ (Lounesto, 2001). $\mathrm{Cl}^+(d)$ has dimension 2^{d-1}, so whether rotors are cheaper than orthogonal matrices depends on the structure of $\mathrm{Cl}^+(d)$.

For $d \leq 4$, $\mathrm{Cl}^+(d)$ is built from associative division algebras: $\mathrm{Cl}^+(2) \cong \mathbb{C}$, $\mathrm{Cl}^+(3) \cong \mathbb{H}$ and, exceptionally, $\mathrm{Cl}^+(4) \cong \mathbb{H} \oplus \mathbb{H}$, the pair of quaternions from Proposition 1. By Hurwitz's and Frobenius's theorems, $\mathbb{R}$, $\mathbb{C}$ and $\mathbb{H}$ are the only associative normed division algebras over $\mathbb{R}$, hence this construction cannot continue for arbitrarily large d. Indeed, for $d \geq 5$, $\mathrm{Cl}^+(d)$ is a full matrix algebra over $\mathbb{R}$, $\mathbb{C}$ or $\mathbb{H}$, or a direct sum of two such algebras when $d \equiv 0 \pmod 4$ (Lounesto, 2001), and composing rotors becomes more expensive than composing orthogonal matrices.

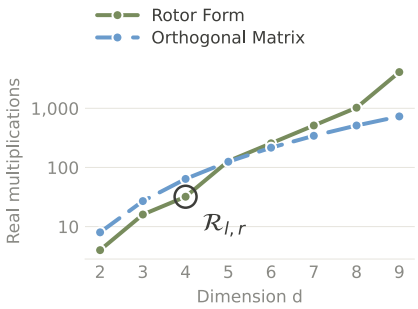


Figure 11: Cost of accumulating two rotations with rotors and with orthogonal matrices. The quaternionic operator $\mathcal{R}_{l,r}$ (circled) requires 32 multiplications at $d = 4$, compared to 64 for the matrix form. Past $d = 4$, rotors are no longer cheaper.

Figure 11 shows the cost of accumulating two rotations with both representations. The quaternionic operator $\mathcal{R}_{l,r}$ is the efficient rotor at $d = 4$. It requires 32 multiplications, compared to the 64 of the matrix form. The efficient rotors at $d = 2$ and $d = 3$ accumulate rotations as $z_1 z_2$ with $z_1, z_2 \in S^1$ and $q_1 q_2$ with $q_1, q_2 \in S^3$, respectively. Dimension 4 is thus the largest in which rotations can be composed more cheaply than with orthogonal matrices.

D PROOFS

D.1 PROOF OF PROPOSITION 2 (ROTATIONAL FLOW OF THE QUATERNIONIC ODE)

As a reminder, the proposition we are proving in this section is:

Proposition 2 (Rotational Flow of the Quaternionic ODE)**.** *Let $l, r\colon [0,T] \to \operatorname{Im}\mathbb{H}$ be continuous, $\mathcal{T}\exp(\cdot)$ the time-ordered exponential, and consider Equation 9 with $a \equiv b \equiv 0$. Its solution is*

$$h(t) = U_L(t)\, h(0)\, \overline{U_R(t)},$$

where $U_L, U_R\colon [0,T] \to S^3$ solve $\dot{U}_L = l\, U_L$ and $\dot{U}_R = r\, U_R$ with $U_L(0) = U_R(0) = 1$; equivalently, $U_L = \mathcal{T}\exp\int_0^t l(s)\, ds$ and $U_R = \mathcal{T}\exp\int_0^t r(s)\, ds$. Conversely, for every continuously differentiable curve $R\colon [0,T] \to \mathrm{SO}(4)$ with $R(0) = \mathrm{id}$ there exist l, r for which the solution satisfies $h(t) = R(t)\, h(0)$.

Proof. We first solve the isolated system $\dot{h}(t) = l(t)h(t) - h(t)r(t)$ with $l(\cdot), r(\cdot)\colon [0,T] \to \operatorname{Im}\mathbb{H}$ continuous, and then establish the converse direction stated in the proposition.

Existence and uniqueness of U_L, U_R.

$U_L(\cdot), U_R(\cdot)$ solving $\dot{U}_L = l\, U_L$ and $\dot{U}_R = r\, U_R$ with $U_L(0) = U_R(0) = 1$ exist and are unique by Picard–Lindelöf on continuous, linear-coefficient ODEs on compact domains. By definition, $U_L = \mathcal{T}\exp\int_0^t l(s)\, ds.$, with $\mathcal{T}\exp$ the time-ordered exponential (Agrachev & Sachkov, 2013).

Norm preservation.

The functions U_L, U_R are shown to be unit quaternions as:

$$\frac{d}{dt}|U_L|^2 = \frac{d}{dt}\big(U_L\overline{U_L}\big) = \dot{U}_L\overline{U_L} + U_L\overline{\dot{U}_L} = l\, U_L\overline{U_L} + U_L\overline{U_L}\,\bar{l} = |U_L|^2(l + \bar{l}) = 0,$$

so $|U_L|^2 = 1$ by the initial condition. The same argument gives $|U_R|^2 = 1$.

The forward direction.

The solution to $\dot{h} = lh - hr$ exists and is unique by Picard–Lindelöf, write $\tilde{h}(t)$ for it, with $\tilde{h}(0) = h_0$. We claim that $\tilde{h}(t) = U_L(t)\, h_0\, \overline{U_R(t)}$ with $U_L(\cdot), U_R(\cdot)$ as above. The right-hand side has the initial value h_0 and satisfies the equation,

$$\dot{\tilde{h}}(t) = \dot{U}_L h_0 \overline{U_R} + U_L h_0 \overline{\dot{U}_R} = l\, U_L h_0 \overline{U_R} - U_L h_0 \overline{U_R}\, r = l\,\tilde{h} - \tilde{h}\, r.$$

so it is the solution by uniqueness.

The converse direction.

By the path-lifting property of the covering map $S^3 \times S^3 \to \mathrm{SO}(4)$ from Proposition 1 (see e.g. (Hatcher, 2002)), a continuously differentiable $R\colon [0,T] \to \mathrm{SO}(4)$ with $R(0) = \mathrm{id}$ lifts to continuously differentiable $U_L, U_R\colon [0,T] \to S^3$ with $U_L(0) = U_R(0) = 1$ and $R(t)\, x = U_L(t)\, x\, \overline{U_R(t)}$.

Therefore, let $R\colon [0,T] \to \mathrm{SO}(4)$ act on $x \in \mathbb{H} \cong \mathbb{R}^4$ as follows:

$$\boldsymbol{R}(t)\boldsymbol{x} = U_L(t)\, x\, \overline{U_R(t)}$$

with $\boldsymbol{R}(0)\boldsymbol{x} = U_L(0)\, x\, \overline{U_R(0)} = x$.

We show that there exist continuous $l(\cdot), r(\cdot)\colon [0,T] \to \operatorname{Im}\mathbb{H}$ such that the $h(t)$ given by $\dot{h}(t) = l(t)h(t) - h(t)r(t)$ with $h(0) = x$ is solved by $U_L(t)\, x\, \overline{U_R(t)}$.

Define $l(t) := \dot{U}_L(t)\, \overline{U_L(t)}$, $r(t) := \dot{U}_R(t)\, \overline{U_R(t)}$; both continuous. Differentiating $U_L\overline{U_L} \equiv 1$ gives $l + \bar{l} = 0$, so $l \in \operatorname{Im}\mathbb{H}$; likewise $r \in \operatorname{Im}\mathbb{H}$. By construction, $\dot{U}_L = l\, U_L$ and $\dot{U}_R = r\, U_R$. Set $h(t) := U_L(t)\, x\, \overline{U_R(t)}$. Since $\overline{\dot{U}_R} = \overline{\dot{U}_R} = \overline{rU_R} = \overline{U_R}\,\bar{r} = -\overline{U_R}\, r$,

$$\dot{h} = \dot{U}_L\, x\, \overline{U_R} + U_L\, x\, \overline{\dot{U}_R} = l\, U_L x \overline{U_R} - U_L x \overline{U_R}\, r = l\, h - h\, r.$$

Finally, $h(0) = U_L(0)\, x\, \overline{U_R(0)} = x$. □

D.2 Proof of Proposition 3 (Closed-Form Expression for the Quaternionic System)

In this proposition, we utilize the variation of constants method to derive a general closed-form solution of the quaternionic ODE as:

Proposition 3 (Closed-Form Expression for the Quaternionic System)**.** *Let $a(\cdot), x(\cdot) \in \mathbb{R}$, $b(\cdot) \in \mathbb{H}$ and $l, r \in \operatorname{Im}\mathbb{H}$ be continuous on $[t_{k-1}, t_k]$. Given h_{k-1}, the analytical expression for h_k in Equation 9 is*

$$h_k = \exp\left(\int_{t_{k-1}}^{t_k} a(s)\,ds\right) \mathcal{T}\exp\Big(\int_{t_{k-1}}^{t_k} l(s)\,ds\Big)\, h_{k-1}\, \overline{\mathcal{T}}\exp\Big(\int_{t_{k-1}}^{t_k} \bar{r}(s)\,ds\Big) + \int_{t_{k-1}}^{t_k} \exp\left(\int_{\tau}^{t_k} a(s)\,ds\right) \mathcal{T}\exp\Big(\int_{\tau}^{t_k} l(s)\,ds\Big)\, b(\tau)\, x(\tau)\, \overline{\mathcal{T}}\exp\Big(\int_{\tau}^{t_k} \bar{r}(s)\,ds\Big)\, d\tau, \tag{11}$$

where $\mathcal{T}\exp(\cdot)$, $\overline{\mathcal{T}}\exp(\cdot)$ denote the time-ordered exponential in forward and backward order.

Proof. Let us start by defining several objects which will be useful in the following. For $s \le \tau \le t$, write

$$\alpha(t,\tau) := \exp\big(\int_{\tau}^{t} a(u)\,du\big),$$
$$U_L(t,\tau) := \mathcal{T}\exp\int_{\tau}^{t} l(u)\,du,$$
$$U_R(t,\tau) := \mathcal{T}\exp\int_{\tau}^{t} r(u)\,du,$$

so that α, U_L, U_R solve

$$\partial_t\alpha(t,\tau) = a(t)\alpha(t,\tau), \qquad \partial_t U_L(t,\tau) = l(t)U_L(t,\tau), \qquad \partial_t U_R(t,\tau) = r(t)U_R(t,\tau),$$

each equal to 1 at $t = \tau$. The first equality for $\partial_t\alpha(t,\tau)$ can be trivially verified, whereas the other two follow directly from the definition of the time-ordered exponential, given in the initial step of the proof in Section D.1.

Composition identities.

By uniqueness of solutions to linear ODEs with continuous coefficients (Picard–Lindelöf), $\alpha(t,s) = \alpha(t,\tau)\alpha(\tau,s)$ for $s \le \tau \le t$, since both sides solve the same ODE in t with the same value at $t = \tau$; likewise $U_L(t,s) = U_L(t,\tau)U_L(\tau,s)$ and $U_R(t,s) = U_R(t,\tau)U_R(\tau,s)$. Since U_L, U_R take values in the unit quaternions, inverses are conjugates, so these rearrange to

$$U_L(t,\tau) = U_L(t,s)\,\overline{U_L(\tau,s)}, \qquad U_R(\tau,s)\,\overline{U_R(t,s)} = \overline{U_R(t,\tau)}. \tag{18}$$

Homogeneous solution.

Consider the homogeneous ansatz $h_{\mathrm{hom}}(t) = \alpha(t,s)\,U_L(t,s)\,h(s)\,\overline{U_R(t,s)}$. Using $\partial_t\overline{U_R(t,s)} = \overline{r(t)U_R(t,s)} = -\overline{U_R(t,s)}\,r(t)$ (as $r(t) \in \operatorname{Im}\mathbb{H}$), the product rule can be verified to give:

$$\dot{h}_{\mathrm{hom}}(t) = a(t)h_{\mathrm{hom}}(t) + l(t)h_{\mathrm{hom}}(t) - h_{\mathrm{hom}}(t)\,r(t),$$

Hence, h_{hom} solves the homogeneous equation ($b \equiv 0$) with $h_{\mathrm{hom}}(s) = h(s)$.

Variation of constants.

Replace the constant $h(s)$ by a function $c(\tau)$, $c(s) = h(s)$, and take $h(\tau) = \alpha(\tau,s)\,U_L(\tau,s)\,c(\tau)\,\overline{U_R(\tau,s)}$ as ansatz for the full inhomogeneous ODE Equation 9. Differentiating as above, but retaining the extra term from $\dot{c}(\tau)$,

$$\dot{h}(\tau) = a(\tau)h(\tau) + l(\tau)h(\tau) - h(\tau)r(\tau) + \alpha(\tau,s)\,U_L(\tau,s)\,\dot{c}(\tau)\,\overline{U_R(\tau,s)}.$$

Equating this to $\dot{h}(\tau) = a(\tau)h(\tau) + l(\tau)h(\tau) - h(\tau)r(\tau) + b(\tau)x(\tau)$ and solving for $\dot{c}(\tau)$ using $\alpha(\tau, s)^{-1} = \alpha(s, \tau)$ and the unit-quaternion inverses $U_L(\tau, s)^{-1} = \overline{U_L(\tau, s)}$, $\overline{U_R(\tau, s)}^{-1} = U_R(\tau, s)$,

$$\dot{c}(\tau) = \alpha(s, \tau)\, \overline{U_L(\tau, s)}\, b(\tau)x(\tau)\, U_R(\tau, s).$$

Integration and assembly.

Integrating from s to t,

$$c(t) = h(s) + \int_s^t \alpha(s, \sigma)\, \overline{U_L(\sigma, s)}\, b(\sigma)x(\sigma)\, U_R(\sigma, s)\, d\sigma.$$

Substituting into the inhomogeneous ansatz at $\tau = t$,

$$\begin{aligned} h(t) &= \alpha(t, s)U_L(t, s)h(s)\overline{U_R(t, s)} \\ &\quad + \int_s^t \alpha(t, s)\alpha(s, \sigma)\, U_L(t, s)\overline{U_L(\sigma, s)}\, b(\sigma)x(\sigma)\, U_R(\sigma, s)\overline{U_R(t, s)}\, d\sigma. \end{aligned}$$

By scalar additivity of α and the identities equation 18, $\alpha(t, s)\alpha(s, \sigma) = \alpha(t, \sigma)$, $U_L(t, s)\overline{U_L(\sigma, s)} = U_L(t, \sigma)$, and $U_R(\sigma, s)\overline{U_R(t, s)} = \overline{U_R(t, \sigma)}$, so

$$h(t) = \alpha(t, s)\, U_L(t, s)\, h(s)\, \overline{U_R(t, s)} + \int_s^t \alpha(t, \sigma)\, U_L(t, \sigma)\, b(\sigma)x(\sigma)\, \overline{U_R(t, \sigma)}\, d\sigma,$$

which holds for all $0 \le s \le t \le T$. Taking $s = t_{k-1}$, $t = t_k$ gives exactly Equation 11. □

D.3 Deriving the Discretized Recurrence in Equations 12 and 13

Recall that our goal is to derive the discrete recurrence and readout of the main text,

$$h_k = e^{a_k\Delta_k} e^{l_k\Delta_k}\, h_{k-1}\, e^{\bar{r}_k\Delta_k} + (1 - \lambda_k)\, \Delta_k\, e^{a_k\Delta_k} e^{l_k\Delta_k}\, b_{k-1}x_{k-1}\, e^{\bar{r}_k\Delta_k} + \lambda_k\, \Delta_k\, b_k x_k, \quad (12)$$

$$y_k = \mathrm{Re}(\bar{c}_k h_k). \quad (13)$$

We start from the closed-form solution of Proposition 3. With $\alpha(t, \tau)$, $U_L(t, \tau)$ and $U_R(t, \tau)$ as defined in Appendix D.2, it reads, for all $0 \le s \le t \le T$,

$$h(t) = \alpha(t, s)\, U_L(t, s)\, h(s)\, \overline{U_R(t, s)} + \int_s^t \alpha(t, \tau)\, U_L(t, \tau)\, b(\tau)x(\tau)\, \overline{U_R(t, \tau)}\, d\tau.$$

We discretize it by instantiating it on a single step, $s = t_{k-1}$, $t = t_k$:

$$h_k = \alpha(t_k, t_{k-1})\, U_L(t_k, t_{k-1})\, h_{k-1}\, \overline{U_R(t_k, t_{k-1})} + \int_{t_{k-1}}^{t_k} \alpha(t_k, \tau)\, U_L(t_k, \tau)\, b(\tau)x(\tau)\, \overline{U_R(t_k, \tau)}\, d\tau. \quad (19)$$

Right-hand rule on the homogeneous term.

Write $a_k := a(t_k)$, $l_k := l(t_k)$, $r_k := r(t_k)$, and freeze the integrands defining α, U_L, U_R at their right endpoint over $[t_{k-1}, t_k]$:

$$\alpha(t_k, t_{k-1}) \approx e^{a_k\Delta_k}, \qquad U_L(t_k, t_{k-1}) \approx e^{l_k\Delta_k}, \qquad U_R(t_k, t_{k-1}) \approx e^{r_k\Delta_k},$$

with $\Delta_k := t_k - t_{k-1}$, matching the approximation $e^{\int_{t_1}^{t_2} a(s)ds} \approx e^{(t_2 - t_1)a(t_2)}$ used for the scalar case in Equation 6. The homogeneous term of equation 19 becomes

$$e^{a_k\Delta_k}\, e^{l_k\Delta_k}\, h_{k-1}\, e^{\bar{r}_k\Delta_k}.$$

Trapezoidal quadrature on the integral term.

Let $f(\tau) := \alpha(t_k, \tau)\, U_L(t_k, \tau)\, b(\tau)x(\tau)\, \overline{U_R(t_k, \tau)}$, so the integral in equation 19 is $\int_{t_{k-1}}^{t_k} f(\tau)\, d\tau$. Approximate this by the generalized trapezoidal rule with input-dependent weight $\lambda_k \in [0, 1]$,

$$\int_{t_{k-1}}^{t_k} f(\tau)\, d\tau \;\approx\; (1 - \lambda_k)\Delta_k\, f(t_{k-1}) + \lambda_k\Delta_k\, f(t_k).$$

At $\tau = t_k$: $\alpha(t_k, t_k) = U_L(t_k, t_k) = U_R(t_k, t_k) = 1$, so

$$f(t_k) = b_k x_k.$$

At $\tau = t_{k-1}$: $f(t_{k-1}) = \alpha(t_k, t_{k-1})\, U_L(t_k, t_{k-1})\, b_{k-1} x_{k-1}\, \overline{U_R(t_k, t_{k-1})}$, and applying the same right-hand-rule approximation as in Step 1,

$$f(t_{k-1}) \approx e^{a_k \Delta_k}\, e^{l_k \Delta_k}\, b_{k-1} x_{k-1}\, e^{\bar{r}_k \Delta_k}.$$

Assembling.

Substituting both pieces into equation 19 gives

$$h_k = e^{a_k \Delta_k}\, e^{l_k \Delta_k}\, h_{k-1}\, e^{\bar{r}_k \Delta_k} + (1 - \lambda_k)\, \Delta_k\, e^{a_k \Delta_k}\, e^{l_k \Delta_k}\, b_{k-1} x_{k-1}\, e^{\bar{r}_k \Delta_k} + \lambda_k\, \Delta_k\, b_k x_k, \quad (20)$$

which is exactly Equation 12. The readout $y_k = \mathrm{Re}(\bar{c}_k h_k)$ of Equation 13 is unchanged, since Equation 10 is evaluated pointwise at t_k.

D.4 Proof of Proposition 4 (Reformulated Scalar Recurrence)

Recall the per-token quaternions $q_k^L := \exp(l_k \Delta_k)$, $q_k^R := \exp(r_k \Delta_k)$ and their accumulated prefixes $Q_k^L := q_k^L\, Q_{k-1}^L$, $Q_k^R := q_k^R\, Q_{k-1}^R$, with $Q_0^L = Q_0^R = 1$. In this section, we prove that the quaternionic recurrence can be reformulated as follows:

Proposition 4 (Reformulated Scalar Recurrence)**.** *Define* $\tilde{h}_k := \overline{Q_k^L}\, h_k\, Q_k^R$, $\tilde{b}_k = \overline{Q_k^L}\, b_k\, Q_k^R$ *and* $\tilde{c}_k = \overline{Q_k^L}\, c_k\, Q_k^R$. *Then Equations 12 and 13 are equivalent to*

$$\tilde{h}_k = e^{a_k \Delta_k}\, \tilde{h}_{k-1} + (1 - \lambda_k) \Delta_k\, e^{a_k \Delta_k}\, \tilde{b}_{k-1} x_{k-1} + \lambda_k \Delta_k\, \tilde{b}_k x_k, \quad (14)$$

$$y_k = \mathrm{Re}(\bar{\tilde{c}}_k\, \tilde{h}_k) = \tilde{\mathbf{c}}_k^\top \tilde{\mathbf{h}}_k. \quad (15)$$

Proof. **Writing the sum-over-products form of the recurrence.**

For non-commutative quaternionic factors, define the ordered products

$$\prod_k^{s=n} q_s = q_n \cdots q_k, \qquad \prod_{s=k}^{n} q_s = q_k \cdots q_n, \qquad \text{both} = 1 \text{ whenever } k > n.$$

Write the per-step forcing quaternion of Equation 12 as

$$w_i := (1 - \lambda_i) \Delta_i\, e^{a_i \Delta_i}\, q_i^L\, b_{i-1} x_{i-1}\, \overline{q_i^R} + \lambda_i \Delta_i\, b_i x_i,$$

so that Equation 12 reads $h_k = e^{a_k \Delta_k}\, q_k^L\, h_{k-1}\, \overline{q_k^R} + w_k$. Taking $h_0 = 0$, as in the model, and unrolling this recurrence forward from $i = 1$ to t gives

$$h_t = \sum_{i=1}^{t} \left(\prod_{i+1}^{s=t} e^{a_s \Delta_s} \right) \left(\prod_{i+1}^{s=t} q_s^L\, (\cdot)\, \overline{q_s^R} \right) [w_i], \quad (21)$$

where $\prod_{i+1}^{s=t} q_s^L (\cdot) \overline{q_s^R}$ denotes the composition, in increasing order of s, of the maps $v \mapsto q_s^L\, v\, \overline{q_s^R}$; the scalar factors $e^{a_s \Delta_s}$ commute with every quaternion and were pulled out of the composition.

Simplifying notation.

The readout of Equation 13 is $y_t = \mathrm{Re}(\bar{c}_t h_t)$. Substituting Equation 21,

$$y_t = \sum_{i=1}^{t} \mathrm{Re} \left(\bar{c}_t \left(\prod_{i+1}^{s=t} e^{a_s \Delta_s} \right) \left(\prod_{i+1}^{s=t} q_s^L\, (\cdot)\, \overline{q_s^R} \right) [w_i] \right). \quad (22)$$

Recall that we defined $Q_k^L := q_k^L\, Q_{k-1}^L$, $Q_k^R := q_k^R\, Q_{k-1}^R$, $Q_0^L = Q_0^R = 1$ in the main text.

Applying the decomposition.

We claim that composing $v \mapsto q_s^L \, v \, \overline{q_s^R}$ in increasing order for $s = i+1, \dots, t$ gives

$$\prod_{i+1}^{s=t} q_s^L \, (\cdot) \, \overline{q_s^R} \, [v] = Q_t^L \, \overline{Q_i^L} \; v \; Q_i^R \, \overline{Q_t^R}. \tag{23}$$

Indeed, composing the maps in order yields $v \mapsto (q_t^L \cdots q_{i+1}^L) \, v \, (\overline{q_{i+1}^R} \cdots \overline{q_t^R})$. By the defining recursion, $Q_t^L = (q_t^L \cdots q_{i+1}^L) \, Q_i^L$, so $q_t^L \cdots q_{i+1}^L = Q_t^L \, \overline{Q_i^L}$ (unit quaternions invert by conjugation). Likewise $Q_t^R = (q_t^R \cdots q_{i+1}^R) \, Q_i^R$, so $q_t^R \cdots q_{i+1}^R = Q_t^R \, \overline{Q_i^R}$; conjugating reverses order and negates, giving $\overline{q_{i+1}^R} \cdots \overline{q_t^R} = \overline{Q_t^R \, \overline{Q_i^R}} = Q_i^R \, \overline{Q_t^R}$. This proves Equation 23.

Substituting Equation 23 into Equation 22,

$$y_t = \sum_{i=1}^{t} \mathrm{Re}\left(\bar{c}_t \left(\prod_{i+1}^{s=t} e^{a_s \Delta_s} \right) Q_t^L \, \overline{Q_i^L} \; w_i \; Q_i^R \, \overline{Q_t^R} \right). \tag{24}$$

Since $\mathrm{Re}(pq) = \mathrm{Re}(qp)$ for $p, q \in \mathbb{H}$, and $Q_t^R \, \overline{Q_t^R} = 1$, we may cyclically move $\overline{Q_t^R}$ to the front:

$$y_t = \sum_{i=1}^{t} \mathrm{Re}\left(\overline{Q_t^R} \, \bar{c}_t \, Q_t^L \left(\prod_{i+1}^{s=t} e^{a_s \Delta_s} \right) \overline{Q_i^L} \; w_i \; Q_i^R \right). \tag{25}$$

Define $\bar{\tilde{c}}_t := \overline{Q_t^R} \, \bar{c}_t \, Q_t^L$ and $\tilde{w}_i := \overline{Q_i^L} \, w_i \, Q_i^R$, matching the definition $\tilde{h}_t := \overline{Q_t^L} \, h_t \, Q_t^R$ of the main text. Then Equation 25 reads

$$y_t = \mathrm{Re}\left(\bar{\tilde{c}}_t \sum_{i=1}^{t} \left(\prod_{i+1}^{s=t} e^{a_s \Delta_s} \right) \tilde{w}_i \right) = \mathrm{Re}(\bar{\tilde{c}}_t \, \tilde{h}_t) = \tilde{\mathbf{c}}_t^{\top} \, \tilde{\mathbf{h}}_t \tag{26}$$

Finally, $\overline{Q_i^L} \, q_i^L = \overline{Q_{i-1}^L}$ and $\overline{q_i^R} \, Q_i^R = Q_{i-1}^R$ give

$$\tilde{w}_i = \overline{Q_i^L} \, w_i \, Q_i^R = (1 - \lambda_i) \Delta_i \, e^{a_i \Delta_i} \, \tilde{b}_{i-1} x_{i-1} + \lambda_i \Delta_i \, \tilde{b}_i x_i,$$

so the sum $\tilde{h}_t = \sum_{i=1}^{t} \big(\prod_{i+1}^{s=t} e^{a_s \Delta_s} \big) \tilde{w}_i$ is the unrolled form of Equation 14, and Equation 26 is Equation 15. □

D.5 Proofs of Section 3.5 (Tracking Finite Groups)

Conventions. We use the objects of Section 3.5: the alphabet $\Sigma \subseteq G$, the words $\omega \in \Sigma^*$, the token rotation $\rho(\kappa) \in \mathrm{SO}(4)^{n_{\mathbb{H}}}$, the word rotation $\rho(\omega)$ and the group element $g(\omega)$, with $\rho(\varnothing) := \mathbf{I}$ and $g(\varnothing) := 1$ for the empty word. By associativity both products reverse concatenation: $\rho(\omega\omega') = \rho(\omega') \, \rho(\omega)$ and $g(\omega\omega') = g(\omega') \, g(\omega)$. Two terms of Equation 12 are set aside: the scalar factor $e^{a_k \Delta_k}$, which commutes with every rotation and leaves the direction of the state unchanged, and the input term, which is not part of the transition. The state-tracking capacity of a linear recurrence is decided by its accumulated transitions along a word (Krohn & Rhodes, 1965; Merrill et al., 2024; Sarrof et al., 2024). The function ψ of Definition 3 recovers the group element from the rotation of a word. The definition asks only that it exist, not that the readout y_k of Equation 13 compute it.

Two hypotheses of Definition 3 carry the theory. The rotation depends on the token: in a time-invariant SSM such as S4 (Gu et al., 2022) every position applies the same transition, the rotations of all words are the powers of one rotation, and only cyclic groups can be tracked, whereas input dependence (Gu & Dao, 2024) lets every token choose its own transition. The set of word rotations is finite: this is the finite-state hypothesis of the diagonal case (Shakerinava et al., 2026), and without it two generic rotations generate a free group, so the rotation of a word would determine the word itself and every finite group would be tracked, by unbounded precision rather than by structure.

Preliminaries. A *section* of a group H, called a *subquotient* (Bergman, 2026, Convention 6.1(ii)), is a quotient of a subgroup $K \leq H$, that is, the image of a surjective homomorphism $K \twoheadrightarrow G$. We call G a *finite section* of H when K can be taken finite. Since a subgroup of a finite subgroup is finite, the finite sections of H are exactly the sections of its finite subgroups. We rely on four standard facts (Dummit & Foote, 2004, Sections 3.1 to 3.3 and 4.6): the *First Isomorphism Theorem*, that the image of a normal subgroup under a surjective homomorphism is normal, that A_5 is simple and not abelian, and that the normal subgroups of S_5 are 1, A_5 and S_5, so that the centre of S_5 is trivial.

The Group of Token Rotations. Let Γ denote the group from which the token rotations of a layer are selected. Proposition 7 shows that $\Gamma = \mathrm{SO}(4)^{n_{\mathbb{H}}}$ for the quaternionic blocks. The real and complex blocks are special cases of the quaternionic blocks and are treated in the proof of Theorem 1. We first relate tracking to Γ.

Proposition 6 (Division and the Word Problem)**.** *Let the token rotations of a layer lie in a group Γ, and let G be a finite group. If the layer tracks G, then G is a finite section of Γ. If every assignment of elements of Γ to the tokens is attainable by some choice of weights, then also every finite group G that is a finite section of Γ is tracked by the layer.*

Proof. **The layer tracks $G \Rightarrow G$ is a finite section of Γ:** By Definition 3 the rotations $\rho(\omega)$ of all words form a finite set H. It contains $\rho(\varnothing) = \mathbf{I}$ and is closed under products, since $\rho(\omega')\rho(\omega) = \rho(\omega\omega')$, so $H \leq \Gamma$ is a finite subgroup. The function ψ of Definition 3 is a homomorphism $H \to G$, as

$$\psi(\rho(\omega')\rho(\omega)) = \psi(\rho(\omega\omega')) = g(\omega\omega') = g(\omega')\, g(\omega)$$

and it is surjective, as the one-token word (κ) demands

$$\psi(\rho((\kappa))) = g((\kappa)) = \kappa$$

for every $\kappa \in \Sigma$. The image $\psi(H)$ is a subgroup of G that contains the generating set Σ, so $\psi(H) = G$, and by the First Isomorphism Theorem $H/\ker\psi \cong G$. Hence G is a quotient of the finite subgroup $H \leq \Gamma$, a finite section of Γ.

G is a finite section of $\Gamma \Rightarrow$ the layer tracks G: Let G be a finite section of Γ. By the definition of a finite section there are a finite subgroup $K \leq \Gamma$ and a surjective homomorphism $\psi\colon K \twoheadrightarrow G$. Choose $\rho(\kappa) \in \psi^{-1}(\kappa) \subseteq \Gamma$ for every token κ. By hypothesis every assignment of elements of Γ to the tokens is attainable, so some choice of weights attains this one. Then every $\rho(\omega)$ lies in the finite group K, and for every word $\omega = (\kappa_1, \ldots, \kappa_k)$

$$\psi(\rho(\omega)) = \psi\big(\rho(\kappa_k)\cdots\rho(\kappa_1)\big) = \psi(\rho(\kappa_k))\cdots\psi(\rho(\kappa_1)) = \kappa_k\cdots\kappa_1 = g(\omega),$$

by the definition of $\rho(\omega)$, because ψ is a homomorphism, by the choice $\rho(\kappa) \in \psi^{-1}(\kappa)$, and by the definition of $g(\omega)$. Hence the layer tracks G by Definition 3. □

In the language of automata the proposition is the division relation of automata theory (Krohn & Rhodes, 1965), applied to the automaton whose states are the rotations of the words. Whether a layer tracks G is now the question of whether G is a finite section of Γ. Next we compute Γ for the quaternionic blocks.

Proposition 7 (Rotations of the Quaternionic Blocks)**.** *Let a layer of Icosian have width $d \geq 6n_{\mathbb{H}} + 1$, let $\mathbf{W}$ have rank $6n_{\mathbb{H}}$ and let $\mathbf{w}_\Delta$ lie outside the row space of $\mathbf{W}$. Then the rotation $\rho(\kappa)$ of a token, as a function of its embedding $\mathbf{x}(\kappa) \in \mathbb{R}^d$, is a surjection $\mathbb{R}^d \twoheadrightarrow \mathrm{SO}(4)^{n_{\mathbb{H}}}$.*

Proof. For a token with embedding $\mathbf{x} = \mathbf{x}(\kappa)$ the rotation $\rho(\kappa)$ is the composition

$$\mathbb{R}^d \xrightarrow{\mathbf{x} \mapsto (\mathbf{W}\mathbf{x},\, \Delta(\mathbf{x}))} \mathbb{R}^{6n_{\mathbb{H}}} \times (0,\infty) \xrightarrow{E} (S^3 \times S^3)^{n_{\mathbb{H}}} \xrightarrow{\mathcal{R}^{n_{\mathbb{H}}}} \mathrm{SO}(4)^{n_{\mathbb{H}}},$$

where $\mathcal{R}^{n_{\mathbb{H}}}$ applies the double cover of Proposition 1 on every block and, with $\mathbf{w}^{(m)} \in \mathbb{R}^6$ the coordinates of block m in $\mathbf{w} = \mathbf{W}\mathbf{x}$,

$$\Delta(\mathbf{x}) := \mathrm{softplus}(\mathbf{w}_\Delta^\top \mathbf{x} + \beta_\Delta),$$
$$E(\mathbf{w}, \Delta) := \Big(\exp\big(\Delta\, \varphi(\ell(\mathbf{w}^{(m)}))\big),\ \exp\big(\Delta\, \varphi(r(\mathbf{w}^{(m)}))\big)\Big)_{m=1}^{n_{\mathbb{H}}}$$

is the quaternion pair of Equation 16 on every block. We show that each arrow is surjective.

Step 1: the third arrow $\mathcal{R}^{n_{\mathbb{H}}}\colon (S^3 \times S^3)^{n_{\mathbb{H}}} \to \mathrm{SO}(4)^{n_{\mathbb{H}}}$ is surjective. This holds since $\mathcal{R}$ is surjective on every block (Proposition 1).

Step 2: the second arrow at fixed step size, $E(\cdot, 2)\colon \mathbb{R}^{6n_{\mathbb{H}}} \to (S^3 \times S^3)^{n_{\mathbb{H}}}$, is surjective. Every unit quaternion is $\exp(\eta)$ for an imaginary η with $\|\eta\| \leq \pi$ (Equation 8), so $\eta/2$ lies in the open ball

of radius π, which is the image of the bijection $\varphi\colon \mathbb{R}^3 \to \{\|\eta\| < \pi\}$. Since $\mathbf{w} \mapsto (\ell(\mathbf{w}), r(\mathbf{w})) = \frac{1}{2}(v-u,\ v+u)$ is a linear bijection $\mathbb{R}^6 \to \mathbb{R}^3 \times \mathbb{R}^3$ on every block (Section 3.4), every element of $(S^3 \times S^3)^{n_{\mathbb{H}}}$ equals $E(\mathbf{w}, 2)$ for some $\mathbf{w} \in \mathbb{R}^{6n_{\mathbb{H}}}$. Any $\Delta > 1$ serves here, whereas for $\Delta \leq 1$ the quaternion -1 is not reached.

Step 3: the first arrow $\mathbf{x} \mapsto (\mathbf{W}\mathbf{x}, \Delta(\mathbf{x}))$ from $\mathbb{R}^d$ to $\mathbb{R}^{6n_{\mathbb{H}}} \times (0, \infty)$ is surjective. Given $\mathbf{w}$ and a target step size, the rank of $\mathbf{W}$ gives $\mathbf{x}_0$ with $\mathbf{W}\mathbf{x}_0 = \mathbf{w}$, and $\mathbf{w}_\Delta \notin \operatorname{row}(\mathbf{W}) = (\ker \mathbf{W})^\perp$ gives $\mathbf{x}_1 \in \ker \mathbf{W}$ with $\mathbf{w}_\Delta^\top \mathbf{x}_1 \neq 0$. Along the line $\mathbf{x}_0 + t\mathbf{x}_1$ the value $\mathbf{W}\mathbf{x} = \mathbf{w}$ is constant and Δ runs through $(0, \infty)$, so some t gives the target step size. Composing the three surjections, every element of $\mathrm{SO}(4)^{n_{\mathbb{H}}}$ is the rotation of some embedding, with $\Delta = 2$ one choice of a step size above 1. □

Step 3 uses that the embedding $\mathbf{x}$ can be any vector of $\mathbb{R}^d$. In the network the input of the layer passes through a normalization first, which restricts it to a scaled sphere, and the proposition does not cover that case.

We now show that S_5 is not a finite section of $\mathrm{SO}(4)^n$.

Lemma 1 (S_5 Is Not a Finite Section of $\mathrm{SO}(4)^n$)**.** *Let $n \geq 1$. Then S_5 is not a finite section of $\mathrm{SO}(4)^n$.*

Proof. Suppose that S_5 is a finite section of $\mathrm{SO}(4)^n$: there are a finite subgroup $H \leq \mathrm{SO}(4)^n$ and a surjective homomorphism $\psi\colon H \twoheadrightarrow S_5$.

Step 1: lift to unit quaternions. The double cover $\mathcal{R}^n\colon (S^3)^{2n} \twoheadrightarrow \mathrm{SO}(4)^n$ is a surjective homomorphism, and by Proposition 1 every rotation has exactly 2^n preimages, one sign per block. Let $\tilde{H} := (\mathcal{R}^n)^{-1}(H)$ be the preimage of H. The preimage of a subgroup under a homomorphism is a subgroup, and $\tilde{H}$ has $2^n\,|H|$ elements, so $\tilde{H}$ is a finite subgroup of $(S^3)^{2n}$ with $\mathcal{R}^n(\tilde{H}) = H$. Hence $\psi \circ \mathcal{R}^n$ is a surjection from $\tilde{H}$ to S_5. Let $P_1, \ldots, P_{2n} \leq S^3$ be the images of $\tilde{H}$ under the coordinate projections, each a finite subgroup. In a diagram,

$$\tilde{H} \ \leq\ P_1 \times \cdots \times P_{2n} \ \leq\ (S^3)^{2n}, \qquad \tilde{H} \xrightarrow{\mathcal{R}^n} H \xrightarrow{\psi} S_5.$$

Step 2: one factor suffices. S_5 is a section of $P_1 \times \cdots \times P_{2n}$. Whenever S_5 is a section of a finite direct product, it is a section of one factor (Bergman, 2026, Proposition 6.6(i) and Example 6.7). Hence S_5 is a section of some finite subgroup of S^3, that is, a quotient of a finite subgroup $F \leq S^3$.

Step 3: classification. The finite subgroups of S^3 are the cyclic groups, the binary dihedral groups and the binary tetrahedral, octahedral and icosahedral groups $2T$, $2O$ and $2I$ of orders 24, 48 and 120 (Conway & Smith, 2003, Theorem 12). A cyclic group has only abelian quotients. A binary dihedral group has a cyclic subgroup of index two, whose image in S_5 would be an abelian normal subgroup of index at most two, yet neither A_5 nor S_5 is abelian. The groups $2T$ and $2O$ have fewer than 120 elements. Hence $F = 2I$.

Step 4: centres. A surjection $2I \twoheadrightarrow S_5$ between groups of order 120 is an isomorphism. But -1 is a central element of $2I$ other than the identity, and the centre of S_5 is trivial. □

We can now prove Theorem 1. Its statement allows any number of real blocks, which do not rotate.

Proof of Theorem 1. **Part (i):** Since $d \geq 6n_{\mathbb{H}} + 1$, we may choose $\mathbf{W}$ of rank $6n_{\mathbb{H}}$ and $\mathbf{w}_\Delta$ outside its row space, and we set the angle weights of the complex blocks to zero, so that the complex blocks do not rotate. By Proposition 7 the token rotations can then be given any values in $\mathrm{SO}(4)^{n_{\mathbb{H}}}$ through the embeddings $\mathbf{x}(\kappa)$, one per token. Hence every assignment of elements of $\mathrm{SO}(4)^{n_{\mathbb{H}}}$ to the tokens is attainable, and by the second statement of Proposition 6 the layer tracks every finite section of $\mathrm{SO}(4)^{n_{\mathbb{H}}}$.

Part (i), in particular: For $n_{\mathbb{H}} = 1$ every finite subgroup of $\mathrm{SO}(4)$ is a finite section of $\mathrm{SO}(4)$ and hence tracked, and A_5 is a finite subgroup of $\mathrm{SO}(3) \leq \mathrm{SO}(4)$.

Part (ii): Let d and the weights be arbitrary, and count now the rotations of the complex blocks as part of the rotation of a token. Every token rotation lies in $\mathrm{SO}(4)^{n_{\mathbb{H}}} \times \mathrm{SO}(2)^{n_{\mathbb{C}}}$, since the quaternionic blocks rotate in $\mathrm{SO}(4)$, the complex blocks in $\mathrm{SO}(2)$ and the real blocks not at all. As $\mathrm{SO}(2) \leq \mathrm{SO}(4)$, this group is a subgroup of $\mathrm{SO}(4)^{n_{\mathbb{H}}+n_{\mathbb{C}}}$. Proposition 6 with $\Gamma = \mathrm{SO}(4)^{n_{\mathbb{H}}+n_{\mathbb{C}}}$ shows that every tracked group is a finite section of $\mathrm{SO}(4)^{n_{\mathbb{H}}+n_{\mathbb{C}}}$.

Part (ii), in particular: Lemma 1 with $n = n_{\mathbb{H}} + n_{\mathbb{C}}$ shows that S_5 is not a finite section of $\mathrm{SO}(4)^{n_{\mathbb{H}}+n_{\mathbb{C}}}$, so the layer does not track it. □