WHY DEPTH MATTERS IN PARALLELIZABLE SEQUENCE MODELS: A LIE ALGEBRAIC VIEW

Anonymous authors
Paper under double-blind review

ABSTRACT

Scalable sequence models, such as Transformer variants and structured state-space models, often trade expressivity power for sequence-level parallelism, which enables efficient training. Here we examine the bounds on error and how error scales when models operate outside of their expressivity regimes using a Lie-algebraic control perspective. Our theory formulates a correspondence between the depth of a sequence model and the tower of Lie algebra extensions. Echoing recent theoretical studies, we characterize the Lie-algebraic class of constant-depth sequence models and their corresponding expressivity bounds. Furthermore, we analytically derive an approximation error bound and show that error diminishes exponentially as the depth increases, consistent with the strong empirical performance of these models. We validate our theoretical predictions using experiments on symbolic word and continuous-valued state-tracking problems.

1 INTRODUCTION

Most sequence problems, from symbolic processing tasks like natural language, mathematics, and coding, to the understanding of physical-world dynamics, are fundamentally *order-sensitive*. Yet, highly scalable sequence models often achieve parallelism via *order symmetry*. Notably, self-attention layers in transformers (Vaswani et al., 2017), and crucial components of diagonal structured state-space models (Orvieto et al., 2023; Gu & Dao, 2024) or linear transformer variants (Katharopoulos et al., 2020; Schmidhuber, 1992; Yang et al., 2024a) exhibit order symmetry in their layer inputs. This strong structural bias has inspired the study of the expressivity limits of such models (Hahn, 2020; Merrill et al., 2020). In particular, recent theoretical work has identified reasoning and state-tracking problems (Kim & Schuster, 2023; Merrill et al., 2024; Grazzi et al., 2025) that are provably unsolvable by constant-depth transformers or diagonal structured state-space models.

Despite these theoretical bounds, deep scalable sequence models remain the most empirically successful architectures, especially in large-scale language modeling. This discrepancy motivates a quantitative question: *How badly does a model operate when applied to tasks it provably cannot solve exactly?* Quantifying this gap matters because more efficient models would be preferable if the error remains tolerable. Although prior work (Walker et al., 2025) provides related intuition, the error-expressivity scaling laws have not, to our knowledge, been clearly derived.

To answer this question, we leverage Lie theory (Iserles et al., 2000; Agrachev & Sachkov, 2004). Geometrically, Lie theory measures the order-sensitivity of operations: the state discrepancy caused by swapping the order of events (see Fig. 1 for an intuitive illustration). This provides both a *hierarchy* for classifying tasks and models under their *degree* of order sensitivity, and a method to quantify approximation error as the *mismatch* between target and model states under the same input.

We show that order-symmetric models incur an unavoidable approximation error when the task is order sensitive. We then identify depth as a mechanism that extends the expressivity of parallelizable sequence models, decreasing this error. Overall, our work characterizes depth-dependent error-expressivity behavior, and provides a structural lens for model choice based on the task structure.

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.

2 MATHEMATICAL PRELIMINARIES

Here we introduce the mathematical framework used throughout the paper. We formalize order sensitivity and approximation error to establish our theories in Sec. 3. Readers primarily interested in the main results and their practical implications may wish to proceed directly to Sec. 3 and Sec. 4.

2.1 LIE GROUPS AND LIE ALGEBRAS

A *Lie group* G is a smooth manifold equipped with a group structure under smooth multiplication. Its tangent space at the identity characterizes a *Lie algebra* $\mathfrak{g}$. $\mathfrak{g}$ is a vector space equipped with a bilinear and antisymmetric Lie bracket operation $[\cdot,\cdot]:\mathfrak{g}\times\mathfrak{g}\to\mathfrak{g}$ satisfying the Jacobi identity:

$$[[X,Y],Z]=[X,[Y,Z]]-[Y,[X,Z]],\quad X,Y,Z\in\mathfrak{g}.$$

A canonical example of the Lie bracket is a matrix commutator. For matrices $\mathbf{A},\mathbf{B}\in\mathbb{R}^{n\times n}$, the Lie bracket corresponds to the matrix commutator $[\mathbf{A},\mathbf{B}]=\mathbf{AB}-\mathbf{BA}$.

Given a vector space spanned by a collection of vector fields $\mathcal{F}:=\{F_i\}$, the smallest Lie algebra is denoted as $\mathrm{Lie}(\mathcal{F})$ (Iserles et al., 2000). Unless otherwise mentioned, Lie algebras are finite-dimensional over $\mathbb{R}$.

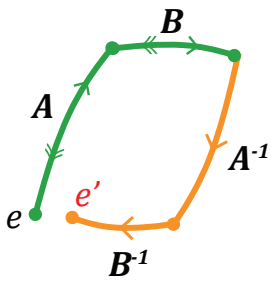


Figure 1: Geometric intuition of Lie theory. Initialized at e, consider *actions* $\mathbf{A}$ and $\mathbf{B}$ sequentially, and *undo* actions $\mathbf{B}^{-1}$ then $\mathbf{A}^{-1}$. Composition of actions $\mathbf{ABB}^{-1}\mathbf{A}^{-1}$ would return to e. However, *switching* the order of undoing, by using $\mathbf{ABA}^{-1}\mathbf{B}^{-1}$, might not return to e, incurring a discrepancy by landing on point e'. Lie theory provides a measure of the potential offset between e and e'.

Classes of Lie algebras Here we classify Lie algebras based on their *degree* of order sensitivity. The *derived* series of $\mathfrak{g}$ is the inductive sequence of *commutator ideals*:

$$\mathfrak{g}^{(0)}=\mathfrak{g},\quad \mathfrak{g}^{(i)}=[\mathfrak{g}^{(i-1)},\mathfrak{g}^{(i-1)}].$$

If $\mathfrak{g}^{(j)}=0$ with the smallest finite number j, $\mathfrak{g}$ is classified as a *solvable* Lie algebra with derived length j. Similarly, the *lower central* series of $\mathfrak{g}$ is a sequence of ideals spanned by nested commutators:

$$\mathfrak{g}^{0}=\mathfrak{g},\quad \mathfrak{g}^{i}=[\mathfrak{g},\mathfrak{g}^{i-1}].$$

If $\mathfrak{g}^{c}=0$ with the smallest finite number c, $\mathfrak{g}$ is *nilpotent* with class c. An *abelian* Lie algebra is the special case with $[\mathfrak{g},\mathfrak{g}]=0$ (Hall, 2015).

These Lie algebras are related by a structural hierarchy. Abelian Lie algebras are nilpotent, and all nilpotent Lie algebras are solvable. Furthermore, one can *glue* abelian Lie algebras to construct a solvable one. A Lie algebra *extension* is a short exact sequence $0\to\mathfrak{a}\to\mathfrak{g}\to\mathfrak{h}\to 0$ where $\mathfrak{a}$ is an *ideal* of $\mathfrak{g}$ and $\mathfrak{h}\cong\mathfrak{g}/\mathfrak{a}$. If $\mathfrak{g}$ has derived length j, $\mathfrak{g}^{(j-1)}$ is an abelian ideal by definition, and $\mathfrak{g}/\mathfrak{g}^{(j-1)}$ is again solvable with derived length $j-1$. Thus, a solvable Lie algebra is a *tower* of abelian algebra extensions (App.A.3).

2.2 STATE-SPACE MODELS AND THE LIE EQUATION

To connect Lie theory with sequence models, we formalize state-space models as controlled dynamical systems (App. B.1). We will observe how the Lie algebra characterizes a system.

Definition 2.1 (State space model)**.** A *state space model* (SSM) is an affine vector field on Euclidean *state space* $\mathcal{M}=\mathbb{R}^n$ and *input space* $\mathcal{X}$ such that:

$$F_x:\mathcal{M}\to T\mathcal{M},\quad F_x(h):=\mathbf{A}(x)h+\mathbf{b}(x). \tag{1}$$

$\mathbf{A}:\mathcal{X}\to\mathbb{R}^{n\times n}$ and $\mathbf{b}:\mathcal{X}\to\mathbb{R}^n$ are real analytic (Sontag, 1998). We call $\mathbf{A}$ the *generator* of the SSM. We denote SSM as $\mathbf{S}$. For $\mathfrak{g}=\mathrm{Lie}(\{F_x\}_{x\in\mathcal{X}})$, we say $\mathbf{S}$ has class of $\mathfrak{g}$ and denote it as $\mathbf{S}_{\mathfrak{g}}$.

We assume standard control-theoretic regularity and minimality (App. B.1). With a locally bounded and integrable path $x(\cdot):[0,T]\to\mathcal{X}$, one can reintroduce time in Eq. 1 and construct an ODE. $\mathbf{S}$ is *initialized* when the initial state $h(0)$ is fixed. In what follows, we focus on the state evolution.

Definition 2.2 (Flow)**.** The evolution of Eq. 1 is governed by the *flow* operator $\psi_x(t,\tau) : \mathcal{M} \to \mathcal{M}$:

$$h(t) = \psi_x(t,\tau)\big(h(\tau)\big) = \Phi_x(t,\tau)h(\tau) + \int_\tau^t \Phi_x(t,s)\mathbf{b}(x(s))\,ds,$$

where Φ is the state-transition matrix. Φ is an evolution operator of the homogeneous flow:

$$\Phi_x(t,\tau) = \mathcal{T}\exp\int_\tau^t \mathbf{A}(x(s))\,ds, \tag{2}$$

where $\mathcal{T}\exp$ denotes the time-ordered exponential (Agrachev & Sachkov, 2004; Sontag, 1998).

Therefore, ψ describes the full affine flow, while Φ focuses on the linear part of the SSM dynamics.

From state-centric to flow-centric view Rather than tracking a single state, one can track *entire flows*. This approach bridges Lie algebras and SSMs. We begin with the homogeneous flow:

$$\dot\Phi(t,0) = \mathbf{A}(x(t))\Phi(t,0), \quad \Phi(0,0) = I. \tag{3}$$

It turns out that this is a *Lie equation* (Iserles et al., 2011; Jurdjevic & Sussmann, 1972):

Definition 2.3 (Controlled Lie equation)**.** A linear controlled Lie equation is a matrix ODE:

$$\dot\Psi(t) := \mathbf{A}(x(t))\Psi(t), \quad \Psi(0) = I, \tag{4}$$

where $\mathrm{Lie}(\{\mathbf{A}(x)\}) = \mathfrak{h} \subseteq \mathfrak{gl}(n,\mathbb{R})$, $\Psi \in H \subseteq \mathrm{GL}(n,\mathbb{R})$, and Lie group H locally integrates $\mathfrak{h}$.

It is immediately clear that Eq. 3 aligns with Def. 2.3: Φ is a local curve on H, and Eq. 4 is a *flow-centric* ODE tracking all sets of local evolutions rather than a single state of SSM.

Definition 2.4 (Lift)**.** Consider an initialized $\mathbf{S}_\mathfrak{h}$ with h_0. *Lifted* $\mathbf{S}_\mathfrak{h}$ is the controlled dynamical system on the augmented flow space $H \times \mathcal{M}$ that pairs the Lie equation (Eq. 4) with h_0:

$$\dot\Psi(t) = \mathbf{A}(x(t))\Psi(t),\ \dot h(t) = 0,\ \big(\Psi(0), h(0)\big) = \big(I, h_0\big), \quad \Psi \in H,\ h_0 \in \mathcal{M}$$

Observe that lifted $\mathbf{S}_\mathfrak{h}$ evolves on the Lie group H, not on the original state space $\mathcal{M}$. The state is recovered by the projection $(\Psi, h_0) \mapsto \Psi h_0$.

For affine flows, we can absorb $\mathbf{b}(x)$ by state augmentation and apply the same construction:

Definition 2.5 (Homogenization)**.** *Homogenization* is exact but non-minimal representation of an SSM as a linear homogeneous ODE on $\mathbb{R}^{n+1}$ via state augmentation. For $\tilde h = (h;1) \in \mathbb{R}^{n+1}$:

$$\dot{\tilde h}(t) = \tilde{\mathbf{A}}(x(t))\tilde h(t), \quad \tilde{\mathbf{A}}(x) = \begin{pmatrix} \mathbf{A}(x) & \mathbf{b}(x) \\ 0 & 0 \end{pmatrix}, \quad \psi_x(t,\tau)(h(\tau)) = \Big[\tilde\Phi_x(t,\tau)\tilde h(\tau)\Big]_{1:n}.$$

Therefore, ψ of an SSM can be represented as Φ of a homogeneous ODE (Krener, 1977).

For the clarity, we use $\mathfrak{g}_{\text{lin}} = \mathrm{Lie}(\{\mathbf{A}(x)\})$ and $\mathfrak{g}_{\text{aff}} = \mathrm{Lie}(\{\tilde{\mathbf{A}}(x)\})$. As homogenization is exact, we use $\mathfrak{g}$ for $\mathfrak{g}_{\text{aff}}$ when clear. Our flow-centric constructions absorb $\mathbf{b}(x)$ without loss of generality.

Restricted SSM Aside from the Lie class of the SSM, we say $\mathbf{S}_\mathfrak{g}$ is *restricted* if $\mathfrak{g}_{\text{lin}}$ is abelian. This is to clarify the role of the symmetry breaking $\mathbf{b}(x)$. Notably, diagonal structured SSMs (Sec. 4) are restricted SSMs under piecewise-constant inputs (Orvieto et al., 2023). As $\mathfrak{g}_{\text{aff}}$ is abelian extension of $\mathfrak{g}_{\text{lin}}$, any abelian SSM is restricted, and restricted SSM is solvable. If not restricted, $\mathbf{S}_\mathfrak{g}$ is *general*.[1]

2.3 Simulation and approximation error

Studying expressivity of a model class is tied to *simulation*: whether one system can reproduce another's behavior (Liu et al., 2023). Analogously, we use a *state-based* notion of simulation.

We say $\mathbf{S}_0$ simulates $\mathbf{S}_1$ if a smooth surjective submersion $\varphi : \mathcal{M}_0 \to \mathcal{M}_1$ maps $\mathbf{S}_0$ state evolution to $\mathbf{S}_1$ states under any inputs (Sussmann, 1976; Krener, 1977). When exact simulation is impossible, we quantify the state mismatch after evolution.[2] It is clear that the lifted Lie equation exactly simulates the original SSM.

[1] Both notions refer to the SSM given as Def. 2.1, so no ambiguity arises.

[2] Details are provided in App. B.2.

Magnus expansion Given the Lie-algebraic perspective of SSMs, we now introduce a quantitative tool for the approximation error. The *Magnus expansion* decomposes Φ via exponential of iterated Lie brackets (App. B.4), providing an explicit measure of order-relevant errors. The n-th order Magnus term, Ω_n, consists of $(n-1)$-fold nested Lie bracket operations of $\mathbf{A}(x)$.

Definition 2.6 (Commutator mass)**.** Given a fixed input path x over the interval $[0, T]$, we define *commutator mass* of $\mathbf{S}$ as the norm of the second-order Magnus term, $\|\Omega_2\|$:

$$\Omega_2 = \frac{1}{4} \iint_{[0,T]^2} \operatorname{sgn}(t_1 - t_2)[\mathbf{A}(x(t_1)), \mathbf{A}(x(t_2))]\, dt_1 dt_2. \tag{5}$$

We work on small windows in which the Magnus series converges.

2.4 Deep structure

Most practical applications of artificial neural networks leverage deep layered structures with additional pointwise computation (e.g. MLP) between layers. To extend our theory, here we discuss dynamical systems with deep layered structure (Krener, 1977).

Definition 2.7 (Deep SSM)**.** The *stack* of $(\mathbf{S}_0, \mathbf{S}_1)$ is defined with a smooth map $v : \mathcal{M}_0 \times \mathcal{X} \to \mathcal{X}_1$:

$$\mathcal{M} \cong \mathcal{M}_0 \times \mathcal{M}_1,\ x_1(t) = v(h_0(t), x(t)),$$
$$\dot{h}_0(t) = \mathbf{A}_0(x(t))h_0(t) + \mathbf{b}_0(x(t)), \quad \dot{h}_1(t) = \mathbf{A}_1(x_1(t))h_1(t) + \mathbf{b}_1(x_1(t)).$$

A deep SSM is obtained inductively:

$$x_i(t) = v_i(h_{i-1}(t), \cdots, h_0(t), x(t)), \quad x_0(t) = x(t), \quad \dot{h}_i(t) := \mathbf{A}_i(x_i(t))h_i(t) + \mathbf{b}_i(x_i(t)),$$

where $v_i : \left(\prod_{j=0}^{i-1} \mathcal{M}_j\right) \times \mathcal{X} \to \mathcal{X}_i$ is smooth.

We refer to each local SSM as the i-th layer. An L-layer SSM is said to be restricted when all of its layers are restricted. This convention applies to the other classes as well.

2.5 Word problems

Lastly, we introduce *word problems*, one of the standard test beds for sequence model expressivity.

Definition 2.8 (Word problem (Merrill et al., 2024; Liu et al., 2023))**.** Given a finite alphabet set A, a *word* is an element of the free monoid A^* on A. For a finite monoid M generated by A, the word evaluation map f is the unique monoid homomorphism $f : A^* \to M$ determined by sending each generator in A to itself in M. Solving the word problem for M is to learn the word evaluation map and reduce any word $a_1 a_2 \dots a_t$ to $a_1 \cdot a_2 \cdot \dots \cdot a_t \in M$ under multiplication $\cdot$.

Consider each *alphabet symbol* as a *primitive action*, such as a permutation, rotation, or state transition operation (Fig. 1). A *word* is then a sequence of *action commands* evaluated by composing primitive actions in order. A sequence of matrix products and a rollout of action-conditioned state transitions in a Markov decision process (MDP) are examples of word evaluation. The *parity* problem is the word problem for the cyclic group C_2, with $\cdot$ corresponding to addition modulo 2.

Thus, word problems are symbolic state-tracking tasks with an explicit algebraic class. As symbols arrive, the model must update its state according to the composition rule and output the resulting element. From the control-theoretic perspective, an automaton solving a word problem is a linear dynamical system with symbol-specific flow. This view incorporates word problems into our SSM framework. We list classes of exemplary word problems and related concrete problems in Table 1.

3 Theory

Our theory has two major parts. Thm. 3.2 derives an approximation error bound for 1-layer SSMs. Thm. 3.4 provides theoretical ground on how depth extends expressivity via Lie algebra extension. Cor. 3.7 and Prop. 3.8 construct an upper bound for depth-dependent approximation errors.

Table 1: Class of representative word problems on finite group.

Class	Finite group example	Related problems
Abelian	C_2, C_3, C_{60}	Parity; Cyclic gridworld
Nilpotent	D_8, H_3	Quantum spin
Solvable	S_3, A_4, S_4	2D affine transformation
Non-solvable	A_5, S_5	Rubik's cube; 3D rotation

3.1 Expressivity bounds of a single layer

Here our theory aligns with prior work (Merrill et al., 2024; Liu et al., 2023; Shakerinava et al., 2026). We begin with the most basic case.

Lemma 3.1. *No abelian SSM $\hat{\mathbf{S}}$ can simulate a general SSM $\mathbf{S}$.*

Proof is provided in App. C.1. Building on Lemma 3.1, we constructively provide an error bound.

Theorem 3.2. *Let a restricted $\hat{\mathbf{S}}$ approximates general $\mathbf{S}$. There exists an input x on a sufficiently small $[0, T]$ that incurs approximation error scaling with the commutator mass $\|\Omega_2\|$ of $\mathbf{S}$.*

Proof sketch. The order symmetry of $\mathfrak{g}_{\text{lin}}$ generates an irreducible order-sensitive error. This corresponds to the gap between e and e' (Fig. 1). The full proof is detailed in App. C.2. [3] □

Therefore, the symmetry-breaking effect of $\mathbf{b}(x)$ is *not* sufficient to unlock the order-sensitive expressivity. A restricted SSM is solvable but fails to simulate SSMs with non-abelian $\mathfrak{g}_{\text{lin}}$.

3.2 Expressivity bounds of deep structure

Here we discuss how depth mitigates the expressivity limits of restricted SSMs.

Exact representation We begin by clarifying an algebraic class of deep SSMs.

Proposition 3.3. *Restricted k-layer SSMs are solvable with at most $2k$ derived length. Abelian k-layer SSMs are solvable with at most k derived length.*

Proof is provided in App. C.3. Thus, finite-depth restricted SSMs are limited to be solvable.

We now arrive at our core theorem, connecting the depth of SSM to Lie algebra extension.

Theorem 3.4. *Let $0 \to \mathfrak{a} \to \mathfrak{g} \to \mathfrak{h} \to 0$ be a Lie algebra extension. Then, there exists a 2-layer deep SSM with smooth output map that simulates $\mathbf{S}_{\mathfrak{g}}$.*

Proof sketch. From the extension structure, we can construct a 2-layer SSM whose first layer realizes the flow of $\mathfrak{h}$, the second layer realizes the flow of $\mathfrak{a}$, and the smooth output map composes two flows to reconstruct the flow of $\mathbf{S}_{\mathfrak{g}}$. The full proof is in App. C.4. □

An immediate corollary for the order-symmetric structure follows.

Corollary 3.5. *Consider $\mathbf{S}_{\mathfrak{g}}$ such that the derived length of $\mathfrak{g}$ is k. Then, there exists an abelian k-layer SSM plus smooth output map that simulates $\mathbf{S}_{\mathfrak{g}}$.*

Proof is provided in App. C.5. With Prop. 3.3, Cor. 3.5 characterizes the exact expressivity of deep abelian SSMs. Abelian k-layer SSMs can simulate any solvable SSM with derived length up to k, but cannot simulate solvable SSMs with derived length greater than k.

We again use MDPs to build a tangible intuition. As noted in Table 1, different orders of action generally induce different action-dependent transition matrices. Modeling order-sensitive dynamics is critical for a sequence model to represent an MDP or serve as a learned *world model* (Chen et al., 2022). Thm. 3.4 and Cor. 3.5 characterize how many layers are needed to represent *compositions* of actions with the correct transition structure. In this view, each layer tracks a collection of *commuting* components of dynamics, whose composition reconstructs non-commuting transitions.

[3] As local errors can cancel out, local-to-global extension requires an alignment assumption (App. C.2).

Remark 3.6 (Conjecture on restricted SSMs). We have shown the exact expressivity of deep abelian SSMs. Thm. 3.2 suggests that restricted SSMs do not substantially extend expressivity beyond abelian SSMs. Motivated by this, we conjecture that there exists a $\mathbf{S}_{\mathfrak{g}}$ of derived length $k+1$ that cannot be simulated by any restricted k-layer SSM. Thm. 3.2 shows the conjecture holds for $k=1$.

Approximation We now turn our attention to approximation. Cor. 3.5 naturally maps to the well-known control theory literature. On a short interval $[0, T]$, we define the local *generator mass*:

$$\epsilon = \int_0^T \|\mathbf{A}(x(t))\|\, dt < 1 \quad \text{(for sufficiently small } T\text{)}.$$

Corollary 3.7. *For non-solvable $\mathbf{S}_{\mathfrak{g}}$, there exists an abelian k-layer SSM plus smooth output map whose approximation error scales with $\mathcal{O}(\epsilon^{2^k})$.*

proof sketch. One can construct a class c nilpotent Lie algebra $\mathfrak{h}_c$ exactly matching the Magnus term of $\mathfrak{g}$ up to order c. $\mathfrak{h}_c$ has bounded derived length $\lceil \log_2(c+1) \rceil$, thus k-layer abelian SSM class can simulate any nilpotent Lie algebra of class up to $2^k - 1$. Therefore, the first mismatch in Magnus series occurs at order 2^k. Detailed proof is provided in App. C.6. □

Cor. 3.7 is a promising result: although algebraically obstructed, deep structure exponentially mitigates the order-sensitive error.[4]

For word problems, we treat each input symbol as a piecewise-constant input. Under piecewise-constant inputs, the SSMs in our theory can be considered as discrete-time sequence models.

Proposition 3.8. *Any word problem with word length bounded by T can be simulated by an abelian deep SSM with at most $\big(\lfloor \log_2 T \rfloor + 1\big)$-layers and a smooth output map.*[5]

Corollary 3.9. *Consider an alphabet of size $|\mathcal{A}| = n$ and word length bounded by T. Then, the total state space dimension of the abelian deep SSM constructed in Prop. 3.8 exponentially grows in T for fixed n with asymptotic order $\mathcal{O}(n^T/T)$.*

Proof is provided in App. C.7 and App. C.8, respectively. Note that this is the worst-case construction, and the required depth and width can be much smaller depending on the problem structure. For example, 1-layer abelian SSM with state size 2 suffices to simulate the parity problem for arbitrary T. What we construct here is the receptacle of all possible word problems given n and T. Together, Prop. 3.8 and Cor. 3.9 clarify how depth and width play roles in constant length simulation.

4 Experiments

Here we use experiments as diagnostic tests of the theory. We examine symbolic word problems (WPs; Sec. 4.1) covering those that are abelian (C_2, C_3), nilpotent (D_8, H_3), solvable (S_3, S_4) or non-solvable (A_5), as well as a physical rotation prediction problem based on A_5 (Sec. 4.3).

Regarding the model architecture, in addition to the basic transformer (Vaswani et al., 2017), we study three types of diagonal structured SSMs: GLA (Yang et al., 2024a), signed Mamba (Grazzi et al., 2025), and AUSSM (Karuvally et al., 2025). Generators of each models are positive, signed, and complex diagonal matrices, respectively, and, thus, can be viewed as discretized restricted SSMs. DeltaProduct (Siems et al., 2025) is included as a general SSM class reference.

4.1 Length generalization on solvable word problems

We first evaluate sequence models on solvable WPs to test the models' algebraic capacities. Following prior work (Merrill et al., 2024; Siems et al., 2025), the model predicts the finite group element obtained by evaluating the word at each position. Results in Table 2 are summarized as follows.

[4] Cor. 3.7 is fundamentally local, as the error bound evaluation requires the Magnus series to converge. Local-to-global extension follows by composing small windows with no further assumptions (App. C.6).

[5] Similar logarithmic result for transformers was reported in Liu et al. (2023).

Table 2: Length generalization performance measured as sequence-level accuracy for various categories of word problems. Models are trained on length 128 and tested on 256. L denotes the number of layers. DeltaProduct uses the theoretically necessary number of Householder products depending on the problem. The blue color cell highlights perfect generalization. The hyper-parameter search space is in App. D.1. The number of training sequences is 500K except for C_2 which used 100K.

		Abelian		Nilpotent		Solvable	
Model	L	C_2	C_3	D_8	H_3	S_3	S_4
Transformer	1	0.00	0.00	0.00	0.00	0.00	0.00
(+-only) Mamba	1	0.00	0.00	0.00	0.00	0.00	0.00
(+-only) GLA	1	0.00	0.00	0.00	0.00	0.00	0.00
	2	0.00	0.00	0.00	0.00	0.00	0.00
Signed Mamba	1	1.00	0.00	0.00	0.00	0.00	0.00
	2	1.00	0.00	1.00	1.00	*0.19*	0.00
AUSSM	1	1.00	1.00	0.00	0.00	0.00	0.00
	2	1.00	1.00	0.00	0.00	0.00	0.00
DeltaProduct	1	1.00	1.00	1.00	1.00	1.00	1.00

Abelian WPs. We evaluate models on abelian WPs as a smoke test. As any abelian finite group embeds into abelian Lie group, a 1-layer abelian SSM can express abelian WPs in theory.

Empirically, echoing prior work, we observe that transformer, positive diagonal only (denoted as +-only) GLA, and Mamba failed to model the simplest abelian WP, C_2. In contrast, Signed Mamba can solve C_2, while failing on C_3, aligned with Grazzi et al. (2025). AUSSM succeeded at both C_2 and C_3, matching its richer complex-valued diagonal structure.

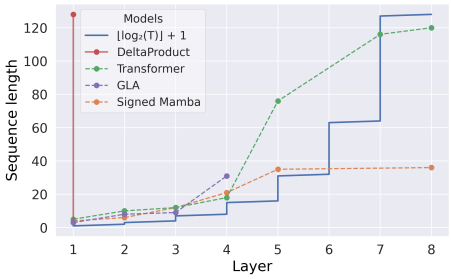


Figure 2: Maximum sequence length on the A_5 WP for each model with varying depths to achieve $> 90\%$ training set accuracy. All models are trained on length up to 128. Deep models (> 4 layers) that failed to outperform shallower models are not shown. T in the legend $\left(\lfloor \log_2 T \rfloor + 1\right)$ is sequence length (i.e., y-axis). The reference curve is the upper bound from Prop. 3.8.

Nilpotent and Solvable WPs. We further evaluate models on solvable but non-abelian WPs. Theoretically, D_8, H_3, and S_3 are extensions of two abelian groups. Following Thm. 3.4, these WPs can be represented by 2-layers of *abelian* SSMs, while 1 layer is insufficient (Thm. 3.2). [6] Consistent with the prediction, none of the 1-layer models except DeltaProduct successfully solved these tasks.

With 2 layers, we observe the inductive bias significantly influences model behavior. Signed Mamba successfully learned D_8 and H_3 and partially generalizes on S_3. In contrast, AUSSM fails to learn a generalizable solution for non-abelian WPs in our setting. This points to the *learnability* issue: our theory describes expressivity, but gradient-based optimization and architecture-specific inductive biases determine whether the solution is easy to find. Signed Mamba has a more restricted eigenvalue range than AUSSM, but empirically performed better in the 2-layer regime.[7]

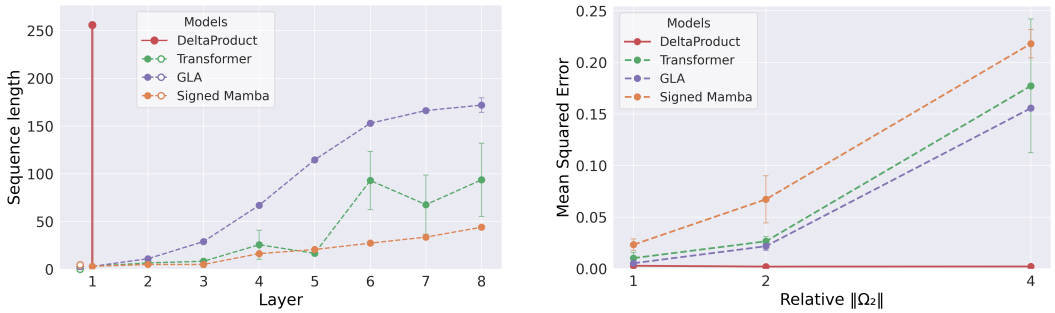
S_4 consists of 3 abelian groups. Rem. 3.6 conjectures 2-layer restricted SSM would not solve S_4.

4.2 Depth vs. length in non-solvable word problems.

Here we focus on A_5 as the smallest non-solvable WP to illustrate how the depth enhances performance beyond each model's expressivity range, in line with Prop. 3.8. Fig. 2 reports the maximum sequence length for which each model achieves $> 90\%$ accuracy at a given depth. We compared results to the theoretical upper bound (Prop. 3.8).

[6] See also Shakerinava et al. (2026).

[7] see further comments on the trainability of AUSSM in App. D.2



(a) Depth-dependent error behavior

(b) Single layer error scaling with commutator mass

Figure 3: Rotation task. All models are trained on length up to 128. Standard error is shown using 3 seeds. DeltaProduct with 4 Householder products is included as a reference. (a) Sequence length before the model error exceeds half of the chance level, with varying depths, tested on length up to 256. Filled marks indicate hidden size $h = 128$ per layer, and hollow marks $h = 1024$. Position-wise curves are in App. D.2. (b) MSE of 1-layer models on commutator probe words measured at the last position, as a function of relative $\|\Omega_2\|$.

We observe that transformer clearly benefits from depth. By varying the depth from 1 to 8, transformer performance improves following the trend of the theoretical bound. However, we observe that deeper models are hard to train; some deep models underperform shallower ones. The 6- and 7-layer Signed Mamba did not improve over the 5-layer one, and 8-layer model performed only slightly better, achieving length 36 vs. 35. Similarly, GLA improves up to depth 4.

What hinders the performance of deeper models remains an open question. Consideration of several factors may offer some insight. Our theory constructs on real arithmetic whereas empirical computation is on finite precision; the impact of finite precision is beyond the scope of the current work. Moreover, how end-to-end gradient-based optimization shapes the solution is unclear, again illustrative of the practical learnability issue discussed earlier.

4.3 Rigid-body rotation problem

To demonstrate the relevance of our results beyond symbolic word problems, we conduct experiments with a continuous-valued state-tracking regression problem with explicit algebraic class.

The A_5 group acts as a rotational symmetry group of a dodecahedron. Two chosen symbols $\{\mathbf{P}, \mathbf{R}\}$, each corresponding to the $120°$ and $72°$ rotation, generate A_5. We chose a vector on a unit sphere as the initial state. The model observes a sequence of A_5 elements, and the task is to predict the corresponding transformed vector after each rotation. Algebraically, the problem corresponds to the action of A_5 embedded as a finite subgroup of $SO(3, \mathbb{R})$. Further details are provided in App. D.2.

Fig. 3a shows the sequence length at which each model's prediction error exceeds half of the chance level predictors.[8] Analogous to the symbolic tasks, depth plays a critical role in reducing the error. Critically, widening 1-layer models by a factor of 8 (hollow marks) barely changes the performance, supporting that the gain comes from depth rather than width. Deep transformer shows training variability, and the gains for other deep models saturate, again likely reflecting learnability issues.

Fig. 3b directly tests Thm. 3.2 by probing 1-layer models with commutator words built from the same symbols in different order. This allows us to vary $\|\Omega_2\|$ while keeping the generator mass fixed. As expected, the error monotonically scales with $\|\Omega_2\|$ except for DeltaProduct.

5 Discussion

Our theory connects the depth of sequence models to an abstract, algebraic extension theory. The connection provides an optimistic message: when the task admits such an algebraic decomposition, depth gives an exact representation by composing simpler layers. When the algebraic obstruction

[8] We found AUSSM to be generally unstable for this task; App. D.2 provides results with further discussions.

precludes an *exact* simulation, deep architectures compensate to rapidly minimize the error. This conclusion is supported by our experiments. We observe that increasing depth systematically reduces error on tasks outside each model's theoretical expressivity range and the error of 1-layer models grows with the commutator mass. Our observations suggest depth as a structural mechanism that mitigates complex order sensitivity. This also motivates parallelizable models with adaptive depth (Dehghani et al., 2019; Csordás et al., 2021; 2022; Csordás et al., 2024; Tan et al., 2023).

5.1 Future directions

Our results naturally suggest several future directions. One is to examine the impact of positional encoding (PE). Although additive PE is unlikely to unlock expressivity (Liu et al., 2023), an advanced form of multiplicative PE does boost expressivity (Yang et al., 2025). Understanding such mechanisms through algebraic structure is a promising and practical avenue of research.

Orthogonally, our theory assumes real arithmetic and may be sensitive to the use of finite precision in real-world applications. Exponentially vanishing error may fall below numerical resolution and blur the algebraic obstruction. On the other hand, finite precision may limit learnability under gradient-based optimization. How finite precision impacts expressivity needs to be deeply considered.

5.2 Related Work

Transformer variants The expressivity of sequence models is often studied via computational complexity and formal language theory (Giles et al., 1989; Gers & Schmidhuber, 2001; Schmidhuber et al., 2001; Weiss et al., 2018; Hahn, 2020; Merrill et al., 2020; Bhattamishra et al., 2020; Delétang et al., 2023; Irie et al., 2023; Strobl et al., 2024; Beck et al., 2024). Liu et al. (2023) showed that constant-depth transformers can simulate semiautomata with bounded length via the Krohn-Rhodes theorem, but cannot simulate for arbitrary length unless $\mathrm{TC}^0 = \mathrm{NC}^1$. Subsequent work extended to Markov models or directed acyclic graphs (Hu et al., 2024; Cheng et al., 2025).

Structured SSMs Merrill et al. (2024) proved that constant-depth commuting SSMs, such as Mamba, admit TC^0 simulations and share similar expressivity limits with Transformers. Following, recent work aims to improve SSM expressivity at tolerable computational expense. Siems et al. (2025) leverage input-dependent compositions of generalized Householder transformations, grounded on Schlag et al. (2021) and Yang et al. (2024b). Complementarily, another line of work leverages controlled differential equation (CDE) and rough path theory (Muca Cirone et al., 2024; Walker et al., 2024; 2025). Shakerinava et al. (2026) proved L-layer complex-valued diagonal SSMs can track solvable groups of $\leq L$ derived length, with empirical learnability gap.

6 Conclusion

Scalable sequence models achieve parallelism by imposing order symmetry, which limits their ability to *exactly* solve many sequence problems. We make this performance gap measurable by Lie-algebraic control theory, and connect the depth of sequence models to algebraic extension theory. Our theory shows that, even when exact simulation is not possible, the order-sensitive approximation error vanishes exponentially with depth. Experiments on symbolic word problems and 3D rotation state-tracking task supports our theory, while also exposing practical challenges such as deep model learnability. Together, we provide an algebraic perspective for understanding depth-dependent error-expressivity in parallelizable sequence models.

7 Limitations

Our work focuses on expressivity and error approximation, rather than learnability, optimization, or inductive biases which are other important challenges for sequence models. While our theory offers guidance for model selection when the target task has an explicit algebraic structure, the exact algebraic class of real-world problems is rarely known. Our main contributions are of a theoretical nature: we provide a theoretical framework to study how depth benefits certain sequence models. Although our theoretical results are relevant at any scale, large-scale experiments in language modeling or multimodal settings are out of scope.

AI USE STATEMENT

In this work, we used generative AI tools for ***implement methods.*** Specifically, part of our experimental code was written in assistance of generative AI tools. We have not used generative AI tools for ***help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses, generate synthetic data sets, clean and reformat dataset, design or provide feedback on research methodology or experiments, support qualitative and thematic data analysis, interpret results***, and ***assist with translation*** are not applicable to this work. Additionally, we used generative AI tools for ***edit a research paper to improve readability***. We have reviewed all AI-assisted work. LLM-generated code was verified and tested for correctness by 2 authors. LLM-edited text was verified and corrected by every authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

For theoretical results, key definitions and assumptions are stated in the main text properly. Full proofs are deferred to appendix (App. C), with proof sketch and intuitive examples provided in the main text.

We will release all our code on a public GitHub repository upon acceptance. The code contains instructions for reproducing the main experimental results. Our experiments primarily use existing models and benchmarks, and we detailed settings in the main text (Sec. 4) and appendix (App. D).

REFERENCES

Andrei A. Agrachev and Yuri L. Sachkov. *Control theory from the geometric viewpoint*. Encyclopaedia of mathematical sciences ; v. 87. Control theory and optimization ; 2. Springer, 2004.

Yehor Avdieiev and Alexander Tsymbaliuk. Affine standard lyndon words: A-type. *International Mathematics Research Notices*, 2024(21):13488–13524, 2024.

Maximilian Beck, Korbinian Pöppel, Markus Spanring, Andreas Auer, Oleksandra Prudnikova, Michael Kopp, Günter Klambauer, Johannes Brandstetter, and Sepp Hochreiter. xLSTM: Extended long short-term memory. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, Vancouver, Canada, December 2024.

Andrew Beckett and José Figueroa-O'Farrill. Symplectic actions and central extensions. *arXiv preprint arXiv:2203.07405*, 2022.

Satwik Bhattamishra, Kabir Ahuja, and Navin Goyal. On the ability and limitations of transformers to recognize formal languages. In *Proc. Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 7096–7116, Virtual only, November 2020.

Sergio Blanes, Fernando Casas, Jose-Angel Oteo, and José Ros. The magnus expansion and some of its applications. *Physics reports*, 470(5-6):151–238, 2009.

Salwa Bouallegue and Mongi Naimi. On primitive words. *Int. J. Algebra*, 4(13-16):693–707, 2010.

Dietrich Burde. Derived length and nildecomposable lie algebras. *arXiv preprint arXiv:1212.3113*, 2012.

Chang Chen, Yi-Fu Wu, Jaesik Yoon, and Sungjin Ahn. Transdreamer: Reinforcement learning with transformer world models. *arXiv preprint arXiv:2202.09481*, 2022.

Kuo-Tsai Chen, Ralph Fox, and Roger Lyndon. Free differential calculus. iv. the quotient groups of the lower central series. *Annals of Mathematics*, 68(1):81–95, 1958.

Yuan Cheng, Yu Huang, Zhe Xiong, Yingbin Liang, and Vincent YF Tan. Transformers provably learn directed acyclic graphs via kernel-guided mutual information. *arXiv preprint arXiv:2510.25542*, 2025.

Ilya Chevyrev and Andrey Kormilitzin. A primer on the signature method in machine learning. In *Signature Methods in Finance: An Introduction with Computational Applications*, pp. 3–64. Springer, 2025.

Ilya Chevyrev, Joscha Diehl, Kurusch Ebrahimi-Fard, and Nikolas Tapia. A multiplicative surface signature through its magnus expansion. *arXiv preprint arXiv:2406.16856*, 2024.

Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, et al. Rethinking attention with performers. In *Int. Conf. on Learning Representations (ICLR)*, Virtual only, 2021.

Róbert Csordás, Kazuki Irie, and Jürgen Schmidhuber. The devil is in the detail: Simple tricks improve systematic generalization of transformers. In *Proc. Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, Punta Cana, Dominican Republic, November 2021.

Róbert Csordás, Kazuki Irie, and Jürgen Schmidhuber. The neural data router: Adaptive control flow in transformers improves systematic generalization. In *Int. Conf. on Learning Representations (ICLR)*, Virtual only, April 2022.

Róbert Csordás, Kazuki Irie, Jürgen Schmidhuber, Christopher Potts, and Christopher D Manning. MoEUT: Mixture-of-experts universal transformers. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, Vancouver, Canada, December 2024.

E.A. De Kerf, G.G.A. Bäuerle, and A.P.E. Ten Kroode. Chapter 18 extensions of lie algebras. In *Lie Algebras*, volume 7 of *Studies in Mathematical Physics*, pp. 5–48. North-Holland, 1997. doi: https://doi.org/10.1016/S0925-8582(97)80002-4. URL `https://www.sciencedirect.com/science/article/pii/S0925858297800024`.

Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Lukasz Kaiser. Universal transformers. In *Int. Conf. on Learning Representations (ICLR)*, New Orleans, LA, USA, May 2019.

Grégoire Delétang, Anian Ruoss, Jordi Grau-Moya, Tim Genewein, Li Kevin Wenliang, Elliot Catt, Marcus Hutter, Shane Legg, and Pedro A Ortega. Neural networks and the Chomsky hierarchy. In *Int. Conf. on Learning Representations (ICLR)*, Kigali, Rwanda, May 2023.

Bo Deval, Xabier García-Martínez, and Tim Van der Linden. A universal kaluzhnin–krasner embedding theorem. *Proceedings of the American Mathematical Society*, 152(12):5039–5053, 2024.

Esraa Elelimy, Adam White, Michael Bowling, and Martha White. Real-time recurrent learning using trace units in reinforcement learning. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, Vancouver, Canada, December 2024.

Peter K Friz, Paul P Hager, and Nikolas Tapia. Unified signature cumulants and generalized magnus expansions. *Forum of Mathematics, Sigma*, 10, 2022.

Felix A Gers and Jürgen Schmidhuber. Lstm recurrent networks learn simple context-free and context-sensitive languages. *IEEE Transactions on Neural Networks*, 12(6):1333–1340, 2001.

C. Lee Giles, Guo-Zheng Sun, Hsing-Hen Chen, Yee-Chun Lee, and Dong Chen. Higher order recurrent networks and grammatical inference. In *Proc. Advances in Neural Information Processing Systems (NIPS)*, pp. 380–387, Denver, CO, USA, November 1989.

Riccardo Grazzi, Julien Siems, Jörg KH Franke, Arber Zela, Frank Hutter, and Massimiliano Pontil. Unlocking state-tracking in linear RNNs through negative eigenvalues. In *Int. Conf. on Learning Representations (ICLR)*, Vancouver, Canada, April 2025.

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

Michael Hahn. Theoretical limitations of self-attention in neural sequence models. *Transactions of the Association for Computational Linguistics*, 8:156–171, 2020.

Brian C Hall. *Lie groups, Lie algebras, and representations: An Elementary Introduction*. Springer, 2nd edition, 2015.

Robert Hermann and Arthur Krener. Nonlinear controllability and observability. *IEEE Transactions on automatic control*, 22(5):728–740, 1977.

Jiachen Hu, Qinghua Liu, and Chi Jin. On limitation of transformer for learning HMMs. *Preprint arXiv:2406.04089*, 2024.

Kazuki Irie, Róbert Csordás, and Jürgen Schmidhuber. Practical computational power of linear transformers and their recurrent and self-referential extensions. In *Proc. Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, Sentosa, Singapore, 2023.

Arieh Iserles, Hans Z Munthe-Kaas, Syvert P Nørsett, and Antonella Zanna. Lie-group methods. *Acta numerica*, 9:215–365, 2000.

Arieh Iserles, Walter D. van Suijlekom, Kurusch Ebrahimi-Fard, and Matilde Marcolli. Magnus expansions and beyond. In *Combinatorics and Physics*, volume 539 of *Contemporary Mathematics*, pp. 171–186. American Mathematical Society, 2011. ISBN 0821853295.

Velimir Jurdjevic and Héctor J Sussmann. Control systems on lie groups. *Journal of Differential equations*, 12(2):313–329, 1972.

Arjun Karuvally, Franz Nowak, Anderson T Keller, Carmen Amo Alonso, Terrence J Sejnowski, and Hava T Siegelmann. Bridging expressivity and scalability with adaptive unitary SSMs. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, San Diego, CA, USA, December 2025.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are RNNs: Fast autoregressive transformers with linear attention. In *Proc. Int. Conf. on Machine Learning (ICML)*, Virtual only, July 2020.

Najoung Kim and Sebastian Schuster. Entity tracking in language models. In *Proc. Association for Computational Linguistics (ACL)*, pp. 3835–3855, Toronto, Canada, July 2023.

Arthur J Krener. Bilinear and nonlinear realizations of input-output maps. *SIAM Journal on Control*, 13(4):827–834, 1975.

Arthur J Krener. A decomposition theory for differentiable systems. *SIAM Journal on Control and Optimization*, 15(5):813–829, 1977.

John M Lee. *Introduction to smooth manifolds*. Springer New York, 2nd edition, 2012.

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

Wilhelm Magnus, Abraham Karrass, and Donald Solitar. *Combinatorial group theory: Presentations of groups in terms of generators and relations*. Interscience Publishers, 1966.

Guy Melançon and Christophe Reutenauer. Lyndon words, free algebras and shuffles. *Canadian Journal of Mathematics*, 41(4):577–591, 1989.

William Merrill, Gail Weiss, Yoav Goldberg, Roy Schwartz, Noah A. Smith, and Eran Yahav. A formal hierarchy of RNN architectures. In *Proc. Association for Computational Linguistics (ACL)*, pp. 443–459, Virtual only, July 2020.

William Merrill, Jackson Petty, and Ashish Sabharwal. The illusion of state in state-space models. In *Proc. Int. Conf. on Machine Learning (ICML)*, Vienna, Austria, July 2024.

James Morrill, Cristopher Salvi, Patrick Kidger, and James Foster. Neural rough differential equations for long time series. In *Proc. Int. Conf. on Machine Learning (ICML)*, pp. 7829–7838. PMLR, 2021.

Nicola Muca Cirone, Antonio Orvieto, Benjamin Walker, Cristopher Salvi, and Terry Lyons. Theoretical foundations of deep selective state-space models. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, Vancouver, Canada, December 2024.

Antonio Orvieto, Samuel L Smith, Albert Gu, Anushan Fernando, Caglar Gulcehre, Razvan Pascanu, and Soham De. Resurrecting recurrent neural networks for long sequences. In *Proc. Int. Conf. on Machine Learning (ICML)*, pp. 26670–26698. PMLR, 2023.

Hao Peng, Nikolaos Pappas, Dani Yogatama, Roy Schwartz, Noah A Smith, and Lingpeng Kong. Random feature attention. In *Int. Conf. on Learning Representations (ICLR)*, Virtual only, 2021.

V Petrogradsky, Yu Razmyslov, and E Shishkin. Wreath products and kaluzhnin-krasner embedding for lie algebras. *Proceedings of the American Mathematical Society*, 135(3):625–636, 2007.

Joel W Robbin and Dietmar A Salamon. *Introduction to differential geometry*. Springer, 2022.

Joseph J Rotman and Joseph J Rotman. *An introduction to homological algebra*. Springer, 2nd edition, 2009.

Imanol Schlag, Kazuki Irie, and Jürgen Schmidhuber. Linear Transformers are secretly fast weight programmers. In *Proc. Int. Conf. on Machine Learning (ICML)*, Virtual only, July 2021.

Jürgen Schmidhuber. Learning to control fast-weight memories: An alternative to dynamic recurrent networks. *Neural Computation*, 4(1):131–139, 1992.

Jürgen Schmidhuber, Sepp Hochreiter, and Yoshua Bengio. Evaluating benchmark problems by random guessing. *A Field Guide to Dynamical Recurrent Networks*, 2001.

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.

Jerome Sieber, Carmen A Alonso, Alexandre Didier, Melanie N Zeilinger, and Antonio Orvieto. Understanding the differences in foundation models: Attention, state space models, and recurrent neural networks. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Julien Siems, Timur Carstensen, Arber Zela, Frank Hutter, Massimiliano Pontil, and Riccardo Grazzi. Deltaproduct: Improving state-tracking in linear RNNs via householder products. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, San Diego, CA, USA, December 2025.

Eduardo D Sontag. *Mathematical control theory: deterministic finite dimensional systems*. Springer New York, 2nd edition, 1998.

Lena Strobl, William Merrill, Gail Weiss, David Chiang, and Dana Angluin. What formal languages can transformers express? A survey. *Trans. Assoc. Comput. Linguistics (TACL)*, 12:543–561, 2024.

Hector J Sussmann. Existence and uniqueness of minimal realizations of nonlinear systems. *Mathematical Systems Theory*, 10(1):263–284, 1976.

Shawn Tan, Yikang Shen, Zhenfang Chen, Aaron C. Courville, and Chuang Gan. Sparse universal transformer. In *Proc. Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 169–179, Singapore, December 2023.

Roland Tóth. *Modeling and identification of linear parameter-varying systems*. Springer, 2010.

Loring W Tu. *An Introduction to Manifolds*. Springer, 1st edition, 2008.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proc. Advances in Neural Information Processing Systems (NIPS)*, pp. 5998–6008, Long Beach, CA, USA, December 2017.

Benjamin Walker, Andrew D McLeod, Tiexin Qin, Yichuan Cheng, Haoliang Li, and Terry Lyons. Log neural controlled differential equations: The lie brackets make a difference. *arXiv preprint arXiv:2402.18512*, 2024.

Benjamin Walker, Lingyi Yang, Nicola Muca Cirone, Cristopher Salvi, and Terry Lyons. Structured linear cdes: Maximally expressive and parallel-in-time sequence models. *arXiv preprint arXiv:2505.17761*, 2025.

Charles A Weibel. *An introduction to homological algebra*. 38. Cambridge university press, 1994.

Gail Weiss, Yoav Goldberg, and Eran Yahav. On the practical computational power of finite precision rnns for language recognition. In *Proc. Association for Computational Linguistics (ACL)*, pp. 740–745, Melbourne, Australia, July 2018.

Charles Wells. Some applications of the wreath product construction. *The American Mathematical Monthly*, 83(5):317–338, 1976.

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

Songlin Yang, Bailin Wang, Yikang Shen, Rameswar Panda, and Yoon Kim. Gated linear attention transformers with hardware-efficient training. In *Proc. Int. Conf. on Machine Learning (ICML)*, Vienna, Austria, July 2024a.

Songlin Yang, Bailin Wang, Yu Zhang, Yikang Shen, and Yoon Kim. Parallelizing linear transformers with the delta rule over sequence length. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, Vancouver, Canada, December 2024b.

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

A Lie algebra for proofs

In this section, we introduce further mathematical preliminaries as necessary to provide self-contained proofs of our theorems. We omit rigorous proofs for these observations, and interested readers are directed to related textbooks and research papers (Iserles et al., 2000; Lee, 2012; Petrogradsky et al., 2007; Tu, 2008; Rotman & Rotman, 2009; Hall, 2015; Robbin & Salamon, 2022; Liu et al., 2023). We open the section with the Lie group and Lie group action to motivate our Lie-algebraic approach.

A.1 Lie groups and Lie algebras

A set G is a semigroup when G is closed under an associative binary operation $\cdot : G \times G \to G$, usually called a "multiplication" operation. If $\exists e \in G$ such that $e \cdot g = g \cdot e = g$ for $\forall g \in G$, G is a monoid. Moreover, G is a group if for $\forall g \in G$ there exists an element $g^{-1} \in G$ such that $g^{-1} \cdot g = g \cdot g^{-1} = e$.

Lie group and Lie group action A Lie group G is a smooth manifold equipped with a group structure under smooth multiplication. For $x, y \in G$ let us denote the left translation $L_x(y) = xy$. From the smoothness of the multiplication, $L_x : G \to G$ is a diffeomorphism of the underlying manifold $\mathcal{G} = G$. In other words, translation can be thought of G *acting* on $\mathcal{G}$. The notion of Lie group *action* generally follows:

Definition A.1 (Lie group action)**.** Given a Lie group G and a smooth manifold $\mathcal{M}$, a (left) Lie group action is a smooth map $\Gamma : G \times \mathcal{M} \to \mathcal{M}$ that satisfies $\Gamma(e, h) = h$ and $\Gamma(xy, h) = \Gamma(x, \Gamma(y, h))$ for an identity $e \in G$, any two elements $x, y \in G$, and a point on manifold $h \in \mathcal{M}$.

Following, we define the *orbit* of h denoted as $\text{Orb}_G(h) := \{\Gamma(g,h) \mid g \in G\}$. *Stabilizer*, or an isotropy subgroup, is a subgroup of G where $\text{Stab}_G(h) := \{g \in G \mid \Gamma(g,h) = h\}$.

While Def. A.1 straightforwardly describes how Γ operates on $\mathcal{M}$, it is often advantageous to consider Γ as inducing a diffeomorphism on $\mathcal{M}$. We can naturally introduce a Lie group homomorphism $\Pi : G \to \text{Diff}(\mathcal{M})$ such that $\Pi(g) = \Gamma(g, \cdot)$, where $\text{Diff}(\mathcal{M})$ denotes collection of diffeomorphisms of $\mathcal{M}$.

Lie bracket and Lie algebra Now one may consider a smooth vector field $V : G \to TG$ and their collection $\mathfrak{X}(G)$. Given the differential of the left translation at point y,

$$(dL_x)_y : T_yG \to T_{L_x(y)}G = T_{xy}G,$$

the pushforward $(L_x)_*V \in \mathfrak{X}(G)$ is well-defined (Tu, 2008; Lee, 2012). Then, V is said to be left-invariant when $(L_x)_*V = V$. In other words, a left-invariant vector field V satisfies $\big((L_x)_*V\big)_{xy} = (dL_x)_yV_y = V_{xy}$ for $\forall x, y \in G$ and completely determined by its evaluation at identity, as $V_x = V_{xe} = (dL_x)_eV_e$. Thus, a collection of left-invariant vector fields, $\mathfrak{X}_L(G)$, has a one-to-one correspondence with T_eG.

As a tangent vector is defined by how it differentiates smooth functions, a smooth vector field on $\mathcal{M}$ is commonly identified as the derivation $C^\infty(\mathcal{M}) \to C^\infty(\mathcal{M})$. Following Leibniz rule, the Lie bracket naturally arises to satisfy the derivation property (Tu, 2008). For $f, g \in C^\infty(\mathcal{M})$,

$$\begin{aligned} XY(fg) &= X((Yf)g + fYg) \\ &= (XYf)g + (Yf)(Xg) + (Xf)(Yg) + f(XYg) \\ &\neq (XYf)g + f(XYg). \end{aligned}$$

Observe the extra term $(Yf)(Xg) + (Xf)(Yg)$ obstructs the derivation property. One can subtract $YX(fg)$ to cancel out the extra term, and naturally $(XY - YX)$ is again a derivation of $C^\infty(\mathcal{M})$.

Definition A.2 (Lie bracket of vector fields)**.** Given two smooth vector fields $X, Y : M \to TM$ on an open subset M of a smooth manifold $\mathcal{M}$, their Lie bracket is defined as following:

$$[X,Y]f = (XY - YX)f = X(Yf) - Y(Xf), \quad \forall f \in C^\infty(M). \tag{6}$$

$\mathfrak{X}_L(G)$ is closed under the Lie bracket. Now we are ready to associate the Lie algebra $\mathfrak{g}$ to the Lie group G:

Definition A.3 (Lie algebra associated to Lie group)**.** For a Lie group G, we identify the vector space of Lie algebra $\mathfrak{g} := T_eG$. For any $X \in \mathfrak{g}$, assign a unique $X^L \in \mathfrak{X}_L(G)$ that satisfies $X^L(y) = (dL_y)_eX$ for $y \in G$. We define the Lie bracket operation on $\mathfrak{g}$ such that for $X, Y \in \mathfrak{g}$,

$$[X,Y] := [X^L, Y^L](e).$$

Then a map $\pi^L : \mathfrak{g} \to \mathfrak{X}_L(G)$, $\pi^L(X) = X^L$ is a Lie algebra isomorphism.

Naturally, Lie algebra $\mathfrak{g}$ can also be identified as a collection of the left-invariant, first-order differential operators on G. Note the Lie bracket operation depends on the identification: with $\mathfrak{g} \cong T_eG$, $[X,Y] \in T_eG$. On the other hand, identifying $\mathfrak{g} \cong \mathfrak{X}_L(G)$ produces $[X,Y] \in \mathfrak{X}_LG$. The identification should be clear from the context.

We can analogously define the right translation on Lie group $R_x(y) = yx$. The differential of the right translation at point y gives $(dR_x)_y : T_yG \to T_{R_x(y)}G = T_{yx}G$, and we denote the collection of right-invariant vector fields as $\mathfrak{X}_R(G)$. Notationally, one can prove that $[X^L, Y^L](e) = -[X^R, Y^R](e)$ (Lee, 2012). In that, $\pi^R : \mathfrak{g} \to \mathfrak{X}_RG$, $\pi^R(X) = X^R$ swaps the Lie bracket sign.

Bridge to geometry As a concrete example, consider the open set M in Eq. 6 with local coordinate charts $x = (x^1, \cdots, x^n)$ around point $p \in M$. Then we can rewrite $X \in \mathfrak{X}(M)$ in a coordinate expression:

$$X = \sum_{a=1}^{n} X^a \partial_a, \quad X^a = X(x^a) \in C^\infty(M),$$

and similarly for $Y \in \mathfrak{X}(M)$. Given the local coordinate, Lie brackets get a tangible interpretation. One can exhaust the vector field Lie brackets with coordinate,

$$[X, Y] = \sum_{c=1}^{n} [X, Y]^c \partial_c, \quad [X, Y]^c = X(Y^c) - Y(X^c),$$

where $[X, Y]^c$ is again a coordinate function of $[X, Y] \in \mathfrak{X}(M)$. Evaluated at p, $[X, Y]^c(p) = X_p(Y^c) - Y_p(X^c)$. Then from the differential it is transparent:

$$[X, Y]^c(p) = X_p(Y^c) - Y_p(X^c) = (dY^c)_p(X_p) - (dX^c)_p(Y_p). \tag{7}$$

One may interpret above as a directional derivative of a smooth coordinate function along the tangent vector direction (Lee, 2012).

A.2 Representation and infinitesimal action

Lie group representation The abstract notion provided at App. A.1 is often easier to work with a tangible object, and matrix is one of such. As $g, g^{-1} \in G$, let us consider the collection of any invertible matrices in $\mathbb{R}^{n \times n}$. Over the real field, we name the collection as the general linear group and denotes as $\mathrm{GL}(n; \mathbb{R})$. Matrix Lie group is defined as a closed subgroup of $\mathrm{GL}(n; \mathbb{R})$, and many Lie groups in application arise as one.

These are the case of *Lie group representations* (Lee, 2012; Hall, 2015). Given a real vector space V, a representation of Lie group G is a Lie group homomorphism $\Pi : G \to \mathrm{GL}(V)$. When $\dim V = n$, we can identify $\mathrm{GL}(V)$ with $\mathrm{GL}(n; \mathbb{R})$ via basis choice. Recall Def. A.1 that representation is just a linear Lie group action of G on V.

Lie algebra representation With Def. A.3 we can think of the corresponding *Lie algebra representations*. For a Lie group G associated with $\mathfrak{g}$, consider $X \in \mathfrak{g} \cong T_e G$ and corresponding $X^L \in \mathfrak{X}_L(G)$. Let $c_X : \mathbb{R} \to G$ be the integral curve of X^L with $c_X(0) = e \in G$, starting at the identity. Then we can construct the exponential map, which is a canonical smooth map from the Lie algebra to its integrating group (Lee, 2012):

$$\exp : \mathfrak{g} \to G, \quad \exp(X) := c_X(1),$$

and we write $\exp(tX) = c_X(t)$ so that $\frac{d}{dt}|_{t=0} \exp(tX) = X$. Notice that X corresponds to the initial velocity of $c_X(t)$.

One concrete example is the matrix Lie algebra $\mathfrak{gl}(n, \mathbb{R})$, which is a collection of any matrices in $\mathbb{R}^{n \times n}$. With the Lie group representation:

$$\Pi : G \to \mathrm{GL}(n; \mathbb{R}), \quad \Pi_* : T_e G \to T_{\Pi(e)} \mathrm{GL}(n; \mathbb{R}) = T_{I_n} \mathrm{GL}(n; \mathbb{R}),$$

where we identified $\mathrm{GL}(n; \mathbb{R})$-associated Lie algebra $\mathfrak{gl}(n, \mathbb{R}) \cong T_{I_n} \mathrm{GL}(n; \mathbb{R})$. Then the Lie algebra representation is naturally induced:

$$\pi := (d\Pi)_e : \mathfrak{g} \to \mathfrak{gl}(n, \mathbb{R}),$$

in that, π is the differential of Π at the identity. For the matrix Lie algebra, exponential map is the matrix exponential expm:

$$\Pi(\exp(X)) = \mathrm{expm}(\pi(X)),\ \pi(X) = \left.\frac{d}{dt}\right|_{t=0} \Pi(\exp(tX)). \tag{8}$$

We extend the differential of representation to the differential of action, or *infinitesimal* action, in the following paragraph.

Fundamental vector field Consider the Lie group G acts on itself by left multiplication. For $h \in G$ and $X \in \mathfrak{g}$, let us define a vector field $X^\sharp$:

$$X^\sharp(h) = \left.\frac{d}{dt}\right|_{t=0} \exp(-tX)h = -(dR_h)_e X \in T_h G.$$

Observe that $X^\sharp(h)$ recovers familiar ODE $\dot{h} = Xh$ up to sign change, when $G \subseteq \mathrm{GL}(n, \mathbb{R})$.

We can extend above to an arbitrary smooth finite manifold $\mathcal{M}$, replacing canonical left action on G to a chosen left Lie group action $\Gamma : G \times \mathcal{M} \to \mathcal{M}$. For each $X \in \mathfrak{g}$ and $h \in \mathcal{M}$, we define the fundamental vector field $X^\sharp \in \mathfrak{X}(\mathcal{M})$:

$$X^\sharp(h) := \frac{d}{dt}\bigg|_{t=0} \Gamma(\exp(-tX), h) \in T_h\mathcal{M}, \tag{9}$$

then the map $\gamma : \mathfrak{g} \to \mathfrak{X}(\mathcal{M})$, $\gamma(X) = X^\sharp$ is a Lie algebra homomorphism (Iserles et al., 2000; Robbin & Salamon, 2022). Therefore γ is an *infinitesimal* Lie group action, and considered as Lie algebra action on $\mathcal{M}$.

Remark A.4. Many text writes $X^\sharp$ in a *right-invariant* vector field form (Robbin & Salamon, 2022):

$$X^\sharp(h) := \frac{d}{dt}\bigg|_{t=0} \Gamma(\exp(tX), h) \in T_h\mathcal{M}, \tag{10}$$

and the map $\gamma(X) = X^\sharp$ is a Lie algebra *anti*homomorphism (Iserles et al., 2000; Robbin & Salamon, 2022). Under this definition, one would observe that $[\pi(X), \pi(Y)] = \pi([X, Y])$, a matrix commutator, aligns with $[X^\sharp, Y^\sharp] = -[X, Y]^\sharp$ up to sign change. In practice, the sign can be absorbed and need not be worried. Some literature swap the Lie bracket direction to handle the sign change, but this is just a choice of convention and does not obstruct any fundamental logic.

Bridge to homogeneous ODE As observed above, when a Lie group $G \subseteq \mathrm{GL}(n, \mathbb{R})$ left-acts on a smooth manifold $\mathcal{M} \cong \mathbb{R}^n$, the fundamental vector field recovers homogeneous ODE. For a point $h \in \mathcal{M}$ and $\mathbf{A} \in \mathfrak{g} \subseteq \mathfrak{gl}(n, \mathbb{R})$, we select Γ to be a left multiplication. Then obviously:

$$\dot{h} = \frac{d}{ds}\bigg|_{s=0} \Gamma(\exp(s\mathbf{A}), h) = \mathbf{A}h,$$

which corresponds to the linear controlled homogeneous ODE, $\dot{h}(t) = \mathbf{A}(x(t))h(t)$, given a fixed input $x(t)$.

Critically, when G acts on itself, we can recover the state-transition matrix Φ. Recall Def. 2.2, Φ satisfies:

$$\dot{\Phi}(t, 0) = \mathbf{A}(x(t))\Phi(t, 0), \quad \Phi(0, 0) = I_n, \tag{11}$$

then Eq. 11 forms a homogeneous ODE system with Φ being its latent state. This observation is very useful: it is immediate that for any admissible input, $\Phi(t, 0) \in G$. In a sense, one may understand Eq. 11 as tracking *every* possible state evolution, rather than a single state. This is a conventional perspective in geometric control theory, commonly referred as control theory on Lie groups (Jurdjevic & Sussmann, 1972; Iserles et al., 2011).

A.3 Lie algebra extension

Analogous to the notion of the group extension, Lie algebra extension formalizes *gluing* two Lie algebras together. Here we introduce the notion of Lie algebra extension (Weibel, 1994; Rotman & Rotman, 2009; De Kerf et al., 1997; Beckett & Figueroa-O'Farrill, 2022).

Definition A.5 (Lie algebra extension)**.** The Lie algebra $\tilde{\mathfrak{h}}$ is an extension of base Lie algebra $\mathfrak{h}$ by (finite-dimensional) kernel Lie algebra $\mathfrak{a}$ if there exists a short exact sequence of Lie algebras:

$$0 \to \mathfrak{a} \xrightarrow{i} \tilde{\mathfrak{h}} \xrightarrow{\theta} \mathfrak{h} \to 0. \tag{12}$$

Redundantly, $i : \mathfrak{a} \to \tilde{\mathfrak{h}}$ is a Lie algebra monomorphism, while $\theta : \tilde{\mathfrak{h}} \to \mathfrak{h}$ is a Lie algebra epimorphism. The short exact sequence satisfies $\mathrm{im}\, i = \ker \theta$, thus $i(\mathfrak{a})$ is an ideal in $\tilde{\mathfrak{h}}$ and $\tilde{\mathfrak{h}}/i(\mathfrak{a}) \cong \mathfrak{h}$. Following common convention, we identify $\mathfrak{a}$ with $i(\mathfrak{a})$ without confusion.

Remark A.6 (Tower of abelian extension). A solvable Lie algebra with derived series $\mathfrak{g} = \mathfrak{g}^{(0)} \supset \mathfrak{g}^{(1)} \supset \cdots \supset \mathfrak{g}^{(k)} = \{0\}$ is an extension:

$$0 \to \mathfrak{g}^{(n)} \to \mathfrak{g} \to \mathfrak{g}/\mathfrak{g}^{(n)} \to 0.$$

When $n = k - 1$, $\mathfrak{g}^{(k-1)}$ is an abelian Lie (sub-)algebra and $\mathfrak{g}/\mathfrak{g}^{(k-1)}$ is a solvable Lie algebra of derived length $k - 1$. In iteration, we recover a k-length *tower* of abelian Lie algebra extension.

Split extension Understanding how $\mathfrak{h}$ acts on $\mathfrak{a}$ is critical in characterizing $\tilde{\mathfrak{h}}$. We begin with a simple case.

Definition A.7 (Split extension)**.** In Eq. 12 consider a linear section $s : \mathfrak{h} \to \tilde{\mathfrak{h}}$ that satisfies $\theta \circ s = 1_{\mathfrak{h}}$, an identity map. If s is a Lie algebra homomorphism, Eq. 12 is *split*.

If Lie algebra extension splits, then we say $\tilde{\mathfrak{h}}$ is a *semidirect product* (or sum) of $\mathfrak{h}$ by $\mathfrak{a}$ and we denote as $\tilde{\mathfrak{h}} = \mathfrak{h} \ltimes \mathfrak{a}$ (or $\mathfrak{a} \rtimes \mathfrak{h}$) (Rotman & Rotman, 2009).[9]

Consider $X, Y \in \mathfrak{h}$ and $u, v \in \mathfrak{a}$. $\tilde{\mathfrak{h}}$ acts on $\mathfrak{a}$ by derivation, as $\mathfrak{a}$ is an ideal in $\tilde{\mathfrak{h}}$. As the linear section s is a Lie algebra homomorphism, $s(X) \in \tilde{\mathfrak{h}}$ naturally acts on $\mathfrak{a}$ by derivation too. Thus, the induced action of $\mathfrak{h}$ on $\mathfrak{a}$ is straightforward here:

$$\rho : \mathfrak{h} \to \mathrm{Der}(\mathfrak{a}), \quad \rho(X)v = [s(X), v],$$

and Lie bracket on split extension follows:

$$[(X, u), (Y, v)]_{\tilde{\mathfrak{h}}} = ([X, Y]_{\mathfrak{h}}, [u, v]_{\mathfrak{a}} + \rho(X)v - \rho(Y)u). \tag{13}$$

One special case of split extension is the trivial extension where $[(X, u), (Y, v)] = ([X, Y], [u, v])$, and we name it as a *direct product* (or sum) of $\mathfrak{h}$ and $\mathfrak{a}$.

Bridge to inhomogeneous ODE Similar to App. A.2, affine vector field naturally arises from the semidirect product. Consider an inhomogeneous ODE $\dot{h}(t) = \mathbf{A}(x(t))h(t) + \mathbf{b}(x(t))$. As noted often (Krener, 1975; 1977; Iserles et al., 2000), one can augment the state to *homogenize* the ODE:

$$\bar{h}(t) = \begin{pmatrix} h(t) \\ 1 \end{pmatrix}, \quad \dot{\bar{h}}(t) = \begin{pmatrix} \mathbf{A}(x(t)) & \mathbf{b}(x(t)) \\ 0 & 0 \end{pmatrix} \bar{h}(t).$$

Then it is immediate that the generator class is affine. Consider a finite-dimensional Lie group of affine transformation, $\mathrm{Aff}(n, \mathbb{R})$. Its associated Lie algebra $\mathfrak{aff}(n, \mathbb{R})$ is the canonical example of the split extension:

$$0 \to \mathbb{R}^n \to \mathfrak{aff}(n, \mathbb{R}) \to \mathfrak{gl}(n, \mathbb{R}) \to 0,$$

such that $(A, b) \in \mathfrak{gl}(n, \mathbb{R}) \ltimes \mathbb{R}^n$. With a standard matrix embedding,

$$(A, b) \longmapsto \begin{pmatrix} A & b \\ 0 & 0 \end{pmatrix},$$

which corresponds to the generator class of the homogenized ODE above.

Let us confirm that the homogenized ODE respects the split extension bracket rule in Eq. 13. From the matrix representation above, one can simply check their matrix commutator. With two fixed inputs x, y:

$$\bar{\mathbf{A}}(x) = \begin{pmatrix} \mathbf{A}(x) & \mathbf{b}(x) \\ 0 & 0 \end{pmatrix}, \quad [\bar{\mathbf{A}}(x), \bar{\mathbf{A}}(y)] = \begin{pmatrix} [\mathbf{A}(x), \mathbf{A}(y)] & \mathbf{A}(x)\mathbf{b}(y) - \mathbf{A}(y)\mathbf{b}(x) \\ 0 & 0 \end{pmatrix},$$

especially when $\mathrm{Lie}\{\mathbf{A}(x)\}$ is abelian, $[\mathbf{A}(x), \mathbf{A}(y)] = 0$ and only the top right entry is nonzero.

Now consider vector field representation. For the brevity, let us keep $\mathrm{Lie}\{\mathbf{A}(x)\}$ be abelian. Recall Eq. 7:

$$\begin{aligned} X(h) = \mathbf{A}(x)h + \mathbf{b}(x), \quad Y(h) = \mathbf{A}(y)h + \mathbf{b}(y) \\ [X, Y]^c = \sum_{a=1}^{n} (\mathbf{b}^a(x)\mathbf{A}^{ca}(y) - \mathbf{b}^a(y)\mathbf{A}^{ca}(x)) \\ [X, Y] = \mathbf{A}(y)\mathbf{b}(x) - \mathbf{A}(x)\mathbf{b}(y), \end{aligned} \tag{14}$$

with sign change noted in Rem. A.4.

Taken together with App. A.2, switching the vector state space with the corresponding Lie group, or flow space, is often referred as *lifting* (Krener, 1977). Except for minor details in implementation, lifting does not lose any generality and we can *project* back to the state space (Iserles et al., 2000; 2011).

[9] Note that the convention of sign direction is not accordant across references (Petrogradsky et al., 2007; Rotman & Rotman, 2009). here we emphasize that $\mathfrak{h}$ acts on $\mathfrak{a}$.

Central extension and beyond If all extension belongs to the split extension class, our discussion would be much easier. However note that the linear section $s : \mathfrak{h} \to \tilde{\mathfrak{h}}$ needs not be a Lie algebra homomorphism. Among non-split extension, central extension is another key class of the Lie algebra extension.

Definition A.8 (Central extension)**.** In Eq. 12 let $\mathfrak{a}$ be abelian. When $i(\mathfrak{a})$ is in the center of $\tilde{\mathfrak{h}}$ such that $[\tilde{X}, u] = 0$ for any $\tilde{X} \in \tilde{\mathfrak{h}}$ and $u \in i(\mathfrak{a})$, we call it as central extension.

As s is not necessarily Lie algebra homomorphism, we introduce a correction term, called *2-cocycle* (Weibel, 1994):

$$\omega : \mathfrak{h} \times \mathfrak{h} \to \mathfrak{a}, \quad \omega(X, Y) := [s(X), s(Y)] - s([X, Y]_{\mathfrak{h}}). \tag{15}$$

Recall Eq. 13. As $\mathfrak{a}$ belongs to the center, the induced action $\rho : \mathfrak{h} \to \mathrm{Der}(\mathfrak{a})$ is trivial. Thus, in central extension, the bracket rule simplifies to:

$$[(X, u), (Y, v)] = ([X, Y], \omega(X, Y)).$$

When s is Lie algebra homomorphism, central extension breaks down to trivial extension.

Taken together, the Lie bracket on general Lie algebra extension follows:

$$[(X, u), (Y, v)]_{\tilde{\mathfrak{h}}} = [[X, Y]_{\mathfrak{h}}, [u, v]_{\mathfrak{a}} + \rho(X)v - \rho(Y)u + \omega(X, Y)].$$

Characterization of general Lie algebra extension is nontrivial. At least under Lie group setting, the Kaloujnine-Krasner universal embedding theorem states that any Lie group extensions embeds into the *wreath* product of Lie group (Wells, 1976; Deval et al., 2024). Fortunately, following studies have extended the universal embedding theorem to Lie algebra (Petrogradsky et al., 2007; Deval et al., 2024).

B Control theory basics

B.1 Abstract control systems

Here we discuss abstract notions of controlled dynamical systems (Iserles et al., 2000). On a real smooth finite-dimensional manifold $\mathcal{M}$, we consider a time-dependent tangent vector field:

$$F : [0, \tau] \times \mathcal{M} \times \mathcal{X} \to T\mathcal{M},$$

which is locally Lipschitz in $h \in \mathcal{M}$ and measurable in time $t \in [0, \tau]$. In this context, we call $\mathcal{M}$ the *state space* and $\mathcal{X}$ the *input space*. Then, with a fixed input path $x : [0, T] \to \mathcal{X}$ we obtain the controlled ODE:

$$\dot{h}(t) = F_x(t, h(t)), \; F_x(t, h(t)) := F(t, h(t), x(t)), \tag{16}$$

assuming the path is locally bounded and integrable (Iserles et al., 2000). We call a tuple $S = (\mathcal{M}, F, \mathcal{X})$ a *controlled dynamical system*. Lastly, we say the system is initialized when $h(0)$ is fixed. Restricting $\mathcal{M}$ to be on a Euclidean state space realizes SSM. As noted in the main text, we assume SSMs are minimal (App. B.3).

B.2 Simulation

Here we formally introduce the notion of simulation (Sussmann, 1976; Hermann & Krener, 1977; Krener, 1977).

Definition B.1 (Simulation)**.** Let two initialized controlled dynamical systems S_0, S_1 share an input space. Consider a smooth map $\varphi : \mathcal{M}_0 \to \mathcal{M}_1$ such that $\varphi(h_0(t)) = h_1(t)$ for any admissible inputs. If φ is a surjective submersion, we say S_0 *simulates* S_1. Moreover, if φ is (locally) diffeomorphic, we say S_0 is equivalent to S_1.

Note that if S_0 simulates S_1 and $\dim S_0 = \dim S_1$, then φ is diffeomorphic. Therefore, equivalence implies simulation.

When restricted to simulation between 1-layer SSMs, arbitrary smooth φ would typically leave the SSM model class (Krener, 1977; Tóth, 2010). Therefore, the notion of simulation in SSMs restrict φ to be full-rank linear map, fixed once for all admissible inputs. Following, diffeomorphic φ is restricted to be invertible linear map. We define approximation error correspondingly.

Definition B.2 (SSM approximation error on horizon)**.** Consider initialized $\mathbf{S}_0$, $\mathbf{S}_1$ with $n_0 = \dim \mathbf{S}_0 \geq \dim \mathbf{S}_1 = n_1$. Let $\mathcal{X}_{ad}$ be shared admissible input path space, with

$$U := \{x \in \mathcal{X}_{ad} \mid x : [0, T] \to \mathcal{X}, \|x\| \leq M\}$$

be measure bounded paths over chosen interval $[0, T]$. Then, we define approximation error:

$$\Delta_h := \inf_{\mathbf{P} \in \mathbb{R}^{n_1 \times n_0}} \sup_{x \in U} \big\| \mathbf{P} h_0(T) - h_1(T) \big\|, \tag{17}$$

where $\mathbf{P}$ is a full-rank linear map.

Note that $\mathbf{P}$ can be replaced with invertible linear map $\mathbf{T}$ by state augmentation on $\mathbf{S}_1$, which leads to the standard state-space equivalence relation in linear parameter-varying (LPV) system theory (Tóth, 2010).

B.3 Minimality

Here we state minimality from the Lie group action perspective (Jurdjevic & Sussmann, 1972; Krener, 1977).

Consider 5-tuple $(\mathcal{M}, F, \mathcal{X}_{ad}, \mathcal{Y}, g)$ and its evolution operator Ψ. Consider a collection $S = \{\Psi_x\}$ for $x \in \mathcal{X}_{ad}$, then one can notice $S \subset \mathrm{Diff}(\mathcal{M})$ is a semigroup. Although any $\Psi_x(t, 0)$ represent as an invertible matrix, the existence of admissible input for $\Psi_{x'} = \Psi_x^{-1}$ is not guaranteed. With completion, we denote a group G be the smallest subgroup of $\mathrm{Diff}(\mathcal{M})$ containing S.

For a chosen point $h \in \mathcal{M}$, consider a collection $S(h) = \{\Psi(h) : \Psi \in S\}$ and $G(h) = \{\Psi(h) : \Psi \in G\}$. It is akin to the orbit of Lie group action: intuitively, $S(h)$ is the set of accessible points on $\mathcal{M}$ under admissible inputs. We say the system is (weakly) *controllable* at point h when $S(h) = \mathcal{M}$. When satisfied, the condition implies $G(h) = \mathcal{M}$.

Now consider the output map. Given any two points $h_1, h_2 \in \mathcal{M}$, denote $y_1(t) = g(\Psi(t, 0)(h_1))$ and $y_2(t)$ correspondingly. Then h_1 and h_2 are said to be indistinguishable when $y_1(t) = y_2(t)$ for any admissible inputs. The system is *observable* when h_1 and h_2 being indistinguishable implies $h_1 = h_2$. Collectively, we say the system $(\mathcal{M}, F, \mathcal{X}_{ad}, \mathcal{Y}, g)$ is *minimal* when both (weakly) controllable and observable (Krener, 1977).

Remark B.3. Often, attention is restricted to open subset $U \subseteq \mathcal{M}$ however small. Without global Lie group action and orbits, one can characterize the system with Lie algebras. For example, weak controllability is replaced by *Lie algebra accessibility rank condition* (Sontag, 1998).

One may easily notice the notion of minimality in LTI system is a special case of the above. We can similarly induce notion of minimality for SSMs, commonly studied as a (quasi-)LPV system (Tóth, 2010).

B.4 Magnus series and path signature

The Magnus series provides Lie-algebraic expansion of the state-transition matrix. As various evolution operators in physical applications are indeed a integral curve on a Lie group, the Magnus expansion (or its truncation) serves as a useful tool in numerical approximation problems (Iserles et al., 2000). Given a system $\mathbf{S}$, we explicitly write down a few first terms:

$$\begin{aligned}
&\Phi(t, 0) = \exp \Omega(t), \quad \Omega(t) = \sum \Omega_i(t) \\
&\Omega_1(t) = \int_0^t \mathbf{A}(t_1)\, dt_1 \\
&\Omega_2(t) = \frac{1}{2} \int_0^t dt_1 \int_0^{t_1} dt_2 [\mathbf{A}(t_1), \mathbf{A}(t_2)] \\
&\Omega_3(t) = \frac{1}{6} \int_0^t dt_1 \int_0^{t_1} dt_2 \int_0^{t_2} dt_3 \Big([\mathbf{A}(t_1), [\mathbf{A}(t_2), \mathbf{A}(t_3)]] + [\mathbf{A}(t_3), [\mathbf{A}(t_2), \mathbf{A}(t_1)]] \Big) \\
&\quad \dots
\end{aligned}$$

Observe how the lower central series contributes to each order of the Magnus series. Specifically, the i-th order Magnus term involves $(i-1)$ Lie bracket operations. In the lower central series, these elements lie in $\mathfrak{g}^{i-1}$. Consequently, if $\mathfrak{g}$ is a class c nilpotent Lie algebra, then the Magnus expansion of $\mathbf{S}_\mathfrak{g}$ truncates at order c and all Magnus terms $\Omega_{>c}$ vanish. As a special case, abelian $\mathfrak{g}$ is the class 1 nilpotent Lie algebra, all commutators of $\mathbf{S}_\mathfrak{g}$ vanish and we have truncation at order 1.

Recent progress in continuous-time neural network modeling leverages rough path theory and path signature (Muca Cirone et al., 2024; Walker et al., 2024; 2025). Path signature is deeply related to the Magnus series (Chevyrev et al., 2024; Chevyrev & Kormilitzin, 2025). For linear CDE, the path signature can be expressed as a tensor exponential of the Magnus expansion (Friz et al., 2022). In other words, the Magnus expansion is a representation of *log-signature* (Morrill et al., 2021; Walker et al., 2024) when input variation is bounded. Log-signature underscores the algebraic property of input path, while the Magnus expansion prioritizes state transition. The study of the Magnus series is relatively mature and often provides both theoretical and computational benefits when the path signature admits the Magnus expansion (Friz et al., 2022).

C Proofs

C.1 Proof of Lemma 3.1

Proof. Algebraically, it is straightforward. Any image of abelian Lie algebra under Lie algebra homomorphism is abelian. There is no representation of abelian Lie algebra that depends on path order.

Constructively, general SSM $\mathbf{S}$ and abelian SSM $\hat{\mathbf{S}}$ are linear ODEs. It suffices to find the existence of nonsingular linear state transformation map $\mathbf{T}$ that satisfies $\mathbf{T}\hat{h}(t) = h(t)$ for any time t and admissible inputs (Sec. 2.3 and App. B.2).

Let us denote $(\Phi, \hat{\Phi})$ be the state-transition matrix of $(\mathbf{S}, \hat{\mathbf{S}})$. It is clear that $\mathbf{T}\hat{h}(t) = h(t)$ implies $\mathbf{T}\hat{\Phi}(t,0) = \Phi(t,0)\mathbf{T}$. Consider an admissible input path $x : [0,T] \to \mathcal{X}$, and its measure-preserving permutation path $x'(t) = x(\sigma(t))$ where $\sigma : [0,T] \to [0,T]$. Assume x' is also admissible. Then, being abelian:

$$\hat{\Phi}_x(T,0) = \hat{\Phi}_{x'}(T,0), \quad \mathbf{T}(\hat{\Phi}_x - \hat{\Phi}_{x'})\mathbf{T}^{-1} = \mathbf{0}.$$

However, for non-abelian $\mathbf{S}$, chronological exponential map depends on the path order and $\Phi_x(T,0) \neq \Phi_{x'}(T,0)$, contradicting the existence of $\mathbf{T}$.

As noted in App. B.2, the proof generalizes to the case $m = \dim \hat{\mathbf{S}} \geq \dim \mathbf{S} = n$. $\mathbf{T}$ simply needs to satisfy:

$$\mathbf{T}\hat{h} = \begin{bmatrix} h \\ * \end{bmatrix} \begin{matrix} \} \, n \\ \} \, m-n, \end{matrix}$$

for some variables $*$ to match dimension (Tóth, 2010). The algebraic nature is intact, and the proof immediately follows. With respect to Eq. 17, truncating $*$-corresponding rows of $\mathbf{T}$ recover full-rank matrix $\mathbf{P}$. □

C.2 Proof of Theorem 3.2

Here we divide the problem into abelian and restricted cases. Notation follows App. C.1.

Abelian SSM. Consider general SSM $\mathbf{S}$. Assume an input path x and permuted x' on window $[0,T]$ are both admissible. With small enough T, we can derive a loose lower bound of $\|\Phi_x(T) - \Phi_{x'}(T)\|$ with Magnus expansion in App. B.4:

$$\begin{aligned} \|\Phi_x(T) - \Phi_{x'}(T)\| &= \|\exp\Omega(T) - \exp\Omega'(T)\| \\ &\geq \exp\Big(-\max\{\|\Omega(T)\|, \|\Omega'(T)\|\}\Big)\|\Omega(T) - \Omega'(T)\| \\ &= \lambda\|\Omega(T) - \Omega'(T)\|, \qquad \lambda \in (0,1]. \end{aligned}$$

Recall Eq. 5,

$$\begin{aligned}\|\Omega_2(T) - \Omega_2'(T)\| &= \left\| \frac{1}{4} \iint_{[0,T]^2} \Big(\operatorname{sgn}(t_1 - t_2) - \operatorname{sgn}(\sigma(t_1) - \sigma(t_2))\Big) [\mathbf{A}(x(t_1)), \mathbf{A}(x(t_2))]\, dt_1 dt_2 \right\| \\ &\le \left\| \frac{1}{2} \iint_{[0,T]^2} \operatorname{sgn}(t_1 - t_2) [\mathbf{A}(x(t_1)), \mathbf{A}(x(t_2))]\, dt_1 dt_2 \right\| \\ &= 2\|\Omega_2(T)\|,\end{aligned}$$

where equality holds when $\sigma(t)$ is a *time-reversal* function.

Time-reversal function simplifies Magnus series further. It is straightforward that $\Omega(T) - \Omega'(T) = \sum_{i=1} \Omega_{2n}(T)$ with $\Omega_{2n}(T) = -\Omega_{2n}'(T)$ and $\Omega_{2n+1}(T) = \Omega_{2n+1}'(T)$. More generally, Ω_2 is the leading term in $\Omega_{\ge 2}(T)$ without assuming that the time-reversal function is a permutation. Its tail $\Omega_{\ge 3}(T)$ shrinks at $\mathcal{O}(T^3)$. Then, we can derive the rest of the lower bound:

$$\|\Omega(T) - \Omega'(T)\| = 2\|\Omega_{2n}(T)\| \ge 2c\|\Omega_2(T)\|, \qquad c \in (0, 1]. \tag{18}$$

On the other hand, App. C.1 implies abelian $\hat{\mathbf{S}}$ generates the same final state under path permutation. Thus, either x or x' should incur state mismatch:

$$\max_{x,x'} \Delta_h \ge \lambda c \|\Omega_2(T)\| \|h_0\|,$$

provides lower bound of approximation error incurred on small window. [10] □

Restricted SSM. Let us construct admissible inputs on interval $[0, 2T]$ by composing two T-length sequence[11]:

$$x_1 = \text{prefix}_1 \oplus \mathbf{xy}, \quad x_1' = \text{prefix}_1 \oplus \mathbf{yx}, \quad x_2 = \text{prefix}_2 \oplus \mathbf{xy}, \quad x_2' = \text{prefix}_2 \oplus \mathbf{yx}$$

with permutation map σ on $(T, 2T]$. Here we abuse notation and denote $(\mathbf{xy}, \mathbf{yx})$ to emphasize permutation.

Consider general SSM $\mathbf{S}$. As x_1 and x_1' shares prefix, $\Phi_{x_1}(T, 0) = \Phi_{x_1'}(T, 0)$. Shortly, we denote $\mathbf{M} := \Phi_{x_1}(2T, T) - \Phi_{x_1'}(2T, T)$. By simple subtraction:

$$h_1(2T) - h_1'(2T) = \mathbf{M} h_1(T) + \int_T^{2T} \Big(\Phi_{\mathbf{xy}}(2T, \tau)\mathbf{b}(\mathbf{xy}(\tau)) - \Phi_{\mathbf{yx}}(2T, \tau)\mathbf{b}(\mathbf{yx}(\tau))\Big) d\tau.$$

Similarly, we can obtain $h_2(2T) - h_2'(2T)$. Then, subtraction between terms cancels inhomogeneous part out:

$$\Big(h_1(2T) - h_1'(2T)\Big) - \Big(h_2(2T) - h_2'(2T)\Big) = \mathbf{M}(h_1(T) - h_2(T)). \tag{19}$$

On the other hand, for restricted $\hat{\mathbf{S}}$, $\mathbf{M} = \mathbf{0}$. Therefore, among these four inputs,

$$\max_{x_1, x_1', x_2, x_2'} \Delta_h \ge \frac{1}{4} \|\mathbf{M}(h_1(T) - h_2(T))\|, \tag{20}$$

with magnitude of $\mathbf{M}$ following Eq. 18.

With minimality condition, prefix_1 and prefix_2 can be chosen such that $h_1(T) \ne h_2(T)$ for $\mathbf{S}$. Up to bounded magnitude, we can make $h_1(T)$ and $h_2(T)$ be as far as possible. □

Local-to-global extension. Our proof so far is local, and here we show that the error can be accumulated over time under reasonable conditions.

Let us generalize the input path construction above to sequence on $[0, nT]$ for $n \ge 2$:

$$x_{n,i} = \text{prefix}_i \oplus (\mathbf{xy})^{\oplus n-1}, \quad x_{n,i}' = \text{prefix}_i \oplus (\mathbf{yx})^{\oplus n-1},$$

[10] For anyone interested in output error, the minimality condition guarantees existence of some common suffix sequence that generates output mismatch from state mismatch.

[11] Choice of $2T$ is arbitrary and two sequences need not have the same length.

for $i \in \{0, 1\}$. For the notational brevity, we denote:

$$\mathbf{V} := \Phi_{x_{1,1}}(2T, T) = \Phi_{\mathbf{xy}}, \quad \mathbf{U} := \Phi_{x'_{1,1}}(2T, T) = \Phi_{\mathbf{yx}}, \quad \mathbf{M} = \mathbf{V} - \mathbf{U}.$$

Then, by reusing the proof strategy above,

$$\Big(h_{n,1}(nT) - h'_{n,1}(nT)\Big) - \Big(h_{n,2}(nT) - h'_{n,2}(nT)\Big) = \Big(\mathbf{V}^{n-1} - \mathbf{U}^{n-1}\Big)(h_1(T) - h_2(T)).$$

Note that $n = 2$ recovers Eq. 19. Now we can apply the telescoping identity:

$$\mathbf{V}^{n-1} - \mathbf{U}^{n-1} = \sum_{i=0}^{n-2} \mathbf{V}^i(\mathbf{V} - \mathbf{U})\mathbf{U}^{n-2-i}. \tag{21}$$

Eq. 21 clearly shows why we need additional assumptions to guarantee the global error behavior. When flow is contractive, the early local error may quickly vanish. Furthermore, each local errors could cancel out, effectively reducing the lower bound.

Assuming each local errors are directionally aligned, we can derive a global lower bound. Let us denote:

$$v_i := \mathbf{V}^i\mathbf{M}\mathbf{U}^{n-2-i}(h_1(T) - h_2(T)), \quad v_{\mathbf{M}} := \mathbf{M}(h_1(T) - h_2(T)).$$

When there exists a unit vector u and a constant $\alpha > 0$ such that:

$$\langle u, v_i \rangle \geq \alpha \|v_{\mathbf{M}}\| \quad \forall i \in \{0, \ldots, n-2\}, \tag{22}$$

then we can derive a global lower bound:

$$\Big\| \Big(\mathbf{V}^{n-1} - \mathbf{U}^{n-1}\Big)(h_1(T) - h_2(T)) \Big\| \geq \alpha(n-1)\|v_{\mathbf{M}}\|. \tag{23}$$

In summary, Thm. 3.2 is a local statement, which extends to global under reasonable assumptions. □

C.3 Proof of Lemma 3.3

We first show abelian k-layer SSM has up to k derived length, and naturally derives $2k$ for restricted SSMs. To obtain notational simplicity, we choose vector field notation for this proof.

Abelian SSM. Consider abelian k-layer SSM with vector field $F(t, h, x) = (F_0, F_1, \ldots, F_{k-1})$ on a manifold $\mathcal{M} \cong \mathbb{R}^n, \mathcal{M}_i \cong \mathbb{R}^{n_i}$. Assume (t, x) is fixed. For the layer index i and state coordinate index a, we trivially extend the local vector field:

$$h_i \in \mathcal{M}_i, \quad \partial_{i,a} = \frac{\partial}{\partial h_i^a}, \quad \bar{F}_i(h) = \sum_{a=1}^{n_i} F_i^a(h_{\leq i})\partial_{i,a} = (\underbrace{0, \ldots, 0}_{0:i-1}, F_i, \underbrace{0, \ldots, 0}_{i+1:k-1}), \tag{24}$$

such that the coefficient function $(\bar{F}_i)_j^c = 0$ for any $j \neq i$ and $\bar{F}_i^c = F_i^c$. Then the global vector field $F(h) = \sum_{i=0}^{k-1} \bar{F}_i(h)$.

Now consider two fixed control inputs x, y and corresponding vector fields X, Y such that $X(h) = F(h, x)$ and $Y(h) = F(h, y)$. Following Eq. 24, let us denote:

$$X = \sum_{i=0}^{k-1} X_i, \quad X_i = \sum_{a=1}^{n_i} X_i^a \partial_{i,a},$$

and similarly for Y.

With Eq. 7 let us construct their Lie bracket $[X, Y] = \sum_{i,j}[X_i, Y_j]$. When $i = j$, trivially $[X_i, Y_i] = 0$ from abelian definition. For $i < j$, note that X_i is independent of h_j by construction, therefore:

$$[X_i, Y_j]_i^c = \sum_{a=1}^{n_i} X_i^a \partial_{i,a}(Y_j)_i^c - \sum_{b=1}^{n_j} Y_j^b \partial_{j,b} X_i^c = 0, \tag{25}$$

$$[X_i, Y_j]_j^c = \sum_{a=1}^{n_i} X_i^a \partial_{i,a} Y_j^c - \sum_{b=1}^{n_j} Y_j^b \partial_{j,b}(X_i)_j^c = \sum_{a=1}^{n_i} X_i^a \partial_{i,a} Y_j^c, \tag{26}$$

such that nonzero component of $[X_i, Y_j]$ only lives in the j-th layer.

Eq. 26 provides a chain of ideals. Let us denote $\mathfrak{g}$ be the Lie algebra generated by the collection of global vector fields F. We define its ideal $I_l := \{X \in \mathfrak{g} \mid X(h) = \sum_{i \geq l} X_i(h)\}$ with zero component below layer l. Then:

$$[X_i, Y_j] \in I_{\max(i,j)}, \quad [X, Y] \in I_1, \quad \mathfrak{g}^{(n)} \in I_n,$$

which allows us to construct a chain,

$$\mathfrak{g} = I_0 \supseteq I_1 \supseteq \cdots \supseteq I_{k-1} \supseteq I_k = \{0\}, \tag{27}$$

where quotients of ideals, I_i/I_{i+1}, are all abelian by construction. Eq. 27 is not necessarily the shortest sequence of abelian ideals, so abelian k-layer SSM has maximally k derived length. □

Restricted SSM. It is trivial from Eq. 27. Each I_i/I_{i+1} is constructed to be restricted, which by themselves are solvable with derived length 2. Again, the series is not necessarily the shortest derived series, and restricted k-layer SSM has derived length at most $2k$. □

C.4 Proof of Theorem 3.4

Proof. Consider lifted Lie equation of $\mathbf{S}_\mathfrak{g}$ with a matrix Lie algebra $\mathfrak{g} \subset \mathfrak{gl}(V)$:

$$\dot{\mathbf{G}}(t) = \mathbf{A}_\mathfrak{g}(x(t))\mathbf{G}(t), \quad \mathbf{G}(0) = I_V, \tag{28}$$

where $\mathbf{A}_\mathfrak{g}(x(t)) \in \mathfrak{g}$. Given the Lie algebra extension:

$$0 \to \mathfrak{a} \xrightarrow{i} \mathfrak{g} \xrightarrow{\theta} \mathfrak{h} \to 0, \tag{29}$$

our interest is to decompose the integral curve $\mathbf{G}(t)$ into curves integrating ideal $\mathfrak{a}$ and quotient $\mathfrak{h} \cong \mathfrak{g}/\mathfrak{a}$.

Let G be a matrix Lie group locally integrating $\mathfrak{g}$ near neighborhood of identity. Again locally, let A be the normal subgroup integrating the ideal $\mathfrak{a}$, and H for the quotient Lie algebra $\mathfrak{h}$. We do not discuss global property of (A, G, H), and shortly call them as local matrix Lie groups. Restricted to sufficiently small neighborhoods,

$$1 \to A \xrightarrow{\boldsymbol{i}} G \xrightarrow{\Theta} H \to 1,$$

where differentiation at the identity recovers Eq. 29. We follow common notational convention and do not distinguish A or $\mathfrak{a}$ from $\boldsymbol{i}(A)$ or $i(\mathfrak{a})$, respectively.

For the quotient part, its Lie equation is straightforward. We often silence input variables without confusion for the clarify. With $\theta = d\Theta_e$:

$$\begin{aligned} &\mathbf{H} := \Theta(\mathbf{G}) \in H, \quad \bar{\mathbf{A}} := \theta(\mathbf{A}_\mathfrak{g}) \in \mathfrak{h}, \\ &\dot{\mathbf{H}} = \frac{d}{dt}\Theta(\mathbf{G}) = (d\Theta_\mathbf{G})(\dot{\mathbf{G}}) = \theta(\mathbf{A}_\mathfrak{g})\Theta(\mathbf{G}) = \bar{\mathbf{A}}\mathbf{H}. \end{aligned} \tag{30}$$

Now let us construct the Lie equation generated by ideal $\mathfrak{a}$. We take a smooth local section,

$$\mathbf{s} : H \to G, \quad \Theta(\mathbf{s}(\mathbf{H}(t))) = \mathbf{H}(t).$$

As $\ker \Theta = \mathrm{im}\boldsymbol{i}$ from the definition of the short exact sequence, we can characterize the ideal. Given an inversion map on group $\iota : G \to G$,

$$\mathbf{a}(t) := \mathbf{Q}^{-1}(t)\mathbf{G}(t), \quad \mathbf{Q}(t) := \mathbf{s}(\mathbf{H}(t)) \in G, \quad \mathbf{Q}^{-1}(t) = \iota(\mathbf{Q}(t)). \tag{31}$$

We can confirm $\mathbf{a}(t)$ is indeed an element of A, sitting in $\ker \Theta$:

$$\Theta(\mathbf{Q}^{-1}\mathbf{G}) = \Theta(\mathbf{Q}^{-1})\Theta(\mathbf{G}) = \Theta(\iota(\mathbf{s}(\mathbf{H})))\mathbf{H} = \mathbf{H}^{-1}\mathbf{H} = e_H,$$

therefore any $\mathbf{a}(t) \in A$. Differentiation gives us the Lie equation of $\mathfrak{a}$.

$$\dot{\mathbf{a}} = \left(\frac{d}{dt}\mathbf{Q}^{-1}\right)\mathbf{G} + \mathbf{Q}^{-1}\dot{\mathbf{G}} = -\mathbf{Q}^{-1}\dot{\mathbf{Q}}\mathbf{Q}^{-1}\mathbf{G} + \mathbf{Q}^{-1}\dot{\mathbf{G}} = (\mathbf{Q}^{-1}\mathbf{A}_\mathfrak{g}\mathbf{Q} - \mathbf{Q}^{-1}\dot{\mathbf{Q}})\mathbf{a}, \tag{32}$$

where $\dot{\mathbf{Q}}(t)$ follows,

$$\dot{\mathbf{Q}} = (d\mathbf{s})_{\mathbf{H}}(\dot{\mathbf{H}}) = (d\mathbf{s})_{\mathbf{H}}(\bar{\mathbf{A}}\mathbf{H}).$$

Again let us confirm $(\mathbf{Q}^{-1}\mathbf{A}_{\mathfrak{g}}\mathbf{Q} - \mathbf{Q}^{-1}\dot{\mathbf{Q}}) \in \mathfrak{a}$ to ensure Eq. 32 is indeed the Lie equation of $\mathfrak{a}$.

First, note that $\mathbf{Q}^{-1}\mathbf{A}_{\mathfrak{g}}\mathbf{Q}$ is an adjoint action (or conjugation) of Lie group on its Lie algebra (Robbin & Salamon, 2022). Being matrix Lie group and algebra, it is a standard matrix calculus:

$$\theta(\mathbf{Q}^{-1}\mathbf{A}_{\mathfrak{g}}\mathbf{Q}) = \theta(\mathrm{Ad}_{\mathbf{Q}^{-1}}\mathbf{A}_{\mathfrak{g}}) = \mathrm{Ad}_{\Theta(\mathbf{Q}^{-1})}\big(\theta(\mathbf{A}_{\mathfrak{g}})\big) = \Theta(\mathbf{Q}^{-1})\theta(\mathbf{A}_{\mathfrak{g}})\Theta(\mathbf{Q}) = \mathbf{H}^{-1}\bar{\mathbf{A}}\mathbf{H}.$$

The second term $\mathbf{Q}^{-1}\dot{\mathbf{Q}}$ is simply derived from Eq. 30:

$$\theta(\mathbf{Q}^{-1}\dot{\mathbf{Q}}) = \mathbf{H}^{-1}\dot{\mathbf{H}} = \mathbf{H}^{-1}\bar{\mathbf{A}}\mathbf{H},$$

then terms cancel out,

$$\theta(\mathbf{Q}^{-1}\mathbf{A}_{\mathfrak{g}}\mathbf{Q} - \mathbf{Q}^{-1}\dot{\mathbf{Q}}) = 0.$$

Recall from Eq. 29 that $\ker\theta = \mathrm{im}\ i$. Therefore, Eq. 32 is a Lie equation generated by the ideal.

Together, Eq. 28 locally admits 2-layer decomposition:

$$\mathbf{G}(t) = \mathbf{s}(\mathbf{H}(t))\mathbf{a}(t), \quad \dot{\mathbf{H}}(t) = \bar{\mathbf{A}}(x(t))\mathbf{H}(t), \quad \dot{\mathbf{a}}(t) = \bar{\mathbf{Q}}(x(t), \mathbf{H}(t))\mathbf{a}(t),$$
$$\bar{\mathbf{Q}}(x(t), \mathbf{H}(t)) = \mathbf{Q}^{-1}(t)\mathbf{A}_{\mathfrak{g}}(x(t))\mathbf{Q}(t) - \mathbf{Q}^{-1}(t)\dot{\mathbf{Q}}(t).$$

Therefore, the lifted Lie equation of $\mathbf{S}_{\mathfrak{g}}$ decomposes into 2 Lie equations where each represents ideal $\mathfrak{a}$ and quotient $\mathfrak{h}$. To project Lie equation back to SSM, we vectorize with the Kronecker product $\otimes$. For example, in Eq. 28, consider $\mathfrak{g} \subset \mathfrak{gl}(n, \mathbb{R})$. Let $\mathbf{z} \in \mathbb{R}^{n^2}$ be a flattened $\mathbf{G} \in \mathbb{R}^{n\times n}$:

$$\dot{\mathbf{z}}(t) = \big(I_n \otimes \mathbf{A}_{\mathfrak{g}}(x(t))\big)\mathbf{z}(t), \quad \mathbf{z}(0) = \mathrm{vec}(I_n). \tag{33}$$

Therefore, $\mathbf{S}_{\mathfrak{g}}$ can be decomposed and simulated by 2-layer deep SSM. □

C.5 Proof of Corollary 3.5

Proof. Recall that a solvable Lie algebra $\mathfrak{g}$ with derived length k satisfies following:

$$\mathfrak{g} = \mathfrak{g}^{(0)} \supseteq \cdots \supseteq \mathfrak{g}^{(k-1)} \supseteq \mathfrak{g}^{(k)} = \{0\}, \quad 0 \to \mathfrak{g}^{(k-1)} \xrightarrow{i} \mathfrak{g} \xrightarrow{\theta} \mathfrak{g}/\mathfrak{g}^{(k-1)} \to 0. \tag{34}$$

We invoke Thm. 3.4 to decompose $\mathbf{S}_{\mathfrak{g}}$ by replacing $\mathfrak{a}$ with $\mathfrak{g}^{(k-1)}$ and $\mathfrak{h}$ with $\mathfrak{g}/\mathfrak{g}^{(k-1)}$ in Eq. 29. As the base (quotient) layer $\mathfrak{g}/\mathfrak{g}^{(k-1)}$ is again solvable with derived length $k-1$, we can again faithfully represent $\mathfrak{g}/\mathfrak{g}^{(k-1)}$ as a matrix Lie algebra and iteratively apply the same decomposition near identity.

Therefore, a solvable $\mathbf{S}_{\mathfrak{g}}$ can be decomposed and simulated by an abelian deep SSM. □

C.6 Proof of Corollary 3.7

Proof. The flow of non-solvable system $\mathbf{S}_{\mathfrak{g}}$ demands infinite nontrivial terms in Magnus expansion. Nilpotentization, or truncation, of the Magnus expansion is an approximation technique that matches up to c-th order Magnus term (Iserles et al., 2011; Blanes et al., 2009).

Let $(A_1, \ldots, A_n)$ be a basis of $\mathfrak{g}$, and write $A(t) = \sum_i a_i(t)A_i$. We formally replace each A_i with a symbol X_i to construct the *free* Lie algebra $\mathcal{L}$, naturally graded as:

$$\mathcal{L} = \bigoplus_{k\geq 1} \mathcal{L}_k,$$

where $\mathcal{L}_k$ is spanned by formal Lie brackets containing k symbols. The degreewise evaluation map sends each X_i to A_i and each formal Lie bracket to corresponding Lie bracket in $\mathfrak{g}$.

As noted in Sec. B.4, recall the c-th order Magnus term of $\mathfrak{g}$ consists of Lie brackets of c elements. By construction, $\mathcal{L}_{\leq c} = \mathcal{L}/\bigoplus_{k>c}\mathcal{L}_k$ retains exactly the formal bracket information needed to match Magnus terms up to order c. $\mathcal{L}_{\leq c}$ is also a finite-dimensional class c nilpotent Lie algebra.

Together, we can construct a class c nilpotent Lie algebra with evaluation map that matches the Magnus series of $\mathfrak{g}$ up to order c, while any higher order terms vanish. This construction applies to any finite-dimensional $\mathfrak{g}$.

Now we invoke Cor. 3.5. Recall that any nilpotent Lie algebra is solvable. Specifically, for any Lie algebra $\mathfrak{h}$, the derived series and lower central series satisfy:

$$\mathfrak{h}^{(k)} \subseteq \mathfrak{h}^{2^k-1}.$$

Therefore, if $\mathfrak{h}$ is nilpotent of class $2^k - 1$, its derived length is upper bounded by k (Burde, 2012). In other words, if $\mathfrak{h}$ is nilpotent of class c, its derived length k is upper bounded by:

$$k \leq \lceil \log_2(c+1) \rceil = \lfloor \log_2 c \rfloor + 1,$$

as c is an integer. Therefore, there exists abelian k-layer SSM with smooth output map that decomposes the nilpotent Lie algebra $\mathcal{L}_{\leq c}$. Taken together, constructed abelian deep SSM can match the Magnus series of $\mathfrak{g}$ up to order c.

Following, the leading term of Magnus series truncation error is the 2^k-th order term, $\|\Omega_{2^k}\| \lesssim \mathcal{O}(\epsilon^{2^k})$ (Blanes et al., 2009). Therefore, adding layers can exponentially increase the truncation order and decrease the magnitude of leading error term. □

Local-to-global extension. Construction of the approximating abelian deep SSM is algebraic and global. What remains local is the Magnus series itself, as the Magnus series can diverge for large T. By partitioning the time interval into sufficiently small subintervals where Magnus series converges, we can provide a global error bound without further assumptions.

Consider a partition of time $0 = t_0 < \cdots < t_N = T$ where each subinterval is sufficiently small enough. Per each subinterval, we denote the flow of target SSM $\mathbf{S}_{\mathfrak{g}}$ as $\mathbf{U}_i = \Phi(t_i, t_{i-1})$, and the truncated flow generated by the approximating deep abelian SSM on the same window as $\mathbf{V}_i = \hat{\Phi}(t_i, t_{i-1})$. Composing the local flows yields the global flow:

$$\prod_{i=1}^{N} \mathbf{U}_i = \Phi(T, 0), \quad \prod_{i=1}^{N} \mathbf{V}_i = \hat{\Phi}(T, 0).$$

Then, by the telescoping identity:

$$\|\Phi(T,0) - \hat{\Phi}(T,0)\| \leq \sum_{i=1}^{N} \left(\prod_{j>i} \|\mathbf{U}_j\| \right) \delta_i \left(\prod_{j<i} \|\mathbf{V}_j\| \right), \quad \delta_i = \|\mathbf{U}_i - \mathbf{V}_i\|.$$

Under our regularity assumptions (Sec. 2.2), we can bound the local flow norms for some $\alpha > 0$:

$$\max\left(\|\mathbf{U}_i\|, \|\mathbf{V}_i\|\right) \leq \exp(\alpha(t_i - t_{i-1})).$$

Therefore,

$$\|\Phi(T,0) - \hat{\Phi}(T,0)\| \leq \exp(\alpha T) \sum_{i=1}^{N} \delta_i \leq \exp(\alpha T) N \delta_{\max}, \quad \delta_{\max} = \max_i \delta_i. \tag{35}$$

Each δ_i follows the doubling-exponent truncation error rate, $\mathcal{O}(\epsilon_i^{2^k})$. The window count N linearly contributes to the global error, but finer partitions can reduce ϵ_i and $\delta_{\max}$. Therefore, when the local generator mass is uniformly distributed, refinement of the partition can tighten the global error bound while increasing the window count. □

C.7 Proof of Proposition 3.8

Here we begin with a key concept for the proof. Notation follows Def. 2.8.

Definition C.1 (Lyndon words (Avdieiev & Tsymbaliuk, 2024))**.** Let an alphabet set $\mathcal{A}$ be totally ordered. For any words, we introduce the *lexicographical* order on $\mathcal{A}^*$:

$$a_1 a_2 \dots a_i < b_1 b_2 \dots b_j \quad \text{if} \begin{cases} a_1 = b_1, a_2 = b_2, \dots, a_n = b_n, a_{n+1} < b_{n+1} \text{ for some } n \geq 0 \\ \text{or} \\ a_1 = b_1, \dots, a_i = b_i \text{ and } i < j. \end{cases} \tag{36}$$

Then, a word $w = a_1 a_2 \dots a_i$ is *Lyndon* if it is smaller than all of its cyclic permutations:

$$a_1 \dots a_{n-1} a_n \dots a_i < a_n \dots a_i a_1 \dots a_{n-1} \quad \forall n \in \{2, \dots, i\}.$$

and we call $a_1 \dots a_{n-1}$ and $a_n \dots a_i$ as *prefix* and *suffix* of the word, respectively.

Proof. With the definition of word problems in Def. 2.8, the Chen-Fox-Lyndon theorem states that any words have a unique *canonical* factorization in Lyndon words (Chen et al., 1958; Melançon & Reutenauer, 1989; Avdieiev & Tsymbaliuk, 2024). Let us denote a set of Lyndon words on $\mathcal{A}$ as L, and its subset up to length T as L_T. For any word w,

$$w = l_1 l_2 \dots l_k, \quad l_1 \geq l_2 \cdots \geq l_k, \quad l_i \in L.$$

Notice that the canonical factorization of a Lyndon word is trivially itself. Instead, Lyndon word has a *costandard* factorization. For any Lyndon word l,

$$l = l_1 l_2, \quad l_1, l_2 \in L,$$

where l_2 is the longest proper Lyndon suffix of l.

Clearly, the canonical factorization provides the key insight in our proof. Intuitively, when word length is bounded at T, it suffices to "memorize" Lyndon words and their composition up to length T to solve the problem.

It is known that L provides a basis of free Lie algebra on $\mathcal{A}$ (Avdieiev & Tsymbaliuk, 2024; Walker et al., 2024). Therefore, we can leverage the similar proof strategy as in App. C.6 to construct a finite-dimensional nilpotent free Lie algebra. Let $\mathfrak{L}$ be a free Lie algebra generated by a finite set $\{e_a\}_{a \in \mathcal{A}}$, where each element is labeled by an alphabet. The bracket rule is given:

$$\mathsf{b}[a] = e_a \in \mathfrak{L} \text{ for } a \in \mathcal{A}, \quad \mathsf{b}[l] = [\mathsf{b}[l_1], \mathsf{b}[l_2]], \tag{37}$$

with $l = l_1 l_2$ be the costandard factorization. For a simple example, consider a Lyndon word $a_1 a_2 a_3$ with costandard factorization $(a_1, a_2 a_3)$. Then,

$$\mathsf{b}[a_1 a_2 a_3] = [\mathsf{b}[a_1], \mathsf{b}[a_2 a_3]] = [e_{a_1}, [\mathsf{b}[a_2], \mathsf{b}[a_3]]] = [e_{a_1}, [e_{a_2}, e_{a_3}]] \in \mathfrak{L}.$$

Observe that Lyndon word length 3 corresponds to the second-order Lie bracket. Analogously, Lyndon word length T corresponds to $(T-1)$ Lie bracket operations.

Therefore, truncating the Lyndon words at L_T implies that we retain the lower central series of $\mathfrak{L}$ up to $(T-1)$-th order, or $\mathfrak{L}_{\leq T} = \mathfrak{L} / \bigoplus_{i > T} \mathfrak{L}_i$. Again, $\mathfrak{L}_{\leq T}$ is a finite-dimensional nilpotent Lie algebra with class T.

Now we invoke Cor. 3.5 just as we did in App. C.6. For a class-T nilpotent $\mathbf{S}_{\mathfrak{g}}$, its derived length is at most $\lfloor \log_2 T \rfloor + 1$. Therefore, we can construct abelian $\big(\lfloor \log_2 T \rfloor + 1\big)$-layer SSM plus output layer to simulate flow (Burde, 2012). Each alphabet symbol is then a piecewise-constant external input signal. □

C.8 Proof of Corollary 3.9

The construction is natural from Prop. 3.8. Following logics and notations in App. C.7, we highlight that the number of Lyndon words determines the dimension of $\mathfrak{g}$. Let the number of alphabet symbols $|\mathcal{A}| = n$, and denote $\mathfrak{g}(T, n) = \mathfrak{L}_{\leq T}$. Then, *Witt's formula* provides following (Magnus et al., 1966; Bouallegue & Naimi, 2010):

$$\dim \mathfrak{g}(T, n) = \sum_{m=1}^{T} \frac{1}{m} \sum_{d \mid m} n^d \, \mu\big(\frac{m}{d}\big),$$

where μ is the Möbius function.

D Experimental Details

Here we provide further experimental details. Upon acceptance, we will release all our code used to produce our experimental results on a public GitHub repository. We conducted all our experiments using a single GPU (A100 or H100) and no run lasted longer than three hours.

D.1 WORD PROBLEMS

Task formulation. We encode each of the word problems used in Table 2 as a causal sequence labeling problem. Each word problem has a finite set of group elements. A task sequence is constructed by assigning a unique index to each group element in the corresponding word problem. Given a sequence of indexed transformations, the task is to predict, at each position in the sequence, the class label corresponding to the group element obtained by composing the transformations in sequence. Training and test data sets are constructed by randomly sampling such sequences.

In practice, the corresponding word problem sequence is prepended with an artificial *beginning* of sequence (BOS) token. Conventionally, we assign the same BOS token as the target. BOS token allows the model to *burn in* an initial state before processing the actual problem sequence.

In our preliminary studies, we also tried to explicitly learn the initial states of the sequence models (for GLA and Mamba variants) as part of the trainable parameters. As we did not find this to be helpful, all our experiments are conducted without learning the initial states. Modes are trained to make prediction at each position in the sequence, up to the maximum length of the data.

A sequence-level prediction is only counted as correct if the model successfully predicts all the elements in the sequence correctly.

Model configurations. Models are separately trained on each word problem. Given that each word problem has a different number of group element, their base vocabulary size is different for each case (namely, 2 for parity/C_2, 3 for C_3, 8 for D_8 and H_3, 6 for S_3, 24 for S_4, and 60 for A_5). In addition to these base vocabulary, additional special tokens such as the padding token, including BOS, are allocated. Such a choice is conventional for most modern sequence model implementations, which we follow; none of them is required in our tasks except BOS.

The numbers reported in Table 2 are the best results from training all hyper-parameters configuration listed in Table 3 for three random seeds each for 100 epochs. We monitor length generalization performance, and stop early if the model achieves 100% sequence-level prediction accuracy. This corresponds to the standard evaluation protocol on word problems (Merrill et al., 2024; Siems et al., 2025), and serves as an operational definition of a model that is unsuccessful at performing a certain task. For all tasks except the simplest C_2, we use 500k training sequences, and evaluation is conducted on 1000 test sequences. As a technical side note, Deltaproduct achieved 100% generalization accuracy with only 100K training sequences for all the tasks; nevertheless, to potentially help other models, we increased it to 500K.

We use sequence length of 128 for training and 256 for testing.

Implementation. We use `flash-linear-attention` (Yang & Zhang, 2024) to implement transformer, GLA, and DeltaProduct models. We use public code from the respective papers for signed Mamba (Grazzi et al., 2025) and AUSSM (Karuvally et al., 2025).

Instead of signed Mamba, one could also use complex or signed diagonals for GLAs in theory. However, GLA's efficient implementations of the chunk-wise training algorithm assume positive diagonals as also noted by Grazzi et al. (2025). Therefore, we focus on signed Mamba only.

Note that the AUSSM implementation requires us to disable mixed precision for compatibility with the use of complex numbers. Furthermore, the AUSSM backward kernel assumes sequence length to be a multiple of 8, requiring some care about the length of sequences we feed in practice.

Model class. AUSSM is complex-valued, but this does not leave our SSM model class. n-dimensional complex diagonal SSMs are equivalent to m-dimensional real block-diagonal SSMs (Orvieto et al., 2023). Algebraically, *complexifying* Lie algebras on real field is a standard practice. For example, any real solvable finite-dimensional Lie algebra complexifies to complex solvable Lie algebra $\mathfrak{g}$. This provides additional clarity as Lie's theorem allows us to represent any $\mathfrak{g}$ as an upper triangular matrix over $\mathbb{C}$ (Hall, 2015).

The transformer is included as a reference model that is parallelizable and order-symmetric with respect to the layer input. We also leverage interpretation in Choromanski et al. (2021); Peng et al.

Table 3: Hyper-parameter search space. The number of heads is set to 8 for transformer variants, and d_state (i.e., the "head dimension") is set to 16 for SSMs. Each configuration is run with 3 random seeds.

Parameters	Values
Hidden size	{64, 128, 256}
Learning rate	{1e-2, 1e-3, 3e-4, 1e-4}
Batch size	{256, 2048}

(2021) and Sieber et al. (2024). Note that the softmax self-attention is algebraically order-symmetric and thus abelian.

It is straightforward to relate GLA, (Signed) Mamba, and DeltaProduct with our SSM model class.

D.2 A_5-ROTATED VECTOR PREDICTION TASK

Here we provide further details of the rotation task in Sec. 4.3 and Fig. 3. Note that a task description is as provided in the main text: the target vectors are generated by using the matrices provided in the main text.

Figure 3a All the 60 elements of A_5 task can be expressed using $\{\mathbf{P}, \mathbf{R} | \mathbf{P}^3 = \mathbf{R}^5 = (\mathbf{PR})^2 = \mathbf{I}\}$. Under uniformly sampled symbols, the target is uniform over the A_5 orbit of the initial vector, whose mean is zero. Therefore, the best mean predictor is simply 0 with coordinate-wise MSE $1/3$. For each model, we report the sequence length that first exceeds half of the best mean predictor MSE, $1/6$, on 1k held-out test sequences. The hollow marks correspond to 1-layer models whose hidden size is 8 times larger. The position-wise training and test results are reported in Fig. 4.

We also conducted experiments with AUSSM. However, we did not report them in the main text as we obtained generally subpar performance. We conjecture this as the practical training issue. In particular, complex-valued models are notoriously known to have practical trainability issue (see, e.g., Elelimy et al. (2024)). For the sake of completeness, we report the corresponding results here in Fig. 5.

Figure 3b As 1-layer error quickly saturates within A_5 rotation setups (Fig. 4), probing the error scale against commutator mass was done with fractional rotations. Specifically, each A_5 symbol is faithfully represented as a rotation matrix, and its f-th root is used to rotate vector instead. This allows us to probe a smaller generator mass regime, which aligns with our theory better. Fractional generators $\{\mathbf{p}, \mathbf{r} | \mathbf{p}^f = \mathbf{P}, \mathbf{r}^f = \mathbf{R}\}$ now generate a dense subgroup of $SO(3, \mathbb{R})$ containing A_5.

Probe words follow the input construction in App. C.2. We chose short commutator words in a form of:

$$C_a = (\mathbf{p}^a \mathbf{r}^a \mathbf{p}^{-a} \mathbf{r}^{-a})^{4/a}, \quad a \in \{1, 2, 4\},$$

so that we test the error scaling without saturation. Note that for chosen symbol set every commutator words uses the same symbol count, so its generator mass is fixed. The design allows us to vary the commutator mass $\|\Omega_2\|$ of sequences, especially in $\|\Omega_2\| \propto a$. Each commutator words are preceded by two fixed prefix sequences, corresponding to $\{\text{prefix}_1, \text{prefix}_2\}$ in App. C.2. We report the worst error read at the final position per model, matching the prediction of Thm. 3.2 (Eq. 20). We have tested $f = \{4, 5, 6\}$ and achieved similar results. We report $f = 6$ result in the main text.

Hyper-parameters. All models in Fig. 3a are trained using a batch size of 256, learning rate of $1e^{-3}$, with a hidden size of 128. Baseline 1-layer models in Fig. 3a are trained with a hidden size of 1024. The number of layers is varied from 1 to 8. All the models are trained on 500k training sequences of length 128 for 100 epochs. For deeper models, we have investigated larger hyper-parameter space following Table 3, but we did not obtain any better results.

All models in Fig. 3b are trained with same hyper-parameters but with a learning rate of $1e^{-2}$, which compensates for the smaller errors of the fractional setup. $1e^{-3}$ gives similar results.

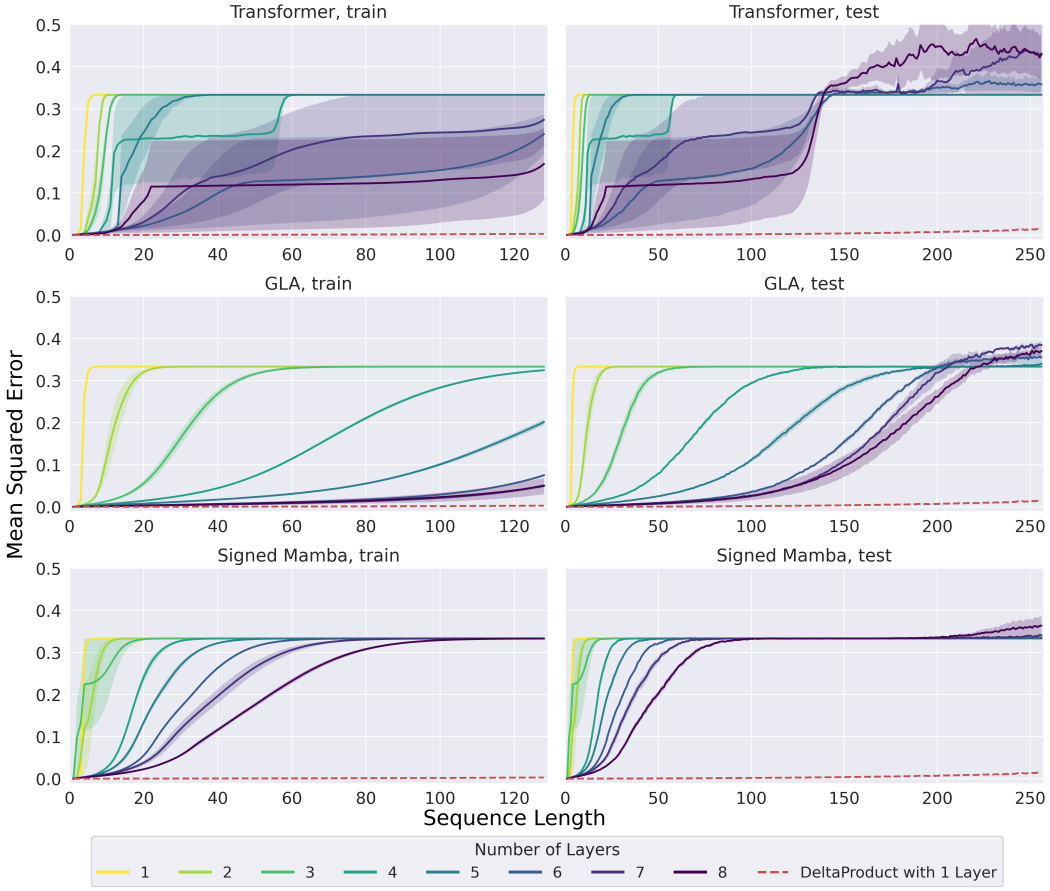


Figure 4: Mean squared error on the rotated vector prediction task at various sequence lengths for transformer, GLA, and signed Mamba, with different numbers of layers. Left and right columns illustrate the results on training and test sets, respectively. Standard error is shown using 3 seeds. Performance of DeltaProduct with 4 Householder products is shown as a reference.

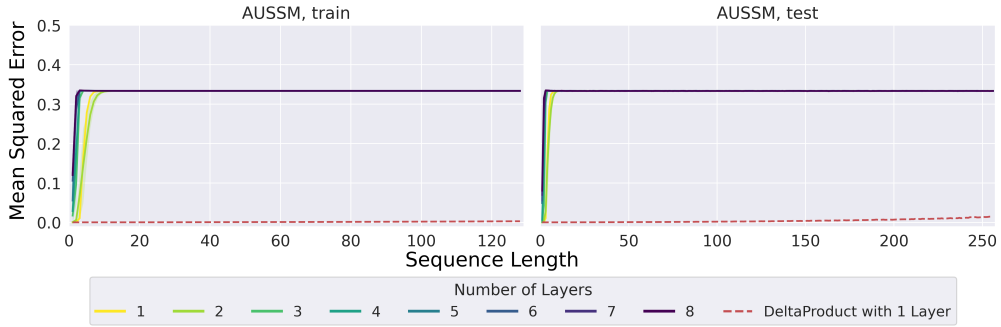


Figure 5: Mean squared loss on the rotated vector prediction task at various sequence length for AUSSM, with different numbers of layers, on train (left) and test (right) sets. Standard error is shown using 3 seeds. Performance of DeltaProduct with 4 Householder products is shown as a reference.