CPTP-SSM: EXPANDING STATE TRACKING TO CONTINUOUS GROUPS WITH UNCERTAINTY

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent selective state-space models (SSMs) have increased transition expressivity to support the state tracking of finite-state automata (FSA). Recently, SSMs are increasingly applied to long sequences from the real world beyond language modeling. We extend state tracking to continuous groups with uncertainty, requiring input-selective transitions that can be either conservative on the continuous state or dissipative under uncertainty. Rather than enlarging the transition, we constrain the state space and admit only transitions that map it into itself. We introduce CPTP-SSM, a selective SSM on the compact convex set of qudit density matrices. The transitions are completely positive trace-preserving (CPTP) maps, a single family that admits both conservative and dissipative transitions. Expressing the Kraus representation in the generalized Gell-Mann basis turns each transition into a real affine map, enabling the recurrence to run as a parallel associative scan. We establish expressivity and stability guarantees for CPTP-SSM and further characterize how the qudit dimension governs the representable groups, while the Kraus rank governs the capacity for dissipation. Experimentally, CPTP-SSM achieves strong length extrapolation on both finite and continuous state tracking, with competitive performance in long time-series classification and continuous sensor prediction.

1 INTRODUCTION

State-space models (SSMs) have emerged as an efficient alternative to Transformers for long-sequence modeling, combining parallel training across the sequence length with linear-time inference at constant memory (Gu et al., 2022; Smith et al., 2023; Orvieto et al., 2023). Mamba adopts an input-selective transition with a diagonal matrix for efficiency and reaches language modeling performance comparable to Transformers (Gu & Dao, 2024). Despite this success, SSMs whose transition is a diagonal matrix are restricted in state tracking tasks such as finite-state automaton (FSA) emulation, tasks that underlie language model abilities such as code execution and program analysis (Merrill et al., 2024). Subsequent analyses established that the structure of the transition determines the expressivity of an SSM (Muca Cirone et al., 2024; Sarrof et al., 2024). This motivated a line of work calibrating transition expressivity through FSA emulation, including PD-SSM, whose column-one-hot-times-diagonal transition can emulate any FSA in a single layer (Terzić et al., 2025). This transition expressivity can be understood through the Krohn–Rhodes decomposition, which shows that every FSA can be realized as a cascade of permutation-reset automata (Krohn & Rhodes, 1965). Permutations are reversible transitions that preserve distinctions between states, whereas resets are irreversible transitions that erase them.

Recently, SSMs are increasingly applied beyond language to long sequences in vision (Zhu et al., 2024; Liu et al., 2024), robotics (Jia et al., 2024; Yoo et al., 2026), and sensing (Bhirangi et al., 2024; Qin et al., 2026). These sequences introduce a different state-tracking challenge: their underlying states evolve continuously, while observations are often uncertain. Inertial orientation tracking provides a representative example. The orientation evolves through the ordered composition of three-dimensional rotations, while each measured increment is corrupted by noise (Titterton & Weston, 2004; Forster et al., 2017).

The permutation and reset characterization of finite-state computation suggests two functional roles for a recurrent transition: preserving distinctions through reversible updates and erasing them through irreversible ones. We carry these roles to continuous dynamics with uncertain increments. On

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.

continuous state spaces, the reversible role of permutations extends to continuous group actions, while the information-erasing role of resets extends to graded contraction under uncertainty. These requirements go beyond finite-state emulation, motivating input-selective transitions in which a single update can preserve some state distinctions while contracting others.

While expressivity establishes what a model can represent in theory, length generalization determines whether it tracks the state in practice. Sequences at deployment routinely exceed training lengths, and these long sequences are precisely what motivates SSMs. Accuracy within the training lengths can be reached through shortcuts even without the required expressivity (Liu et al., 2023). Recent studies report a gap between expressivity and learnability, in which models that can represent a task still fail to generalize in length (Shakerinava et al., 2026; Chung et al., 2026). Buitrago & Gu (2025) analyze length-generalization failures of recurrent models in language modeling as *unexplored states hypothesis*, attributing them to states never reached during training.

We introduce CPTP-SSM, a selective SSM with qudit density-matrix states and input-selected completely positive trace-preserving (CPTP) transitions. CPTP-SSM provides a single architecture for finite-state tracking and continuous-group tracking under uncertainty while retaining efficient long-sequence computation through associative scan. Using normalized Kraus operators and real affine coordinates, we characterize its reversible and dissipative dynamics, including group tracking and contraction. Empirically, CPTP-SSM shows strong length extrapolation on group word problems, accurately tracks noisy $\mathrm{SO}(3)$ dynamics, and achieves strong performance on long time-series classification and real-world sensor prediction.

2 Background

2.1 State-Space Models and Parallel Scan

A state-space layer carries a state $\mathbf{x}_t \in \mathbb{R}^N$ along the input tokens u_t through the recurrence

$$\mathbf{x}_t = \mathbf{A}(u_t)\,\mathbf{x}_{t-1} + \mathbf{b}(u_t), \tag{1}$$

where the transition $\mathbf{A}(u_t)$ and the offset $\mathbf{b}(u_t)$ are selected by the current token (Gu & Dao, 2024). We call the set $\mathcal{F}$ of admissible transitions the transition family of the layer. Affine pairs compose associatively, $(\mathbf{A}_2, \mathbf{b}_2) \circ (\mathbf{A}_1, \mathbf{b}_1) = (\mathbf{A}_2\mathbf{A}_1,\ \mathbf{A}_2\mathbf{b}_1 + \mathbf{b}_2)$, so a parallel prefix scan computes every state of a sequence in logarithmic depth (Blelloch, 1990; Smith et al., 2023). A scan requires only that $\mathcal{F}$ be closed under composition, and the cost of each composition is set by the structure of $\mathbf{A}$.

2.2 State Tracking on Finite States

A finite-state automaton (FSA) is a tuple (Q, Σ, δ, q_0) with a finite state set Q, an alphabet Σ, a transition function $\delta\colon Q \times \Sigma \to Q$, and an initial state q_0. State tracking asks a model to output the state $q_t = \delta(q_{t-1}, \sigma_t)$ after every input symbol (Merrill et al., 2024). The word problem of a finite group G is the FSA with $Q = G$ and $\delta(h, g) = gh$, whose state is the running product of the inputs. An FSA is realized in a single layer if there are an injective encoding ι of Q into the state space, a transition in $\mathcal{F}$ for each symbol that maps $\iota(q)$ to $\iota(\delta(q, \sigma))$, and a readout that recovers q_t.

Krohn & Rhodes (1965) show that every FSA is simulated by a cascade of permutation-reset automata, in which each symbol acts on the states either by a permutation or by a constant map, called a reset. Stacked layers implement such a cascade one component at a time (Grazzi et al., 2025; Siems et al., 2025). A single layer instead applies one transition per symbol and must therefore realize each map $\delta(\cdot, \sigma)$ on Q directly. Every map on Q is a composition of permutations and merges, where a merge m sends one state a to another state b and fixes the rest, $m(a) = b$ and $m(q) = q$ for $q \neq a$. A merge removes the distinction between two states and preserves every other, and a reset is the composition of $|Q| - 1$ merges. As a transition matrix a merge is the identity with the column of a replaced by that of b, which Peng et al. (2025) and Siems et al. (2025) call a copy matrix.

Prior work has widened the automata that a single layer can realize by changing the transition structure. Diagonal plus rank-one transitions couple the coordinates of the state (Yang et al., 2024), and with negative eigenvalues a single generalized Householder matrix realizes a transposition of two states (Grazzi et al., 2025). Block-diagonal transitions reach the expressivity of dense transition matrices (Fan et al., 2024; Walker et al., 2025). DeltaProduct composes $n - 1$ generalized Householder

matrices per token and solves the S_n word problem in a single layer, keeping every transition at spectral norm at most one, which excludes merges (Siems et al., 2025). RWKV-7 realizes merges through an asymmetric rank-one update but gives up this bound, and its recurrence can diverge (Peng et al., 2025; Siems et al., 2025). PD-SSM multiplies a column-one-hot matrix by a complex diagonal matrix and emulates any FSA in a single layer with state dimension $|Q|$ (Terzić et al., 2025). Diagonal SSMs, DeltaProduct, and PD-SSM obtain stability by bounding the norm of every transition, PD-SSM by keeping its diagonal entries of modulus below one, which yields bounded-input, bounded-output stability (Orvieto et al., 2023; Siems et al., 2025; Terzić et al., 2025).

Beyond exact updates, recent work extends state tracking to partial observation and shows that a linear recurrence cannot carry the full belief over states but only its marginal (Siems et al., 2026).

2.3 SSMs on Continuous Sequence

SSMs also serve as backbones in vision (Zhu et al., 2024; Liu et al., 2024), as policy networks in robot learning (Jia et al., 2024; Yoo et al., 2026), and in inertial and visual odometry (Qin et al., 2026; Li et al., 2025). Continuous sequence prediction from raw sensor streams is central to control and estimation. CSP-Bench was built to measure this ability in sequence models and consists of six real-world labeled datasets, three tactile and three inertial, each mapping the stream of a single low-cost sensor to a sequence of physical quantities (Bhirangi et al., 2024). Tactile tasks predict the velocity of a robot end-effector from a ReSkin magnetic sensor on a gripper and the state of a joystick from XELA sensors on a robot hand. Three inertial tasks predict user velocity from IMU sensors of the smartphone in RoNIN (Herath et al., 2020), SLAM platform in VECtor (Gao et al., 2022), and body-worn devices in TotalCapture (Trumble et al., 2017).

HiSS is the model proposed with the benchmark and stacks two SSMs at different temporal resolutions. A low-level SSM turns fixed-size chunks of the input into local features and a high-level SSM maps the chunk features to the output. While flat SSMs outperform Transformers on the benchmark, HiSS improves over the best flat model on five of the six tasks. Yet all SSMs evaluated were diagonal.

2.4 Density Matrices and Quantum Channels

We briefly review the standard density-matrix, generalized Gell–Mann, and Kraus representations used in Section 3. A Qudit is a d-level quantum system whose states are density matrices

$$\mathcal{D}_d = \{\rho \in \mathbb{C}^{d\times d} \mid \rho = \rho^\dagger,\ \rho \succeq 0,\ \operatorname{Tr}\rho = 1\}, \tag{2}$$

forming a compact convex set of real dimension $d^2 - 1$ (Nielsen & Chuang, 2010). The generalized Gell–Mann matrices $\{\Lambda_i\}_{i=1}^{d^2-1}$ form an orthogonal basis of the traceless Hermitian matrices, satisfying $\Lambda_i^\dagger = \Lambda_i$, $\operatorname{Tr}(\Lambda_i) = 0$, and $\operatorname{Tr}(\Lambda_i\Lambda_j) = 2\delta_{ij}$ (Kimura, 2003; Bertlmann & Krammer, 2008). Together with the identity matrix, they span the Hermitian matrices. Every density matrix therefore has the expansion

$$\rho = \frac{I}{d} + \frac{1}{2}\sum_{i=1}^{d^2-1} x_i\Lambda_i, \qquad x_i = \operatorname{Tr}(\Lambda_i\rho), \tag{3}$$

where $\mathbf{x} \in \mathbb{R}^{d^2-1}$ is the generalized Bloch vector, providing a real-coordinate representation of ρ.

For $d = 2$, the Qudit is called a Qubit, and the generalized Gell–Mann matrices reduce to the Pauli matrices. This gives $\rho = \frac{1}{2}(I + \mathbf{x}\cdot\sigma)$ and $\mathcal{D}_2$ corresponds exactly to the unit ball in $\mathbb{R}^3$, known as the Bloch ball. For $d \geq 3$, the admissible Bloch vectors form a proper convex subset of a ball.

A quantum channel is a completely positive trace-preserving (CPTP) linear map Φ on $d \times d$ matrices that maps $\mathcal{D}_d$ into itself (Nielsen & Chuang, 2010). The Kraus theorem provides a constructive parameterization of quantum channels.

Theorem 1 (Kraus). *A linear map Φ is a channel if and only if $\Phi[\rho] = \sum_{\ell=1}^{r} K_\ell\,\rho\,K_\ell^\dagger$ for some $K_1, \ldots, K_r \in \mathbb{C}^{d\times d}$ with $\sum_\ell K_\ell^\dagger K_\ell = I$, and $r \leq d^2$ suffices (Kraus, 1983; Choi, 1975).*

The smallest such r is the Kraus rank of Φ. Any matrices that satisfy the completeness condition define a channel, so the converse direction of the theorem turns the Kraus form into a parameterization.

		Conservative		Dissipative		Stability	
Transition Structure	Example Model	Permutation	Rotation	Merge	Mixing	Non-expansive	State Invariance
Diagonal ($\mathbb{R}$, $\mathbb{C}$)	Mamba	×	×	×	✓	✓	✓
Block-diagonal	BD-SLiCE	✓	✓	✓	✓	×	×
Householder products	DeltaProduct	✓	✓	×	✓	✓	×
Diagonal plus rank-one	RWKV-7	×	×	✓	×	×	×
Column-one-hot × diagonal	PD-SSM	✓	×	✓	×	✓	✓
Quantum channel	CPTP-SSM (ours)	✓	✓	✓	✓	✓	✓

Table 1: **Transition structures along the conservative and dissipative axes with stability.** Entries refer to the transitions a single token can select. Rotation fails where the admissible groups are abelian or have real spectra, and mixing fails where the family is not convex. Structures that rotate and mix, such as block-diagonal and Householder products, lack state invariance, and the entries of PD-SSM hold in different regimes of its diagonal. Only quantum channels satisfy all six properties. Appendix A.7 gives the reason for each entry.

Density matrices and Kraus operators have appeared in sequence models as fixed symbol-indexed updates (Monras et al., 2011; Srinivasan et al., 2018; Adhikary et al., 2020), sequentially evaluated recurrent cells (Seo & Montiel, 2026), and output layers (Singh & Bhatia, 2026). We instead use input-selected channels as the transition family of a scan-based SSM.

3 METHOD

3.1 CONTINUOUS STATE TRACKING WITH UNCERTAINTY

We now extend state tracking to continuous groups under uncertainty. In view of conservative and dissipative transitions, we mathematically formulate the requirements that continuous states and uncertainty each place on the transition structure of a single layer, corresponding with those of finite state tracking.

Let B be a state set and $\mathcal{F}$ a family of affine transitions on B that contains the identity and is closed under composition. The continuous state extends conservative transitions from permutations of finite elements to rotations of a continuum of orientations.

Definition 1 (Permutation)**.** *$\mathcal{F}$ tracks a finite group G if injective homomorphism $\varphi\colon G \to \mathcal{F}$ exists.*

Definition 2 (Rotation)**.** *$\mathcal{F}$ tracks a connected non-abelian group G if injective continuous homomorphism $\varphi\colon G \to \mathcal{F}$ exists.*

Under uncertain observation an update is known only through a distribution, which requires a belief over states. Tracking the expected state mixes the admitted updates and contracts the distinctions between states according to the uncertainty, a selective dissipation like the merge transitions of finite-state tracking.

Definition 3 (Merge)**.** *Let $\iota\colon Q \to B$ be an injective encoding of a finite state set. $\mathcal{F}$ tracks merge on $\iota(Q)$ if for every pair of distinct states $a, b \in Q$ it contains a transition mapping $\iota(a)$ to $\iota(b)$ and every other $\iota(q)$ to itself.*

Definition 4 (Mixing)**.** *Let $T_\nu = \int T \, d\nu(T)$ for a probability measure ν on $\mathcal{F}$. $\mathcal{F}$ tracks mixing if $T_\nu \in \mathcal{F}$ for every such ν. For a state $\mathbf{x}$ and a random update $T \sim \nu$, the expected state $\mathbb{E}_\nu[T\mathbf{x}]$ then equals $T_\nu\mathbf{x}$.*

Existing transition structures meet these requirements in an instructive pattern. Built for FSA emulation, PD-SSM expresses both conservative and dissipative transitions in the finite case but neither in the extended case (Terzić et al., 2025). DeltaProduct cannot track merge because of its bound on the spectral norm (Siems et al., 2025), yet it tracks mixing, confirming that mixing is not an extension of merging. Table 1 gives a comparison of the transition structures.

Convexity required by mixing, together with rotation and the boundedness needed for stability, constrains the state set B. The smallest such B is characterized by the following lemma from standard facts on compact groups.

Lemma 1 (State Minimality). *Let $B \subset \mathbb{R}^k$ be a compact convex body admitting a connected non-abelian group of affine automorphisms. Then $k \geq 3$.*

At $k = 3$ the group is $SO(3)$ and B is a Euclidean ball up to a linear change of coordinates. In Pauli coordinates the density matrices of a qubit form the unit ball of $\mathbb{R}^3$, so the qubit realizes the smallest state that meets all three requirements. We therefore take density matrices as the state and channels as the transitions. While positive trace-preserving (PTP) maps also send density matrices to density matrices, we choose CPTP maps by reason of their Kraus representation, which yields a parameterization without projection or penalty.

Prior structures obtain stability mainly by bounding the norm of each transition, which constrains its linear part but not its offset. Recurrent models have been observed to fail at long lengths when their state distribution leaves the region visited during training (Buitrago & Gu, 2025), which motivates a stronger condition, a fixed state set that no transition can leave.

Definition 5 (State Invariance). *For a set U of inputs, let $\mathcal{F}_U$ be the transitions selected by inputs in U. The layer has state invariance if for every bounded U there is a compact set K with nonempty interior such that $T(K) \subseteq K$ for every $T \in \mathcal{F}_U$.*

Every channel maps $\mathcal{D}_d$ into itself, so CPTP transitions have state invariance with $K = \mathcal{D}_d$, the state set itself, even without a bound on the inputs.

3.2 CPTP-SSM

A CPTP-SSM layer consists of n_q parallel qudit heads. Each head $h = 1, \ldots, n_q$ carries a density-matrix state $\rho_t^{(h)} \in \mathcal{D}_d$, and the input u_t selects a CPTP transition $\Phi_t^{(h)} := \Phi^{(h)}(u_t)$:

$$\rho_t^{(h)} = \Phi_t^{(h)}[\rho_{t-1}^{(h)}] = \sum_{\ell=1}^{r} K_{t,\ell}^{(h)} \rho_{t-1}^{(h)} K_{t,\ell}^{(h)\dagger}. \tag{4}$$

Each head uses a learned affine projection of u_t to produce r unconstrained complex matrices $Z_{t,\ell}^{(h)} \in \mathbb{C}^{d\times d}$, represented by their real and imaginary parts. These matrices are jointly normalized as

$$S_t^{(h)} = \sum_{\ell=1}^{r} Z_{t,\ell}^{(h)\dagger} Z_{t,\ell}^{(h)}, \qquad K_{t,\ell}^{(h)} = Z_{t,\ell}^{(h)} \big(S_t^{(h)}\big)^{-1/2}. \tag{5}$$

For $S_t^{(h)} \succ 0$, this normalization gives $\sum_\ell K_{t,\ell}^{(h)\dagger} K_{t,\ell}^{(h)} = I$, so $\Phi_t^{(h)}$ is CPTP by construction. The network therefore predicts unconstrained matrices and obtains a valid input-selected transition without projection or penalty. Using the generalized Bloch coordinates of equation 3, the density-matrix state of each head and its real coordinates are related by

$$x_{t,i}^{(h)} = \mathrm{Tr}(\Lambda_i \rho_t^{(h)}), \qquad \rho_t^{(h)} = \frac{I}{d} + \frac{1}{2} \sum_{i=1}^{d^2-1} x_{t,i}^{(h)} \Lambda_i \tag{6}$$

Substituting this representation into the channel update yields the equivalent real affine recurrence $\mathbf{x}_t^{(h)} = \mathbf{A}_t^{(h)} \mathbf{x}_{t-1}^{(h)} + \mathbf{b}_t^{(h)}$, where

$$[\mathbf{A}_t^{(h)}]_{ij} = \frac{1}{2} \sum_{\ell=1}^{r} \mathrm{Tr}\Big(\Lambda_i K_{t,\ell}^{(h)} \Lambda_j K_{t,\ell}^{(h)\dagger}\Big), \qquad [\mathbf{b}_t^{(h)}]_i = \frac{1}{d} \sum_{\ell=1}^{r} \mathrm{Tr}\Big(\Lambda_i K_{t,\ell}^{(h)} K_{t,\ell}^{(h)\dagger}\Big). \tag{7}$$

Both $\mathbf{A}_t^{(h)}$ and $\mathbf{b}_t^{(h)}$ are real, so each input-selected CPTP channel induces an affine pair $(\mathbf{A}_t^{(h)}, \mathbf{b}_t^{(h)})$ on $\mathbb{R}^{d^2-1}$. Since the pair depends only on u_t, each head can be evaluated using the associative affine scan; the scan operates on affine pairs rather than Kraus operators, whose rank may grow under composition. The resulting head states are concatenated and mapped to the output through a readout function $\mathbf{y}_t = g_\theta(\mathbf{x}_t)$, with a total recurrent dimension of $N = n_q(d^2 - 1)$.

3.3 Expressivity and Stability

We now show that a single CPTP-SSM layer meets the requirements of Section 3.1. Throughout, $\mathcal{F}$ denotes the admissible transition family of a single head, the CPTP maps on $\mathcal{D}_d$ with Kraus rank at most r. Proofs are provided in Appendix A.

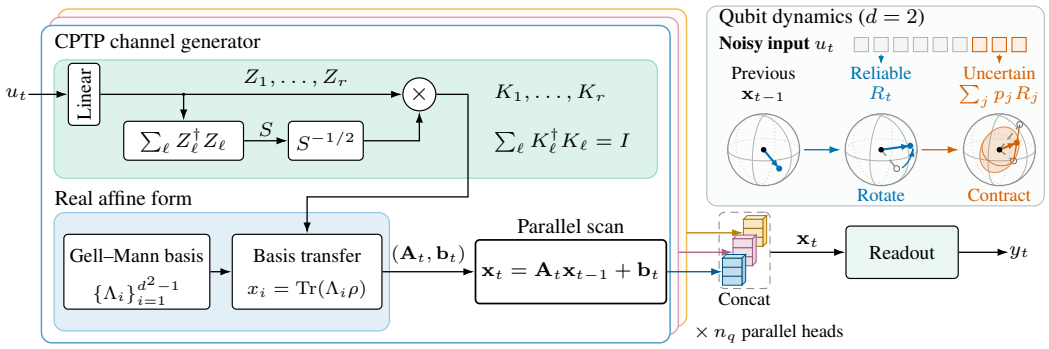


Figure 1: **CPTP-SSM layer.** Input-selected Kraus channels become real affine recurrences evaluated by a parallel scan. A reliable input rotates the qubit state while an uncertain input contracts it.

Proposition 1 (Permutation and merge)**.** *Let $d = |Q|$ and encode $q \in Q$ as $\rho_q = |q\rangle\langle q|$. For a finite group G, taking $Q = G$, the regular representation acts by permutation unitaries, so $\mathcal{F}$ tracks G at $r = 1$. For any Q, $\mathcal{F}$ tracks merge on the encoded states at $r = 2$. At $r = d$, one head with one-hot inputs exactly emulates any deterministic finite automaton on Q, with the state read out affinely from the Bloch coordinates.*

Proposition 2 (Rotation and mixing)**.** *Let G be a connected non-abelian group that admits a continuous injective homomorphism into $PU(d)$. Then $\mathcal{F}$ tracks G at $r = 1$. At $r = d^2$, $\mathcal{F}$ is closed under mixing, and a mixture of m unitary channels has a Kraus representation with at most $\min(m, d^2)$ operators. In particular, a qubit head tracks $SO(3) \cong PU(2)$, and $r = 4$ covers every mixture of rotations.*

The regular representation is not always dimension-efficient. One qubit head already encodes $A_5 \subset SO(3)$ exactly, whereas S_5 requires $d > 2$, and $d = 4$ suffices (Appendix A.4).

Proposition 3 (Stability)**.** *Every $\Phi \in \mathcal{F}$ maps $\mathcal{D}_d$ into itself, so the head has state invariance with $K = \mathcal{D}_d$. Moreover, $\|\Phi[\rho] - \Phi[\rho']\|_1 \leq \|\rho - \rho'\|_1$ for the trace norm $\|\cdot\|_1$, with equality if Φ is unitary. For $d = 2$, $\|\rho - \rho'\|_1$ equals the Euclidean distance between the Bloch vectors, so $\|\mathbf{A}_t\|_2 \leq 1$ and $\|\partial \mathbf{x}_T / \partial \mathbf{x}_t\|_2 \leq 1$ at every length.*

4 Experiments

4.1 Synthetic state tracking

We first verify the stability and expressivity of CPTP-SSM on synthetic state-tracking tasks, which also confirm the roles of the two parameters: the qudit dimension d decides what the state can hold, and the Kraus rank r determines what a transition may mix.

Length generalization and state geometry on A_5. The A_5 word problem asks a model to track the running product of even permutations on five elements (Merrill et al., 2024). We follow the length-generalization experiment of Walker et al. (2025), with matching criteria and protocol of training on lengths 3–40 and selecting checkpoints on validation lengths 40–128. CPTP-SSM uses one layer of 32 qubit heads at $r = 4$. While Walker et al. (2025) reports extrapolation results up to a length of 5120, Figure 2(a) extends evaluation to a length of 327,680, 8,192 times the training length. CPTP-SSM maintains perfect accuracy over 5,120 and has the highest mean accuracy at the longest evaluated length. The full curve also shows its eventual degradation, rather than implying unlimited length generalization. Note PD-SSM with $K = 60$ does not learn this full-group task; the experiments of Terzić et al. (2025) use a restricted set of input permutations.

One possible explanation for the length-generalization results comes from state invariance, which keeps each head within a fixed compact state space regardless of sequence length. Although this property does not guarantee that evaluation states overlap with those encountered during training, Figure 2(b–d) provides empirical observation of such overlap in the learned model. In the PCA

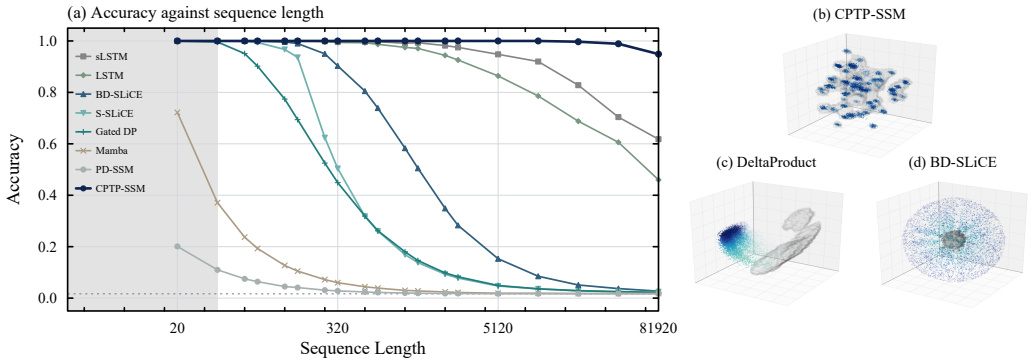


Figure 2: **Length generalization and state geometry on A_5.** (a) Per-token test accuracy, mean over three seeds. (b–d) Last layer states projected onto each model's top three principal components, fitted jointly on hidden states from training and testing. Grey volumes show the 95% density regions of 3,000 states sampled at training length, colored points are hidden states at test lengths 20–1,280 indicating each horizon by color.

projections of recurrent states, CPTP-SSM remains within the training density region across the sampled lengths. DeltaProduct exhibits directional drift away from the region despite a nearly unchanged state norm, whereas BD-SLiCE exhibits pronounced norm growth. These observations suggest that the invariant state geometry may help CPTP-SSM maintain familiar state representations over longer sequences, consistent with the unexplored-states hypothesis of Buitrago & Gu (2025).

Necessity of a higher qudit dimension. We next test the role of d on the S_5 word problem, using the same protocol with dropout disabled. Table 2 reports the best validation accuracy on lengths 40–128, averaged over three seeds. Qubit heads at their full Kraus budget $r = 4$ reach only 0.083, whereas $d = 4$ heads solve the task even with $r = 1$ and 96k parameters. Increasing the available dissipation does not compensate for the qubit restriction in this comparison: the higher-dimensional reversible action is sufficient without additional Kraus operators.

Table 2: S_5 **tracking.** Best validation accuracy on lengths 40–128, mean over three seeds.

(d, r)	Heads	Params	Acc.
$(2, 4)$	32	192k	0.083
$(4, 1)$	8	96k	**1.000**
$(4, 6)$	8	261k	**1.000**

Table 3: $SO(3)$ **tracking.** Mean R^2 over positions 40–128, mean and standard deviation over three seeds.

Model	Haar	Haar + noise	Small incr. + noise
CPTP-SSM$_{d=2,r=1}$	1.00 ± 0.00	0.618 ± 0.0300	0.6729 ± 0.0949
CPTP-SSM$_{d=2,r=1}$, scalar gated	–	0.8208 ± 0.0416	0.9391 ± 0.0029
CPTP-SSM$_{d=2,r=4}$	1.0000 ± 0.00	$\mathbf{0.9927 \pm 0.0012}$	$\mathbf{0.9590 \pm 0.0021}$
Gated DeltaProduct$_{n_h=4}$	0.9247 ± 0.0146	0.8361 ± 0.0635	0.8480 ± 0.0389

Continuous group tracking with and without observation noise. We extend product tracking in the finite-group word problems to continuous rotations in $SO(3)$, first with exact observations and then with observation noise. We evaluate Haar-distributed increments without noise, Haar increments and restricted increments with observation noise. In the noisy settings, the target is the exact posterior mean of the accumulated rotation. All models train on lengths 3 to 40 for 100,000 steps and are evaluated using R^2 against the posterior mean, over positions 40 to 128 summarized across three seeds. We additionally evaluate a scalar-gated CPTP-SSM, which multiplies each unitary $r = 1$ head's transition by an input-dependent scalar gate. Detailed task generation, posterior references, and configurations are given in Appendix C.6.

Without observation noise, both $r = 1$ and $r = 4$ accurately track rotation composition. Under noise, an input-dependent scalar gate substantially improves the rank-one model, but remains below $r = 4$ on both tasks. This progression distinguishes reversible-only heads, scalar-gated rotations, and general CPTP transitions: allowing contraction improves noisy tracking, and the general channel family improves further beyond scalar contraction. CPTP-SSM at $r = 4$ obtains the highest mean R^2 on all three tasks, including under the comparison with Gated DeltaProduct.

Finite-state automaton emulation. We evaluate on the four automaton tasks of Delétang et al. (2023), following the protocol training on lengths 3–40 and reporting best validation accuracy on

lengths 40–256 over five seeds. CPTP-SSM uses two layers of 42 qubit heads for a state dimension of 126 per layer against the two layers of state dimension 128 of PD-SSM (Terzić et al., 2025). CPTP-SSM with Kraus rank 4 solves all four tasks and attains $99.9 \pm 0.1\%$ average accuracy. The key distinction appears in Modular Arithmetic, which contains a non-invertible transition. Multiplication by zero resets every residue to zero, so a dissipative update is required. Rank-one model solves the other three tasks and fails Modular Arithmetic at $26.3 \pm 0.4\%$. This again identifies the Kraus rank as the parameter that admits dissipation.

Table 4: **Automaton tasks.** Mean and standard deviation over five seeds of the best validation accuracy (%) on sequences of length 40 to 256. Baseline results are taken from Terzić et al. (2025).

Model	Cycle Nav.	Even Pairs	Mod Arith.	Parity	Average
LSTM	100.0 ± 0.0	100.0 ± 0.0	99.9 ± 0.1	100.0 ± 0.0	100.0 ± 0.0
Transformer	24.4 ± 0.5	90.4 ± 10.4	23.6 ± 0.7	52.2 ± 0.4	47.7 ± 2.6
Mamba	48.4 ± 2.2	**100.0 ± 0.0**	33.1 ± 6.6	54.2 ± 2.1	58.9 ± 1.8
Gated DP	46.3 ± 6.6	**100.0 ± 0.0**	78.4 ± 10.9	98.0 ± 1.4	80.7 ± 3.2
RWKV-7	37.8 ± 5.0	88.1 ± 14.2	39.5 ± 6.1	51.1 ± 0.3	54.1 ± 4.1
BD-SLiCE	99.8 ± 0.2	85.9 ± 11.3	54.0 ± 12.5	95.3 ± 3.9	83.8 ± 4.3
D-DE-SLiCE	73.3 ± 29.4	84.8 ± 8.5	98.4 ± 0.7	83.8 ± 11.3	85.1 ± 8.2
PD-SSM	99.5 ± 0.7	99.7 ± 0.3	96.2 ± 3.4	99.9 ± 0.1	98.8 ± 0.9
CPTP-SSM$_{d=2,\, r=1}$	**100.0 ± 0.0**	99.5 ± 0.7	26.3 ± 0.4	**100.0 ± 0.0**	81.4 ± 0.3
CPTP-SSM$_{d=2,\, r=4}$	**100.0 ± 0.0**	**100.0 ± 0.1**	**99.7 ± 0.5**	**100.0 ± 0.0**	**99.9 ± 0.1**
Random	20.0	50.0	20.0	50.0	35.0

4.2 Real-world sequence modeling

We next evaluate CPTP-SSM on two complementary aspects of real-world sequence modeling: classification of long multivariate time series and prediction from physical sensor streams. The former tests the model on sequences of up to 17,984 steps; the latter evaluates its applicability to tactile and inertial observations of physical processes.

Long multivariate time-series classification We use the six long timeseries datasets from University of East Anglia (UEA) Multivariate Time-Series Classification Archive (UEA-MTSCA)(Bagnall et al., 2018) selected by Walker et al. (2024). We follow the splits and hyperparameter grid of Terzić et al. (2025), report test accuracy over five initializations. For state dimension, we used $n_q \in \{11, 43, 85\}$, matching $3n_q$ real recurrent coordinates to the storage of complex states of size $\{16, 64, 128\}$. CPTP-SSM with qubit heads achieves the highest mean accuracy on three of the six datasets(Table 5), including the longest dataset EigenWorms attaining 96.1% accuracy. These results demonstrate competitive long-sequence classification, extending the evaluation from synthetic state tracking to long real-world observations. Full configurations are given in Appendix C.5.

Table 5: **Long-sequence classification.** Test accuracy (%) on the six long UEA datasets, mean and standard deviation over five random initializations. Baseline results are taken from Terzić et al. (2025). The best result in each column is in bold.

Seq. length	Worms 17,984	SCP1 896	SCP2 1,152	Ethanol 1,751	Heartbeat 405	Motor 3,000	Avg ↑
NRDE	83.9 ± 7.3	80.9 ± 2.5	53.7 ± 6.9	25.3 ± 1.8	72.9 ± 4.8	47.0 ± 5.7	60.6
NCDE	75.0 ± 3.9	79.8 ± 5.6	53.0 ± 2.8	29.9 ± 6.5	73.9 ± 2.6	49.5 ± 2.8	60.2
Log-NCDE	85.6 ± 5.1	83.1 ± 2.8	53.7 ± 4.1	34.4 ± 6.4	75.2 ± 4.6	53.7 ± 5.3	64.3
LRU	87.8 ± 2.8	82.6 ± 3.4	51.2 ± 3.6	21.5 ± 2.1	78.4 ± 6.7	48.4 ± 5.0	61.7
S5	81.1 ± 3.7	**89.9 ± 4.6**	50.5 ± 2.6	24.1 ± 4.3	77.7 ± 5.5	47.7 ± 5.5	61.8
S6	85.0 ± 16.1	82.8 ± 2.7	49.9 ± 9.4	26.4 ± 6.4	76.5 ± 8.3	51.3 ± 4.7	62.0
Mamba	70.9 ± 15.8	80.7 ± 1.4	48.2 ± 3.9	27.9 ± 4.5	76.2 ± 3.8	47.7 ± 4.5	58.6
LinOSS-IMEX	80.0 ± 2.7	87.5 ± 4.0	**58.9 ± 8.1**	29.9 ± 1.0	75.5 ± 4.3	57.9 ± 5.3	65.0
LinOSS-IM	95.0 ± 4.4	87.8 ± 2.6	58.2 ± 6.9	29.9 ± 0.6	75.8 ± 3.7	**60.0 ± 7.5**	67.8
PD-SSM	90.0 ± 5.7	80.9 ± 2.0	56.1 ± 8.6	34.7 ± 4.0	80.0 ± 2.6	**60.0 ± 3.7**	67.0
CPTP-SSM$_{d=2,\, r=4}$	**96.1 ± 2.8**	86.4 ± 4.1	52.3 ± 3.4	**37.5 ± 5.6**	**81.3 ± 3.9**	52.3 ± 6.6	67.6

Physical sensor-stream prediction. CSP-Bench combines tactile prediction (MarkerWriting, IntrinsicSlip, JoystickControl) and inertial prediction (RoNIN, VECtor, TotalCapture), with 50 Hz inputs and 5 Hz targets (Bhirangi et al., 2024; Herath et al., 2020; Gao et al., 2022; Trumble et al., 2017). Unlike the synthetic tasks, these benchmarks require prediction from sensor measurements rather than directly specified state transitions. Their noisy, indirect observations motivate recurrent updates that can both retain information and revise it as new measurements arrive. In addition to the flat baselines and HiSS, we compare CPTP-SSM with recent SSMs, namely Mamba-3 (Lahoti et al., 2026) and Gated DeltaProduct. Table 6 reports RMSE over three seeds on the six tasks. Selection grids, normalization changes, parameter counts, and standard deviations are in Appendix C.2.

Table 6: **Sensor prediction.** CSP-Bench RMSE, mean over three seeds; lower is better. MW: MarkerWriting; IS: IntrinsicSlip; JC: JoystickControl; R: RoNIN; V: VECtor; TC: TotalCapture. Bold marks the best mean. Standard deviations: Table 13.

	Tactile				Inertial		
	MW	IS		JC	R	V	TC
	(cm/s)	Trans. (cm/s)	Rot. (deg/s)	(–)	(m/s)	(m/s)	(m/s)
Transformer	1.572	1.678	11.14	0.322	0.314	0.198	0.642
LSTM	1.083	1.343	8.09	0.339	0.189	0.144	**0.454**
S4	1.143	1.286	8.10	**0.320**	0.196	0.165	0.556
Mamba-3	0.993	1.214	7.00	0.332	0.188	0.145	0.628
Gated DeltaProduct$_{n_h=2}$	1.197	1.359	8.25	0.328	0.185	0.150	0.616
Gated DeltaProduct$_{n_h=4}$	1.169	1.317	8.15	0.328	0.187	0.153	0.624
HiSS	1.048	1.194	7.16	0.325	0.181	0.150	0.525
CPTP-SSM$_{d=2,\, r=4}$	**0.980**	**1.158**	**6.89**	0.329	**0.171**	**0.141**	0.530

CPTP-SSM achieves the lowest mean RMSE on four of six tasks and also outperforms HiSS without an explicit temporal hierarchy. HiSS retains lower error on JoystickControl and TotalCapture. This comparison highlights recurrent-transition expressivity as an important axis of performance improvement alongside temporal hierarchy. Rather than organizing the sequence into multiple temporal scales, CPTP-SSM enriches the family of state updates within each recurrent layer. Its performance on tactile and inertial streams supports the applicability of this approach to physical sensor prediction.

5 Conclusion

We expanded state tracking from FSA to continuous groups under uncertainty and formulated the requirements mathematically. Taking the alternative route of constraining the state space, we introduced CPTP-SSM, whose density-matrix states evolve under input-selected CPTP transitions computed by a parallel scan. We proved its expressivity and stability and confirmed them empirically, with strong length generalization consistent with state invariance. Its results on long time-series classification and continuous sensor prediction highlight expressive recurrent transitions as an alternative route to improved physical sequence modeling.

Limitations and future work. Heads with $d \geq 3$ remain costly, since composing their transitions grows as d^6, so our experiments widen layers through qubit heads, and efficient kernels for higher-dimensional heads are the next step. CPTP-SSM carries the expected state rather than the full posterior, and its accuracy still declines at the longest evaluated lengths. Because one layer covers both finite and continuous state tracking, integrating an optimized CPTP-SSM into larger models such as vision-language models is a promising direction.

Reproducibility statement

We provide a link to access the source code `https://anonymous.4open.science/r/cptp-ssm/`. For all theoretical results, we clearly state the assumptions and provide complete proofs in the appendix.

AI use statement

Generative AI tools assisted with manuscript editing, LaTeX formatting, vector-figure design, and consistency checks. We have reviewed all AI-assisted work and take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

Ethics statement

This work proposes a sequence-model architecture and evaluates it on synthetic tasks and on public benchmarks released by their authors. It involves no new data collection.

References

Sandesh Adhikary, Siddarth Srinivasan, Geoff Gordon, and Byron Boots. Expressiveness and learning of hidden quantum Markov models. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2020.

Martin Arjovsky, Amar Shah, and Yoshua Bengio. Unitary evolution recurrent neural networks. In *International Conference on Machine Learning (ICML)*, 2016.

Anthony Bagnall, Hoang Anh Dau, Jason Lines, Michael Flynn, James Large, Aaron Bostrom, Paul Southam, and Eamonn Keogh. The UEA multivariate time series classification archive, 2018. *arXiv preprint arXiv:1811.00075*, 2018.

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 *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Reinhold A. Bertlmann and Philipp Krammer. Bloch vectors for qudits. *Journal of Physics A: Mathematical and Theoretical*, 41(23):235303, 2008.

Raunaq Bhirangi, Chenyu Wang, Venkatesh Pattabiraman, Carmel Majidi, Abhinav Gupta, Tess Hellebrekers, and Lerrel Pinto. Hierarchical state space models for continuous sequence-to-sequence modeling. In *International Conference on Machine Learning (ICML)*, 2024.

Guy E. Blelloch. Prefix sums and their applications. Technical Report CMU-CS-90-190, School of Computer Science, Carnegie Mellon University, 1990.

Theodor Bröcker and Tammo tom Dieck. *Representations of Compact Lie Groups*. Springer, 1985.

Ricardo Buitrago and Albert Gu. Understanding and improving length generalization in recurrent models. In *International Conference on Machine Learning (ICML)*, pp. 5856–5874, 2025.

Justin T. Chiu and Alexander M. Rush. Scaling hidden Markov language models. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

Man-Duen Choi. Completely positive linear maps on complex matrices. *Linear Algebra and its Applications*, 10(3):285–290, 1975.

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

Grégoire Delétang, Anian Ruoss, Jordi Grau-Moya, Tim Genewein, Li Kevin Wenliang, Elliot Catt, Chris Cundy, Marcus Hutter, Shane Legg, Joel Veness, and Pedro A. Ortega. Neural networks and the Chomsky hierarchy. In *International Conference on Learning Representations (ICLR)*, 2023.

Ting-Han Fan, Ta-Chung Chi, and Alexander I. Rudnicky. Advancing regular language reasoning in linear recurrent neural networks. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pp. 45–53, 2024.

Christian Forster, Luca Carlone, Frank Dellaert, and Davide Scaramuzza. On-manifold preintegration for real-time visual-inertial odometry. *IEEE Transactions on Robotics*, 33(1):1–21, 2017.

Ling Gao, Yuxuan Liang, Jiaqi Yang, Shaoxun Wu, Chenyu Wang, Jiaben Chen, and Laurent Kneip. VECtor: A versatile event-centric benchmark for multi-sensor SLAM. *IEEE Robotics and Automation Letters*, 7(3):8217–8224, 2022.

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

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

Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. In *International Conference on Learning Representations (ICLR)*, 2022.

Sachini Herath, Hang Yan, and Yasutaka Furukawa. RoNIN: Robust neural inertial navigation in the wild: Benchmark, evaluations, & new methods. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3146–3152, 2020.

Nicholas J. Higham. *Functions of Matrices: Theory and Computation*. SIAM, 2008.

Royina Karegoudra Jayanth, Yinshuang Xu, Ziyun Wang, Evangelos Chatzipantazis, Kostas Daniilidis, and Daniel Gehrig. EqNIO: Subequivariant neural inertial odometry. In *International Conference on Learning Representations (ICLR)*, 2025.

Xiaogang Jia, Qian Wang, Atalay Donat, Bowen Xing, Ge Li, Hongyi Zhou, Onur Celik, Denis Blessing, Rudolf Lioutikov, and Gerhard Neumann. MaIL: Improving imitation learning with selective state space models. In *Conference on Robot Learning (CoRL)*, 2024.

Gen Kimura. The Bloch vector for N-level systems. *Physics Letters A*, 314(5–6):339–349, 2003.

Karl Kraus. *States, Effects, and Operations: Fundamental Notions of Quantum Theory*, volume 190 of *Lecture Notes in Physics*. Springer, 1983.

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

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

Mario Lezcano-Casado and David Martínez-Rubio. Cheap orthogonal constraints in neural networks: A simple parametrization of the orthogonal and unitary group. In *International Conference on Machine Learning (ICML)*, 2019.

Yan Li, Yang Xu, Changhao Chen, Zhongchen Shi, Wei Chen, Liang Xie, Hongbo Chen, and Erwei Yin. M2EIT: Multi-domain mixture of experts for robust neural inertial tracking. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 28207–28216, 2025.

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

Wenxin Liu, David Caruso, Eddy Ilg, Jing Dong, Anastasios I. Mourikis, Kostas Daniilidis, Vijay Kumar, and Jakob Engel. TLIO: Tight learned inertial odometry. *IEEE Robotics and Automation Letters*, 5(4):5653–5660, 2020.

Yue Liu, Yunjie Tian, Yuzhong Zhao, Hongtian Yu, Lingxi Xie, Yaowei Wang, Qixiang Ye, Jianbin Jiao, and Yunfan Liu. VMamba: Visual state space model. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

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

Jacob Miller, Guillaume Rabusseau, and John Terilla. Tensor networks for probabilistic sequence modeling. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2021.

Alex Monras, Almut Beige, and Karoline Wiesner. Hidden quantum Markov models and non-adaptive read-out of many-body states. *Applied Mathematical and Computational Sciences*, 3:93–122, 2011.

Sajad Movahedi, Felix Sarnthein, Nicola Muca Cirone, and Antonio Orvieto. Fixed-point RNNs: Interpolating from diagonal to dense. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

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

Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 10th anniversary edition, 2010.

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

Bo Peng, Ruichong Zhang, Daniel Goldstein, Eric Alcaide, Xingjian Du, Haowen Hou, Jiaju Lin, Jiaxing Liu, Janna Lu, William Merrill, Guangyu Song, Kaifeng Tan, Saiteja Utpala, Nathan Wilce, Johan S. Wind, Tianyi Wu, Daniel Wuttke, and Christian Zhou-Zheng. RWKV-7 "goose" with expressive dynamic state evolution. *arXiv preprint arXiv:2503.14456*, 2025.

Wenyuan Qin, Zhi Lin, Lu Bai, Shaojun Yao, Hao Xu, and Xiwang Dong. SS-VIO: A novel cross modal state space visual inertial odometry. *IEEE Robotics and Automation Letters*, 11(2): 2010–2017, 2026. doi: 10.1109/LRA.2025.3643327.

Michael O. Rabin. Probabilistic automata. *Information and Control*, 6(3):230–245, 1963.

Lawrence R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

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

Joonwon Seo and Mariana Montiel. Density matrix RNN (DM-RNN): A quantum information theoretic framework for modeling musical context and polyphony. In *Mathematics and Computation in Music (MCM)*, Lecture Notes in Computer Science, pp. 397–411. Springer, 2026. doi: 10.1007/978-3-032-27827-2_24.

Mehran Shakerinava, Behnoush Khavari, Siamak Ravanbakhsh, and Sarath Chandar. The expressive limits of diagonal SSMs for state-tracking. In *International Conference on Learning Representations (ICLR)*, 2026.

Julien Siems, Timur Carstensen, Arber Zela, Frank Hutter, Massimiliano Pontil, and Riccardo Grazzi. DeltaProduct: Improving state-tracking in linear RNNs via Householder products. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Julien Siems, Riccardo Grazzi, Korbinian Pöppel, Kirill Kalinin, Hitesh Ballani, and Babak Rahmani. Learning state-tracking from code using linear RNNs. *arXiv preprint arXiv:2602.14814*, 2026.

Priyanshi Singh and Krishna Bhatia. Kraus constrained sequence learning for quantum trajectories from continuous measurement. *arXiv preprint arXiv:2603.05468*, 2026.

Jimmy T. H. Smith, Andrew Warrington, and Scott W. Linderman. Simplified state space layers for sequence modeling. In *International Conference on Learning Representations (ICLR)*, 2023.

Siddarth Srinivasan, Geoff Gordon, and Byron Boots. Learning hidden quantum Markov models. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2018.

Aleksandar Terzić, Nicolas Menet, Michael Hersche, Thomas Hofmann, and Abbas Rahimi. Structured sparse transition matrices to enable state tracking in state-space models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

David H. Titterton and John L. Weston. *Strapdown Inertial Navigation Technology*. The Institution of Engineering and Technology, 2nd edition, 2004.

Matthew Trumble, Andrew Gilbert, Charles Malleson, Adrian Hilton, and John Collomosse. Total capture: 3D human pose estimation fusing video and inertial sensors. In *British Machine Vision Conference (BMVC)*, 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. In *International Conference on Machine Learning (ICML)*, 2024.

Benjamin Walker, Lingyi Yang, Nicola Muca Cirone, Cristopher Salvi, and Terry Lyons. Structured linear CDEs: Maximally expressive and parallel-in-time sequence models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

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

Youngju Yoo, Jiaheng Hu, Yifeng Zhu, Bo Liu, Qiang Liu, Roberto Martín-Martín, and Peter Stone. RoboSSM: Scalable in-context imitation learning via state-space models. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2026.

Lianghui Zhu, Bencheng Liao, Qian Zhang, Xinlong Wang, Wenyu Liu, and Xinggang Wang. Vision Mamba: Efficient visual representation learning with bidirectional state space model. In *International Conference on Machine Learning (ICML)*, 2024.

A Proofs and Additional Results

All statements concern exact channels in exact arithmetic. Numerical inverse-square-root approximations are treated in Appendix B.2. We write $\mathcal{C}_d$ for the channels on $\mathcal{D}_d$ and $\mathcal{C}_{d,r}$ for those of Kraus rank at most r, so that $\mathcal{F} = \mathcal{C}_{d,r}$.

A.1 Realizability and Reversible Transitions

Every $\Phi \in \mathcal{C}_{d,r}$ is attained by the normalization in equation 5. Choose Kraus operators $K_1, \dots, K_s$ with $s \le r$, set $Z_\ell = K_\ell$ for $\ell \le s$, and pad the remaining slots with zeros. Then

$$S = \sum_{\ell=1}^{r} Z_\ell^\dagger Z_\ell = I, \qquad Z_\ell S^{-1/2} = K_\ell. \tag{8}$$

This holds pointwise. It does not mean that one affine input projection realizes every map from continuous inputs to channels. For one-hot inputs it does, as shown in the proof of Proposition 1.

Proposition 4 (Reversible transitions)**.** *A channel on $\mathcal{D}_d$ has a CPTP inverse if and only if it is a unitary conjugation $\mathrm{Ad}_U[\rho] = U\rho U^\dagger$. The reversible channels form the group $\{\mathrm{Ad}_U : U \in \mathrm{U}(d)\} \cong PU(d)$, and each has Kraus rank one.*

Proof. A unitary channel has the CPTP inverse $\mathrm{Ad}_{U^\dagger}$ and is attained with $Z_1 = U$ and all other slots zero. Conversely, let Φ and its inverse Ψ have Kraus operators K_ℓ and L_j. The identity channel $\Psi \circ \Phi$ has Kraus operators $L_j K_\ell$. For a pure input, every nonzero term of a Kraus sum of the identity lies on the input ray, and a linear operator preserving every ray is scalar, so

$$L_j K_\ell = c_{j\ell} I. \tag{9}$$

Some $c_{j_0\ell_0}$ is nonzero, so L_{j_0} is invertible and $K_\ell = c_{j_0\ell} M$ with $M = L_{j_0}^{-1}$. Completeness gives $aM^\dagger M = I$ for $a = \sum_\ell |c_{j_0\ell}|^2 > 0$, so $U = \sqrt{a}\, M$ is unitary and $\Phi = \mathrm{Ad}_U$. Two unitaries define the same conjugation exactly when they differ by a phase, which gives $PU(d) = \mathrm{U}(d)/\mathrm{U}(1)$. □

A.2 Proof of Proposition 1

Proof of Proposition 1. Identify Q with the standard basis of $\mathbb{C}^d$, $d = |Q|$.

Permutation. Take $Q = G$ and $P_g = \sum_{h \in G} |gh\rangle\langle h|$. The map $g \mapsto P_g$ is the regular representation, an injective homomorphism into $\mathrm{U}(d)$. For $g \neq e$, P_g has zero diagonal, so it is not a scalar, and $P_g^{-1} P_{g'} = P_{g^{-1}g'}$ is a scalar only if $g = g'$. Hence $g \mapsto \mathrm{Ad}_{P_g}$ is an injective homomorphism into $\mathcal{F}$, attained at $r = 1$ by equation 8, and Definition 1 holds. On the encoded states it acts as $\mathrm{Ad}_{P_g}[\rho_h] = \rho_{gh}$.

Merge. Fix $a \neq b$. The operators $K_1 = |b\rangle\langle a|$ and $K_2 = I - |a\rangle\langle a|$ satisfy $K_1^\dagger K_1 + K_2^\dagger K_2 = |a\rangle\langle a| + (I - |a\rangle\langle a|) = I$. On ρ_a only K_1 contributes and gives ρ_b, and on ρ_q with $q \neq a$ only K_2 contributes and fixes it. The encoding $q \mapsto \rho_q$ is injective, so Definition 3 holds at $r = 2$.

Automata. For a symbol σ, the d operators $K_q^\sigma = |\delta(q, \sigma)\rangle\langle q|$ satisfy

$$\sum_{q \in Q} (K_q^\sigma)^\dagger K_q^\sigma = \sum_{q \in Q} |q\rangle\langle q| = I, \tag{10}$$

and on an encoded state only the term indexed by q survives,

$$\Phi_\sigma[\rho_q] = |\delta(q, \sigma)\rangle\langle \delta(q, \sigma)|. \tag{11}$$

Starting at ρ_{q_0}, induction on the input length gives $\rho_t = \rho_{q_t}$ for the automaton state q_t. The affine selector realizes this choice. For $u_t = \mathbf{e}_\sigma$, column σ of the input projection holds the real and imaginary parts of $\{K_q^\sigma\}_q$ and the bias is zero, so the raw matrices are the Kraus operators, and equation 10 gives $S_t = I$. Finally,

$$[\rho_t]_{qq} = \frac{1}{d} + \frac{1}{2} \sum_{i=1}^{d^2-1} x_{t,i} [\Lambda_i]_{qq} \tag{12}$$

is affine in the Bloch coordinates and equals the one-hot vector of q_t. Summing it over the accepting states gives an exact acceptance indicator. □

The automaton channels discard off-diagonal entries even when $\delta(\cdot, \sigma)$ is a permutation, so they preserve distinctions only among the encoded states. A permutation π of Q can instead use the unitary $U_\pi = \sum_q |\pi(q)\rangle\langle q|$, which is reversible on all of $\mathcal{D}_d$. The budget $r = d$ accommodates both, and it is sufficient, not minimal.

A.3 Proof of Proposition 2

Proof of Proposition 2. Rotation. Let $\iota \colon G \to PU(d)$ be a continuous injective homomorphism, choose a representative U_g of each $\iota(g)$, and set $\varphi(g) = \mathrm{Ad}_{U_g}$. Conjugation does not depend on the phase of U_g, so φ is well defined, continuous, and injective, and

$$\varphi(g) \circ \varphi(h) = \varphi(gh), \qquad \varphi(e) = \mathrm{id}\,. \tag{13}$$

Each $\varphi(g)$ has one Kraus operator, so φ maps into $\mathcal{F}$ at $r = 1$ and Definition 2 holds. No continuous choice of representatives is needed.

Mixing. For a probability measure ν on $\mathcal{C}_d$, let

$$\Phi_\nu[\rho] = \int_{\mathcal{C}_d} \Phi[\rho]\, d\nu(\Phi). \tag{14}$$

Integration preserves complete positivity and trace preservation, so $\Phi_\nu \in \mathcal{C}_d$. By the Kraus theorem $\mathcal{C}_{d,d^2} = \mathcal{C}_d$, so at $r = d^2$ the family $\mathcal{F}$ is closed under mixing (Definition 4) with $T_\nu = \Phi_\nu$. The affine coefficients in equation 7 are linear in the channel, so the same holds in Bloch coordinates. For a mixture of m unitary channels Ad_{U_j} with weights p_j, the operators $K_j = \sqrt{p_j}\, U_j$ satisfy $\sum_j K_j^\dagger K_j = \sum_j p_j I = I$, so m operators suffice, and the Kraus theorem bounds the rank by d^2 as well.

Qubits. For $d = 2$, unitary conjugation rotates the Bloch vector,

$$\mathrm{Ad}_U[\rho(\mathbf{x})] = \rho(R(U)\mathbf{x}), \qquad R(UV) = R(U)R(V), \tag{15}$$

and $SU(2) \to SO(3)$ is onto with kernel $\{\pm I\}$, so $PU(2) \cong SO(3)$ (Nielsen & Chuang, 2010). Every mixture of rotations then has Kraus rank at most $d^2 = 4$. □

Expected state. Definition 4 gives $\mathbb{E}_\nu[T\mathbf{x}] = T_\nu \mathbf{x}$ for one step. If $\rho_t = \Phi_t[\rho_{t-1}]$ with $\Phi_t \sim \nu_t$ independent of ρ_{t-1}, conditioning and linearity give

$$\bar{\rho}_t = \mathbb{E}\big[\Phi_{\nu_t}[\rho_{t-1}]\big] = \Phi_{\nu_t}[\bar{\rho}_{t-1}]. \tag{16}$$

This propagates the expected state, not a full posterior, and it needs the independence.

Selective dissipation. Let U induce $R \in SO(3)$ and $p = (1+\lambda)/2$ with $\lambda \in [0,1]$. The operators

$$K_1 = \sqrt{p}\, U, \qquad K_2 = \sqrt{1-p}\, U\sigma_z \tag{17}$$

satisfy completeness. Conjugation by σ_z maps (x_1, x_2, x_3) to $(-x_1, -x_2, x_3)$, so the channel acts as $\mathbf{x} \mapsto R\,\mathrm{diag}(\lambda, \lambda, 1)\mathbf{x}$. It preserves distinctions along the z-axis, maps them to the direction $R\mathbf{e}_3$, and contracts those in the orthogonal plane by λ. The same map is the equal mixture of $RR_z(\theta)$ and $RR_z(-\theta)$ with $\cos\theta = \lambda$. A single qubit state does not encode a full orientation, since rotations about the Bloch vector fix it. Three heads started at $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$ under the same rotations carry the columns of the rotation matrix.

A.4 Finite Group Encodings

Proposition 1 uses $d = |G|$. Much smaller heads suffice for the groups of Section 4.

Proposition 5 (Finite group encodings)**.** *One qubit head encodes the A_5 word problem exactly, with a distinct state for every group element. For S_5 no finite number of independent qubit heads does, while one head with $d = 4$ does.*

Proof. An exact encoding assigns one joint state to each group element and updates it by left multiplication. For a finite faithful projective unitary action, a pure state off the finitely many proper eigenspaces of the nonidentity elements has trivial stabilizer, so its orbit is injective. The orbit vectors have equal norm, so an argmax over inner products with them recovers the group element, and one-hot group symbols select the unitaries exactly.

A_5 is the rotation group of the icosahedron, so it acts on one qubit through $PU(2) \cong SO(3)$. A unit Bloch vector off every rotation axis has 60 distinct images.

For S_5, suppose finitely many independent qubit heads encode it exactly. Each head's orbit carries a quotient action of S_5, whose image is S_5, $\mathbb{Z}_2$, or trivial. If the orbit spans affine dimension three, a channel permuting it has finite order, since an affine map fixing a spanning set is the identity. Its inverse is then a power of it, so Proposition 4 makes it a rotation, and the finite subgroups of $SO(3)$ are cyclic, dihedral, A_4, S_4, and A_5. In affine dimension at most two the action is conjugate into $O(2)$ and is cyclic or dihedral (Bröcker & tom Dieck, 1985). Either way the image is not S_5, so every head factors through the sign, and distinct even permutations share the joint state, a contradiction. The argument concerns exact encodings by independent heads, not deeper models or approximate prediction.

For $d = 4$, restrict the permutation representation of S_5 on $\mathbb{R}^5$ to the sum-zero subspace, a faithful $\pi\colon S_5 \to O(4)$. Its only possible scalar images are $\pm I_4$, and $-I_4$ is excluded because a permutation of five coordinates has a -1 eigenspace of dimension at most two. So $g \mapsto \mathrm{Ad}_{\pi(g)}$ embeds S_5 in $PU(4)$ at $r = 1$, and a generic pure state has 120 distinct images. □

A.5 Proof of Proposition 3

Proof of Proposition 3. Every channel maps $\mathcal{D}_d$ into itself, so $K = \mathcal{D}_d$ satisfies Definition 5 for every set of inputs, bounded or not. For Hermitian $X = X_+ - X_-$ with $X_\pm \succeq 0$, positivity and trace preservation give

$$\|\Phi[X]\|_1 \le \|\Phi[X_+]\|_1 + \|\Phi[X_-]\|_1 = \operatorname{Tr} X_+ + \operatorname{Tr} X_- = \|X\|_1. \tag{18}$$

With $X = \rho - \rho'$ this is the claimed bound, and a unitary conjugation preserves the trace norm. For $d = 2$, $\rho(\mathbf{x}) - \rho(\mathbf{x}') = \frac{1}{2}(\mathbf{x} - \mathbf{x}') \cdot \boldsymbol{\sigma}$ has eigenvalues $\pm\|\mathbf{x} - \mathbf{x}'\|_2/2$, so

$$\|\rho(\mathbf{x}) - \rho(\mathbf{x}')\|_1 = \|\mathbf{x} - \mathbf{x}'\|_2. \tag{19}$$

The offset $\mathbf{b}_t$ cancels in differences, and taking $\mathbf{x} = \mathbf{v}$, $\mathbf{x}' = -\mathbf{v}$ for unit $\mathbf{v}$ gives $\|\mathbf{A}_t\|_2 \le 1$. For fixed inputs the recurrence is affine, so $\partial \mathbf{x}_T / \partial \mathbf{x}_t = \mathbf{A}_T \cdots \mathbf{A}_{t+1}$, whose spectral norm is at most one. □

Invariance also bounds the real state at every length,

$$\operatorname{Tr} \rho^2 = \frac{1}{d} + \frac{1}{2}\|\mathbf{x}\|_2^2 \le 1, \qquad \|\mathbf{x}\|_2 \le \sqrt{\frac{2(d-1)}{d}}. \tag{20}$$

The Euclidean Jacobian bound is stated only for $d = 2$, since for $d \ge 3$ channels contract the trace norm but not in general the Euclidean norm of Bloch vectors. Both bounds compare trajectories under the same inputs. They do not bound gradients with respect to parameters or inputs, or the Jacobian of the surrounding network, and they do not prevent vanishing sensitivity along repeatedly contracted directions.

A.6 State Minimality

Proof of Lemma 1. Let $\mathcal{K} \subset \mathbb{R}^k$ be a compact convex body and let G be a connected non-abelian group of affine automorphisms. Its closure is still connected and non-abelian, so take G closed. Every affine automorphism fixes the centroid. After translation, G acts linearly. If $\mathcal{K}$ contains a ball of radius ε and lies in one of radius R, every $g \in G$ has operator norm at most R/ε. Thus G is a compact subgroup of $GL(k)$. Averaging an inner product over G conjugates it to a closed subgroup of $O(k)$. The connected closed subgroups of $O(1)$ and $O(2)$ are trivial or conjugate to $SO(2)$, hence abelian. Therefore $k \ge 3$.

For the case $k = 3$ discussed after the lemma in Section 3.1, a connected non-abelian closed subgroup of $O(3)$ is $SO(3)$, so $G = Q\,SO(3)\,Q^{-1}$ for some $Q \in GL(3)$. Then $Q^{-1}\mathcal{K}$ is invariant under $SO(3)$, and its support function is constant on the unit sphere. Hence it is a Euclidean ball centered at the origin, and $\mathcal{K}$ in centered coordinates is a linear image of a ball. Undoing the centroid translation makes the original body affinely equivalent to a ball. □

This geometric consequence does not uniquely select CPTP transitions.

A.7 Transition-Family Comparison

Entries of Table 1 refer to the transitions a single token can select, and the width and the number of Householder factors may grow with the task. The CPTP entries follow from Propositions 1–3.

Conservative and dissipative columns. Permutation and merge follow Merrill et al. (2024), Walker et al. (2025), Siems et al. (2025), and Terzić et al. (2025). RWKV-7 merges in the copy regime of Peng et al. (2025). PD-SSM permutes and merges with unit-modulus diagonals but attains state invariance with diagonals below one, so its entries hold in different regimes. Rotation fails for

diagonal and column-one-hot transitions, whose connected groups are abelian, and for diagonal plus rank-one transitions, whose spectra are real, so that every element of a compact group of them is an involution. Mixing requires a convex family. It holds for diagonal and block-diagonal transitions, for Householder products with enough factors, which fill the spectral-norm unit ball (Grazzi et al., 2025), and for channels, and it fails for column-one-hot and diagonal plus rank-one transitions, whose families are not convex.

Stability columns. Transitions are non-expansive in every ℓ_p norm for diagonal SSMs, in the spectral norm for Householder products, in the ℓ_1 norm for PD-SSM, and in the trace norm for channels. Mamba and PD-SSM attain state invariance through discretization and BIBO stability, respectively. Block-diagonal and rank-one products can have an eigenvalue above one and so expand, and Householder products have fixed points that are unbounded over a bounded input set, so none of these has an invariant set. State invariance implies non-expansiveness: if $T(K) \subseteq K$ for every transition $T(x) = Ax + b$, then $C = \operatorname{conv} K$ is invariant and $A(C - C) \subseteq C - C$, so A is non-expansive in the norm whose unit ball is $C - C$, and $b = T(c) - Ac$ stays bounded.

B Implementation details

This section covers what is needed to interpret the reported results. Kernel-level details are in the released code.

B.1 Computation of a qudit head

One layer computes the following for each token. All arithmetic is real; complex matrices are stored as separate real and imaginary parts.

1. For each head h, read $(\operatorname{Re} Z_\ell, \operatorname{Im} Z_\ell)_{\ell=1}^{r}$ from $W_{\text{in}}^{(h)} u_t + \mathbf{b}_{\text{in}}^{(h)}$.
2. Form $S = \sum_\ell Z_\ell^\dagger Z_\ell$ and $S^{-1/2}$ (Appendix B.2).
3. Form the Kraus operators $K_\ell = Z_\ell S^{-1/2}$.
4. Form $G = \sum_\ell \operatorname{vec}(K_\ell) \operatorname{vec}(K_\ell)^*$, a $d^2 \times d^2$ Hermitian matrix of rank at most r.
5. Obtain $(\mathbf{A}^{(h)}(u_t), \mathbf{b}^{(h)}(u_t))$ from G by the fixed contraction equation 21.
6. Compose the pairs along the sequence with the scan of Section 2.1.
7. Read out $y_t = W_o(\mathbf{x}_t^{(1)} \| \cdots \| \mathbf{x}_t^{(n_q)}) + \mathbf{b}_{\text{out}}$.

Steps 1–5 are independent across heads and tokens. Step 6 is the only one that couples tokens.

With row-major vectorization, $G_{(b,c),(a,e)} = \sum_\ell [K_\ell]_{bc} \overline{[K_\ell]_{ae}}$ is the Choi matrix of the transition up to a fixed index permutation (Choi, 1975), and expanding the traces in equation 7 gives

$$\big[\mathbf{A}^{(h)}(u_t)\big]_{ij} = \frac{1}{2} \sum_{a,b,c,e} [\Lambda_i]_{ab} [\Lambda_j]_{ce} \, G_{(b,c),(a,e)}, \qquad \big[\mathbf{b}^{(h)}(u_t)\big]_i = \frac{1}{d} \sum_{a,b,c} [\Lambda_i]_{ab} \, G_{(b,c),(a,c)}. \tag{21}$$

This is a single product of the $2d^4$ real entries of G with a constant matrix that depends only on the basis. We use the generalized Gell–Mann basis with $\operatorname{Tr}(\Lambda_i \Lambda_j) = 2\delta_{ij}$ (Kimura, 2003; Bertlmann & Krammer, 2008). For $d \geq 3$ the code stores coordinates in a Hilbert–Schmidt orthonormal basis, which rescales them by $1/\sqrt{2}$.

B.2 The inverse square root

At $d = 2$, S has eigenvalues $\lambda_\pm = \frac{1}{2}\big(\operatorname{Tr} S \pm \sqrt{(s_{11} - s_{22})^2 + 4|s_{12}|^2}\big)$, and Cayley–Hamilton gives the closed form

$$S^{-1/2} = \beta S + \alpha I, \qquad \beta = \frac{\lambda_+^{-1/2} - \lambda_-^{-1/2}}{\lambda_+ - \lambda_-}, \qquad \alpha = \frac{\lambda_-^{-1/2} \lambda_+ - \lambda_+^{-1/2} \lambda_-}{\lambda_+ - \lambda_-}. \tag{22}$$

We evaluate it in single precision. The smaller eigenvalue is floored at $10^{-6}\lambda_+$, and near-coincident eigenvalues use the limit of the divided differences so that the gradient stays finite. If a floor is active, the floored $\tilde{S} \succeq S$ gives $\sum_\ell K_\ell^\dagger K_\ell = \tilde{S}^{-1/2} S \tilde{S}^{-1/2} \preceq I$: the map is still completely positive and trace non-increasing, so the Bloch ball is still mapped into itself.

For $d \geq 3$ we use the coupled Newton–Schulz iteration (Higham, 2008) on $S/\operatorname{Tr} S$,

$$W_k = \tfrac{1}{2}(3I - Z_k Y_k), \qquad Y_{k+1} = Y_k W_k, \qquad Z_{k+1} = W_k Z_k, \tag{23}$$

from $Y_0 = S/\operatorname{Tr} S$ and $Z_0 = I$, for 14 iterations, then rescale by $(\operatorname{Tr} S)^{-1/2}$. No floor is applied on this path.

B.3 BACKWARD PASS

The backward pass is hand-written. With $\mathbf{g}_t$ the direct loss gradient at step t, the adjoint

$$\mathbf{a}_T = \mathbf{g}_T, \qquad \mathbf{a}_t = \mathbf{g}_t + \mathbf{A}_{t+1}^\top \mathbf{a}_{t+1} \tag{24}$$

is the forward recurrence run backwards with transposed transitions, so it reuses the forward kernel. $\mathbf{A}_t^\top$ is the coordinate form of the adjoint channel $\Phi_t^*[X] = \sum_\ell K_{t,\ell}^\dagger X K_{t,\ell}$. The remaining gradients are $\partial\mathcal{L}/\partial\mathbf{A}_t = \mathbf{a}_t \mathbf{x}_{t-1}^\top$ and $\partial\mathcal{L}/\partial\mathbf{b}_t = \mathbf{a}_t$. Only transitions and states are stored, not the per-step graph.

Table 7 compares scan implementations. The custom adjoint stays below twice the forward cost (1.90 at $T = 1024$, 1.81 at $T = 16{,}384$), and adding the backward pass raises peak memory by 34%.

Table 7: Backward/forward runtime ratio of the scan at $d = 2$, $r = 4$, $n_q = 192$, $B = 4$, $T = 16{,}384$.

Scan implementation	Backward / forward
Chunked scan, custom adjoint	1.81
Framework associative scan, autograd	2.57
Sequential loop	14.77

B.4 COST

Table 8 counts multiply-accumulates per token for one layer. For qubit heads the projections dominate at practical widths. At $D = 128$, $n_q = 128$, $r = 2$ the channel algebra is about a sixth of the projection cost, and at $D = 1024$, $n_q = 8$ about two percent. This does not carry over to larger d. At $d = 4$, $n_q = 8$ the dense basis contraction alone costs about 983k multiply-accumulates per token, against 48k for the projections, which is why we widen layers through the number of qubit heads.

Table 8: Per-token multiply-accumulates of one layer, $D = D_{\text{in}} = D_{\text{out}}$.

Stage	Cost	$d = 2$, $r = 2$, per head
Input projection	$2n_q r d^2 D$	$16D$
Polar normalization	$O(n_q r d^3)$	≈ 150
Choi matrix	$O(n_q r d^4)$	≈ 130
Transfer (factored)	$O(n_q d^6)$	48
Scan composition	$O(n_q(d^2-1)^3)$, two per token	72
Output projection	$n_q(d^2-1)D$	$3D$
Recurrent state	$n_q(d^2-1)$ reals	3

C EXPERIMENTAL DETAILS

C.1 THE HISS ARCHITECTURE

CSP-Bench (Bhirangi et al., 2024) pairs a sensor stream sampled at 50 Hz with a target signal sampled at 5 Hz, so one target follows every ten input samples. A flat model processes the whole input sequence at 50 Hz and its predictions are subsampled to the target rate. HiSS is a two-level model. A low-level sequence model is applied to consecutive windows of ten input samples and emits its final hidden state for each window. A high-level sequence model runs over the resulting 5 Hz sequence of window features, followed by a linear output layer. The low-level and high-level models are chosen from S4, Mamba, and a causal Transformer, and we run the combinations and sizes that the authors' released configurations select for each task, listed in Table 11. The shipped JoystickControl configuration selects Mamba over S4, whereas the paper reports Transformer over S4 as the best combination for that task, so we compose the latter from the released configuration groups at the shipped sizes.

In the released code the low-level window for a target at time t ends nine samples after t, so the hierarchical model saw nine future inputs. We shift the windows so that each ends at its target timestamp. The first target of every sequence, an identically zero row that the tasks prepend, then has no window and is dropped, and the model emits $T-1$ predictions paired with targets $1, \ldots, T-1$. All flat rows, including CPTP-SSM, are causal by construction.

C.2 CSP-BENCH

Tasks. Table 9 summarizes the six tasks as prepared by the released preprocessing. The three tactile datasets were collected by Bhirangi et al. (2024), and the three inertial tasks are curated from RoNIN (Herath et al., 2020), VECtor (Gao et al., 2022), and TotalCapture (Trumble et al., 2017). The released preprocessing resamples every input to 50 Hz and every target to 5 Hz. MarkerWriting and IntrinsicSlip use ReSkin, a magnetic tactile sensor whose five three-axis magnetometers report the field of a magnetized elastomer, and JoystickControl uses Xela uSkin taxels, which report three-axis forces through Hall sensors. The three inertial tasks use accelerometer and gyroscope readings, and TotalCapture additionally provides the orientation of each of its thirteen IMUs as the first two columns of a rotation matrix. The targets are velocities obtained as differences of consecutive positions, except JoystickControl, whose target is the absolute joystick state. IntrinsicSlip reports the translational and the rotational component of the slip separately, which gives seven metric columns for six tasks.

Table 9: CSP-Bench tasks as used here. Input and target rates are 50 Hz and 5 Hz for every task.

Task	Sensor	Input	Target	Unit	Split
MarkerWriting	ReSkin	magnetometer	marker velocity in the plane	cm/s	sequential
IntrinsicSlip	ReSkin	magnetometer	slip translation, rotation	cm/s, deg/s	random
JoystickControl	Xela uSkin	taxel forces	joystick state, three axes	raw	random
VECtor	IMU	accel., gyro.	velocity in space	m/s	sequential
RoNIN	IMU	accel., gyro.	velocity in the plane	m/s	sequential
TotalCapture	13 IMUs	accel., orientation	21 joint velocities	m/s	sequential

Protocol. We follow the released training loop. Every model trains for 600 epochs with Adam at a constant learning rate of 10^{-3} and batch size 4, dropout is 0.1 where the model accepts it, and the checkpoint with the lowest validation loss is kept. The benchmark has no test split, and selection and reporting both use the validation split, as in the original paper. Each cell is the mean and standard deviation over seeds 1, 2, and 3.

Metric. The metric is the root mean squared error over the targets at positions $j \geq 1$ in the native units of each task, recomputed from the saved checkpoints by one script for every row. The error of a predictor that outputs zero at every step is 3.61 cm/s on MarkerWriting, 2.42 cm/s and 15.27 deg/s on IntrinsicSlip, 0.400 on JoystickControl, 0.173 m/s on VECtor, 0.676 m/s on RoNIN, and 0.798 m/s on TotalCapture, and a row is only meaningful below it. RoNIN predictions are un-normalized with

the training-split standard deviation for every row; the released code uses the validation file's own. RoNIN training applies a random rotation about the gravity axis, as released.

Models and hyperparameters. CPTP-SSM, Gated DeltaProduct, and Mamba-3 are swept on the comparable grids, taking the configuration from a single seed with the lowest best-validation loss. CPTP-SSM sweeps the model width $D \in \{64, 128, 256\}$ and the number of heads $n_q \in \{16, 64, 128\}$ for each Kraus rank $r \in \{2, 4\}$, Gated DeltaProduct sweeps $D \in \{64, 128, 256\}$ and the head dimension in $\{16, 64, 128\}$ for $n_h \in \{2, 4\}$, and Mamba-3 sweeps $D \in \{64, 128, 256\}$ and the state dimension in $\{32, 64, 128\}$. All three use six layers. The original baselines (S4, LSTM, and HiSS) use the authors' per-task configurations unchanged. We exclude the original Mamba, as its successor, Mamba-3, is included. Table 10 lists every selected configuration, Table 11 the HiSS configurations, and Table 12 the resulting parameter counts.

The block is pre-norm with a residual connection and dropout, a causal depthwise convolution of width four with a delta-tap initialization precedes the head, and the output projection has no bias. Gated DeltaProduct uses the same block with the layer taken as shipped from the flash-linear-attention library, and Mamba-3 uses the released Mamba model with its mixer replaced by Mamba-3 as shipped.

Table 10: Selected configurations behind each CSP-Bench cell. Cells give (D, n_q) for CPTP-SSM, (D, d_h) for Gated DeltaProduct with d_h the head dimension, (D, d_s) for Mamba-3 with d_s the state dimension, (D, L) for S4 and the Transformer with L the number of layers, and (h, h_c) for the LSTM with h the hidden and h_c the cell width. The columns are MarkerWriting, IntrinsicSlip, JoystickControl, VECtor, RoNIN, and TotalCapture.

	Marker	Slip	Joystick	VECtor	RoNIN	TotalCap.
CPTP-SSM, $r = 2$	128, 64	64, 128	128, 128	64, 64	128, 128	128, 128
CPTP-SSM, $r = 4$	128, 128	128, 128	128, 128	128, 128	64, 128	128, 128
Gated DeltaProduct, $n_h = 2$	128, 128	256, 128	128, 128	256, 128	256, 128	128, 128
Gated DeltaProduct, $n_h = 4$	256, 128	256, 128	128, 128	256, 128	256, 128	128, 128
Mamba-3	256, 64	256, 128	128, 128	64, 64	128, 32	128, 128
S4	512, 6	512, 6	512, 6	128, 6	256, 4	256, 4
LSTM	256, 1024	256, 512	256, 1024	64, 128	128, 256	256, 1024
Transformer	512, 4	128, 6	256, 6	128, 4	128, 6	128, 4

Table 11: HiSS configurations per task, as released. Each level is written as model, width $\times$ layers.

Task	High-level model	Low-level model
MarkerWriting	Mamba, 128×6	S4, 256×6
IntrinsicSlip	Mamba, 256×4	S4, 256×6
JoystickControl	Transformer, 256×4	S4, 256×4
VECtor	Mamba, 64×4	Mamba, 32×4
RoNIN	S4, 128×6	S4, 256×6
TotalCapture	Mamba, 128×4	S4, 256×6

Full results. Table 13 repeats Table 6 with the standard deviation over the three seeds.

Differences from the original paper. Our HiSS numbers differ from those published with the benchmark for documented reasons. The released hierarchical model looked nine samples ahead, and we made it causal. We recompute every metric from the saved checkpoints on targets $j \geq 1$ in native units, un-normalize RoNIN with the training-split standard deviation, and compose the JoystickControl configuration by hand as described above.

C.3 Group word problems

We follow the group word-problem protocol and hyperparameters of Walker et al. (2025), replacing the sequence layer in their block with CPTP-SSM. Baseline sizes follow their matching scheme

Table 12: Parameters, in millions, of the configuration behind each CSP-Bench cell. The columns are those of Table 10.

	Marker	Slip	Joystick	VECtor	RoNIN	TotalCap.
CPTP-SSM, $r = 2$	0.95	0.95	2.17	0.48	1.89	1.92
CPTP-SSM, $r = 4$	3.48	3.48	3.75	3.47	1.75	3.51
Gated DeltaProduct, $n_h = 2$	2.22	4.40	2.50	4.39	4.39	2.25
Gated DeltaProduct, $n_h = 4$	6.80	6.80	3.72	6.79	6.79	3.47
Mamba-3	2.64	2.87	1.11	0.21	0.66	0.87
S4	3.98	3.98	5.08	0.40	0.80	0.87
LSTM	13.67	3.70	14.22	0.23	0.92	13.77
Transformer	6.34	0.61	2.94	0.40	0.60	0.44
HiSS	1.94	3.02	3.01	0.17	1.62	1.76

Table 13: Root mean squared error on CSP-Bench, mean and standard deviation over three seeds, in the native unit of each task. Columns as in Table 6. The lowest error in each column is in bold.

	MW	IS		JC	R	V	TC
	(cm/s)	Trans. (cm/s)	Rot. (deg/s)	(–)	(m/s)	(m/s)	(m/s)
Transformer	1.572 ± 0.017	1.678 ± 0.017	11.14 ± 0.18	0.322 ± 0.002	0.314 ± 0.005	0.198 ± 0.012	0.642 ± 0.001
LSTM	1.083 ± 0.012	1.343 ± 0.017	8.09 ± 0.13	0.339 ± 0.007	0.189 ± 0.003	0.144 ± 0.007	$\mathbf{0.454 \pm 0.001}$
S4	1.143 ± 0.003	1.286 ± 0.013	8.10 ± 0.14	$\mathbf{0.320 \pm 0.001}$	0.196 ± 0.004	0.165 ± 0.008	0.556 ± 0.002
Mamba-3	0.993 ± 0.006	1.214 ± 0.038	7.00 ± 0.36	0.332 ± 0.003	0.188 ± 0.005	0.145 ± 0.010	0.628 ± 0.038
Gated DeltaProduct$_{n_h=2}$	1.197 ± 0.027	1.359 ± 0.028	8.25 ± 0.13	0.328 ± 0.004	0.185 ± 0.002	0.150 ± 0.003	0.616 ± 0.006
Gated DeltaProduct$_{n_h=4}$	1.169 ± 0.011	1.317 ± 0.007	8.15 ± 0.09	0.328 ± 0.002	0.187 ± 0.001	0.153 ± 0.007	0.624 ± 0.003
HiSS	1.048 ± 0.003	1.194 ± 0.003	7.16 ± 0.19	0.325 ± 0.002	0.181 ± 0.002	0.150 ± 0.009	0.525 ± 0.007
CPTP-SSM$_{d=2,\ r=2}$	0.989 ± 0.021	1.206 ± 0.012	7.26 ± 0.18	0.332 ± 0.001	$\mathbf{0.170 \pm 0.002}$	$\mathbf{0.139 \pm 0.004}$	0.528 ± 0.021
CPTP-SSM$_{d=2,\ r=4}$	$\mathbf{0.980 \pm 0.012}$	$\mathbf{1.158 \pm 0.024}$	$\mathbf{6.89 \pm 0.22}$	0.329 ± 0.003	0.171 ± 0.002	0.141 ± 0.006	0.530 ± 0.016

of 1024 hidden dimensions or approximately 1024 nonzero transition parameters per layer. Gated DeltaProduct and Mamba use the minimum depths reported to solve the in-distribution task in their Figure 1(a). Models are trained on lengths 3–40 using AdamW with batch size 256, dropout 0.1, and at most 100,000 steps. We use a peak learning rate of 10^{-3} with 10% linear warmup and cosine decay to 10^{-5}. Training stops early when validation accuracy on lengths 40–128 reaches 0.9995. For A_5, test accuracy is evaluated on 512 sequences per length and averaged over three seeds. The S_5 experiments use the same protocol with dropout set to zero and report the best validation accuracy averaged over three seeds. Model configurations are listed in Table 14.

Table 14: Configurations of the group word problem experiments. All arms use width 128 unless stated.

Arm	Layers	Width	Model settings
CPTP-SSM, qubit, $r = 4$	1	128	32 heads
S-SLiCE	1	128	sparsity $128^{\epsilon-1}$ with $\epsilon = 3/7$, block size 1
BD-SLiCE	1	256	block size 4
Gated DeltaProduct	2	1024	$n_h = 4$ Householder factors, gated
Mamba	3	1024	Mamba-2 block, library defaults
LSTM	1	1024	
sLSTM	1	1024	xLSTM block with the sLSTM cell
PD-SSM	1	128	state size 128, $K = 60$
S_5: CPTP-SSM, qubit, $r = 4$	1	128	32 heads
S_5: CPTP-SSM, qudit $d = 4$, $r \in \{1, 6\}$	1	128	8 heads

C.4 FINITE-STATE AUTOMATON EMULATION

The four tasks and the protocol are those of Terzić et al. (2025), who take them from Walker et al. (2025): 100,000 training steps on sequences of length 3 to 40, evaluation on lengths 40 to 256, and the mean and standard deviation over five seeds of the best validation accuracy. All baseline numbers are copied from their Table 2. CPTP-SSM uses two layers of $n_q = 42$ qubit heads, a state dimension of 126 per layer against the two layers of state dimension 128 of PD-SSM, at Kraus rank $r \in \{1, 4\}$.

C.5 MULTIVARIATE TIME-SERIES CLASSIFICATION

We follow the original UEA experimental protocol of Walker et al. (2024), replacing the state-space layer in the block of Terzić et al. (2025) with our CPTP-SSM layer. Models are trained for up to 100,000 steps with early stopping, and test accuracy is reported over five seeds at the checkpoint with the highest validation accuracy. Hyperparameters are selected by validation accuracy over learning rates $10^{-5}, 10^{-4}, 10^{-3}$, numbers of blocks $2, 4, 6$, embedding dimensions $16, 64, 128$, and inclusion of a time channel. We use $n_q \in \{11, 43, 85\}$ qubit heads to approximately match the real-state storage of the baseline complex states of size $16, 64, 128$.

Table 15: Selected UEA configurations: learning rate, number of blocks, embedding dimension, number of heads, and whether the time channel is appended.

	CPTP-SSM, $r = 2$					CPTP-SSM, $r = 4$				
	lr	blocks	emb.	n_q	time	lr	blocks	emb.	n_q	time
EigenWorms	10^{-4}	2	128	43	no	10^{-3}	4	16	43	yes
SCP1	10^{-5}	6	64	43	no	10^{-5}	4	128	43	no
SCP2	10^{-4}	2	64	11	yes	10^{-3}	4	128	11	yes
EthanolConcentration	10^{-3}	6	64	85	no	10^{-3}	6	64	43	no
Heartbeat	10^{-3}	4	128	85	yes	10^{-3}	2	128	85	yes
MotorImagery	10^{-5}	2	16	85	yes	10^{-3}	4	16	43	yes

Table 16: Test accuracy (%) of CPTP-SSM at $r = 2$ on the six UEA datasets, mean and standard deviation over five random initializations.

	Worms	SCP1	SCP2	Ethanol	Heartbeat	Motor	Avg
CPTP-SSM$_{d=2,\, r=2}$	95.6 ± 2.2	85.7 ± 6.0	56.5 ± 6.6	37.5 ± 4.8	81.0 ± 4.1	53.0 ± 6.2	68.2

C.6 NOISY $SO(3)$ BELIEF TRACKING

Exact reference. Let G_t be independent Haar rotations. Let E_t be iid rotation noises, independent of all G_s. Define

$$O_t = G_t E_t, \qquad R_t = G_t R_{t-1}, \qquad R_0 = I.$$

Haar invariance makes O_t independent of E_t. The pairs (O_t, E_t) are independent across time. Conditioning on $O_{1:t}$ therefore leaves the noises independent with their prior laws. Since $G_t = O_t E_t^{-1}$,

$$M_t = \mathbb{E}[R_t \mid O_{1:t}] = O_t \, C \, M_{t-1}, \qquad C = \mathbb{E}[E_t^{-1}]. \tag{25}$$

For $E_t = R_x(\phi_x) R_y(\phi_y) R_z(\phi_z)$ with independent $\phi_i \sim \mathcal{N}(0, \sigma_i^2)$, set $c_i = e^{-\sigma_i^2/2}$. Sine terms vanish and each cosine has expectation c_i. Independence gives

$$C = \mathrm{diag}(c_y c_z, c_x c_z, c_x c_y).$$

The map $v \mapsto O_t C v$ is a mixture of rotations and hence a unital qubit-channel Bloch action. Ball preservation alone would not prove complete positivity.

For an explicit channel, let $(\lambda_x, \lambda_y, \lambda_z) = (c_y c_z, c_x c_z, c_x c_y)$ and

$$p_0 = (1 + \lambda_x + \lambda_y + \lambda_z)/4, \quad p_x = (1 + \lambda_x - \lambda_y - \lambda_z)/4,$$
$$p_y = (1 - \lambda_x + \lambda_y - \lambda_z)/4, \quad p_z = (1 - \lambda_x - \lambda_y + \lambda_z)/4.$$

They sum to one. For example, $4p_x \geq (1 - c_y)(1 - c_z)$, and the other two inequalities follow by permutation. Thus all are nonnegative. With $\sigma_0 = I$ and any pointwise unitary lift $U(O_t)$, the operators $K_j = \sqrt{p_j} U(O_t)\sigma_j$ give the desired action. At the noise levels used here all four are positive, and since the Pauli operators are linearly independent the reference channel has Kraus rank four. This does not mean that a multi-head predictor needs $r = 4$.

Since $\|R_t\|_F^2 = 3$, orthogonality of conditional expectation gives

$$\mathbb{E}[\|R_t - M_t\|_F^2 \mid O_{1:t}] = 3 - \|M_t\|_F^2.$$

After averaging observations, the right side is $3 - \mathbb{E}\|M_t\|_F^2$. For any observation-measurable square-integrable prediction $\widehat{M}_t$,

$$\mathbb{E}\|R_t - \widehat{M}_t\|_F^2 = \mathbb{E}\|R_t - M_t\|_F^2 + \mathbb{E}\|M_t - \widehat{M}_t\|_F^2.$$

Agreement with M_t therefore measures excess squared-loss risk.

Haar increments are uniform on $SO(3)$ and re-randomize the body frame at every step, so the belief contracts almost isotropically. For a non-uniform increment prior, we numerically approximate the per-step posterior mean using Gauss–Hermite quadrature over the three noise angles.

Generative settings and reference scale. The experiment uses independent axis noises $\phi_i \sim \mathcal{N}(0, \sigma_i^2)$ with $\sigma = (0.0508,\ 0.0762,\ 0.1017)$ radians, or $(2.9^\circ,\ 4.4^\circ,\ 5.8^\circ)$ per step. These values give $C = \mathrm{diag}(0.9920,\ 0.9936,\ 0.9958)$ in equation 25. The ratios $1 : 1.5 : 2$ make the contraction anisotropic. The noise level is set so that $\mathbb{E}\|M_{128}\|_F/\sqrt{3} = 0.45$, so the belief has clearly shrunk by the end of evaluation. It is a design choice, not a sensor model. For reference, $\mathbb{E}\|M_t\|_F^2 = 2.963$, 1.821, and 0.607 at $t = 1$, 40, and 128.

Protocol. Every arm uses the code path of Appendix C.3 with three changes. The token embedding is replaced by a linear map from the nine observed entries to the model width, identically for every arm. The loss is the masked per-token squared error on nine real outputs. There is no early stopping, and the final checkpoint after 100,000 steps is the only evaluation point, since the validation score is a regression error on one fixed set and selecting on it would import its noise. Training streams a fresh batch of 256 sequences of lengths 3 to 40 at every step, with dropout 0, one recurrent block of width 128 (two for the two-layer arm), and AdamW with a linear warm-up over the first ten percent of steps and cosine decay from 2×10^{-3} to 10^{-5}. Evaluation uses one fixed set of 512 sequences of length 128 scored at every position. No model carries a length or position term, so position t of this set is statistically a sequence of length t, which was checked against separately generated sets of lengths 40, 64, and 96.

Metrics. With $\hat{M}_t$ the prediction and M_t the exact posterior mean, averaged over the 512 sequences, $\mathcal{R}_t^2 = 1 - \mathbb{E}\|\hat{M}_t - M_t\|_F^2 / \mathbb{E}\|M_t\|_F^2$.

D Related work

State tracking in state-space models. Formal analyses bound what input-selective layers with diagonal transitions can track (Merrill et al., 2024; Sarrof et al., 2024), and a series of transition families has been proposed to pass the bound: negative eigenvalues and Householder products (Grazzi et al., 2025; Siems et al., 2025), asymmetric rank-one updates (Peng et al., 2025), column-one-hot matrices (Terzić et al., 2025), and block-diagonal, sparse, and Walsh–Hadamard structures (Walker et al., 2025). Nonlinear recurrences reach the same tasks through memory mixing (Beck et al., 2024), and fixed-point iterations of a diagonal recurrence recover a dense one at the cost of depth (Movahedi et al., 2025). Partial observation, where the tracked state becomes a belief, has recently entered this line of work (Siems et al., 2026). All of these designs constrain the transition. We begin with a compact state geometry and choose CPTP maps as a composition- and mixture-closed family preserving that geometry.

Sequence models of sensor streams. Bhirangi et al. (2024) introduced CSP-Bench, six sensor-to-state prediction tasks over raw tactile and inertial streams, because prior work on such prediction had studied one class of sensors at a time and general-purpose sequence models could not be compared on it. On this benchmark state-space layers outperform the best Transformer and LSTM baselines by 10% and 14% in median error, and their hierarchical model HiSS improves on the best flat model by a further 23%. Linear recurrences and state-space layers also serve as policy backbones in robot learning (Yoo et al., 2026; Jia et al., 2024). Learned inertial navigation and odometry form a separate literature (Liu et al., 2020; Jayanth et al., 2025), in which state-space layers now appear as well (Qin et al., 2026; Li et al., 2025). Our contribution is the layer itself, and its integration into such systems is left to future work.

Unitary and stochastic transitions. Unitary recurrent neural networks constrain every transition to the unitary group, so that the norm of the state is preserved over long horizons (Arjovsky et al., 2016; Lezcano-Casado & Martínez-Rubio, 2019). Probabilistic automata and hidden Markov models constrain every transition to be column-stochastic, so that the state remains a distribution (Rabin, 1963; Rabiner, 1989; Chiu & Rush, 2020). Each family realizes one of the two capabilities discussed and places both inside the family of channels.

Quantum-formalism sequence models. Hidden quantum Markov models carry a density matrix and update it with a Kraus operator indexed by the observed symbol (Monras et al., 2011; Srinivasan et al., 2018; Adhikary et al., 2020). Their transitions form a fixed finite set per symbol, they are not input-selective functions of continuous tokens, and they are not state-space layers. Tensor-network language models carry a pure quantum state and admit parallel evaluation (Miller et al., 2021). A concurrent recurrent cell with a density-matrix state and channel transitions uses the same polar normalization of Kraus operators as ours (Seo & Montiel, 2026). CPTP-SSM develops input-selected multi-head channels into a real-coordinate SSM with closed-form qubit computation and parallel associative scan. Its geometric and algebraic characterization connects state minimality, group tracking, automaton emulation, and stability, with experiments spanning discrete and continuous sequence tasks.