Pathwise Transported Memory Priors for Autoregressive Generative Models

Tomoya Mizuguchi [1] **Bum Jun Kim** [2]

Abstract

Autoregressive generative models sometimes continue from histories in which later causal operations transform earlier bindings; success may then reflect fixed-coordinate memorization, generic recurrent capacity, or an inductive bias toward transported structure. We introduce Sylvester HiPPO (SHiPPO), a pathwise transported online-projection memory prior that lifts HiPPO-style coefficient memories into a moving channel frame. Conditional on any fixed or realized right-transport path, SHiPPO transports the channel metric and approximation family together, so its coefficient state follows ordinary HiPPO in the tied moving frame and obeys Sylvester dynamics. For selective sequence layers, we derive a restricted group-local realization whose right transport is compatible with the controller, together with exponential-adjusted updates, an exact block-affine scan, and a reducibility collapse criterion. On Transport-MQAR, a finite-field diagnostic for transported multi-query associative recall under length extrapolation, a scan-compatible SHiPPO realization improves coordinate-wise transported recovery over no-right and static-basis controls. Intra-family comparisons further indicate that the diagnostic supports a transported right-action pathway while leaving the choice of discrete right-action realization open. We position SHiPPO as a transported-memory prior and introduce Transport-MQAR as a diagnostic protocol for testing when autoregressive models transport memories rather than memorize fixed-coordinate associations.

[1]Kyoto University, Japan [2]The University of Tokyo, Japan. Correspondence to: Bum Jun Kim <bumjun.kim@weblab.t.u-tokyo.ac.jp>.

Published as a paper at the 1st FoGen workshop, ICML 2026, Seoul, South Korea, 2026. Copyright 2026 by the author(s).

1. Motivation and Contributions

Autoregressive generative models compress a revealed history into a causal state and then continue the sequence from that state. When the history contains bindings that are transformed by later causal operations, successful generation requires more than recalling fixed key–value pairs: the model must maintain the coordinate frame in which past information should be read. This gives a controlled setting for a foundational question about deep generative models: whether apparent extrapolative success reflects memorization of fixed-coordinate patterns, generic recurrent capacity, or an inductive bias for structured generalization. We study this question through recurrent memory priors, using "prior" in the modeling-bias sense of selecting a structured family of causal states rather than in the narrow Bayesian sense (Mitchell, 1980; Gordon & desJardins, 1995; Fortuin, 2022).

Projection-based recurrent memory provides a useful starting point because the hidden state has an explicit semantics. The Legendre Memory Unit derives a continuous-time memory from a Legendre representation of a sliding history window, and HiPPO formulates online history compression as projection onto a polynomial basis (Voelker et al., 2019; Gu et al., 2020). In HiPPO, the recurrent ordinary differential equation (ODE) is not merely an architectural template: it is the coefficient dynamics induced by an online approximation problem. This principle has shaped structured state-space models (SSMs) and their variants, including S4 and generalized-basis interpretations of HiPPO-style projections (Gu et al., 2022a; 2023). However, ordinary HiPPO is one-sided in an important sense: it stores temporal coefficients of a vector-valued history in fixed channel coordinates. It does not by itself say how a memory should evolve when the relevant channel frame is causally transported by subsequent observations.

Modern sequence backbones have moved far beyond independent one-sided memories. S5 uses multi-input multi-output (MIMO) state-space layers, H3 adds multiplicative structure for recall and comparison, Mamba makes SSM parameters input-dependent, and structured state-space duality (SSD), as used in Mamba-2, explains selective recurrences (Smith et al., 2023; Fu et al., 2023; Gu & Dao, 2024; Dao & Gu, 2024). Gated linear attention, DeltaNet, HGRN2, Gated DeltaNet, and recent Mamba variants further show

that matrix-valued states, gates, token-dependent updates, and channel interactions are powerful ingredients for efficient sequence modeling (Yang et al., 2024a;b; Qin et al., 2024; Yang et al., 2025; Lahoti et al., 2026). At the same time, diagnostic and theoretical studies of state tracking, input selectivity, and length generalization show that generic recurrence or selectivity alone does not settle what structure the model has learned (Merrill et al., 2024; Terzić et al., 2025a; Huang et al., 2025). The question we ask is therefore narrower than whether channel interaction is useful: can channel interaction itself be given online-projection memory semantics?

We introduce *SHiPPO* (*Sylvester HiPPO*), a pathwise transported online-projection memory prior. Given an online-memory basis and an admissible right-transport path, SHiPPO transports the channel metric and the approximation family together. Conditional on any fixed or realized transport path, the coefficient state is ordinary HiPPO in a tied moving channel frame and satisfies Sylvester dynamics,

$$\dot{C}_S(t) = A_L(t)C_S(t) + B_L(t)f(t)^\top + C_S(t)A_R(t).$$

The left operators A_L and B_L are inherited from the chosen online approximation problem, while A_R describes how channel coordinates are transported along the realized history. If A_R is produced by a causal controller, the full input–output map may be nonlinear, but the projection statement remains pathwise: after conditioning on the realized controller trajectory, the inner optimization is still over the coefficient matrix C, not over the transport path. This is the main distinction from a generic matrix-valued recurrence or a free encoder–decoder around HiPPO: the history encoder, coefficient decoder, metric, and Sylvester gauge term are tied by the same transport path.

The operator-level construction does not automatically yield a lightweight selective SSM layer. To obtain an efficient realization, we impose computational restrictions: channels are partitioned into small transport groups, the left dynamics remain diagonal in the state dimension and tied within each group, and the right action is controller-compatible and group-local. The resulting discrete recurrence has exact block-affine prefix-scan closure for the chosen right action. We also identify a simultaneous-reducibility collapse criterion: if the right-generator family preserves a fixed nontrivial channel decomposition, then the apparent transport reduces, after static mixing, to independent scalar or blockwise transported banks. This motivates non-reducible right-action families in the scan-compatible realization.

We evaluate this prior on Transport-MQAR, a finite-field diagnostic for transported associative recall under length extrapolation. The diagnostic probes fixed-coordinate recall, static-basis flexibility, standard sequence-model capacity, and the usefulness of a learned right-action pathway. Our empirical claim is deliberately limited: a scan-compatible SHiPPO realization improves coordinate-wise transported recovery over no-right and static-basis controls in this diagnostic, and a controller-suffix intervention shows that the trained predictor uses the right-action pathway. Intra-family comparisons further indicate that the diagnostic supports a transported right-action pathway while leaving the choice of discrete right-action realization open. We therefore present SHiPPO as a structured memory prior, with Transport-MQAR and the controller-suffix intervention serving as diagnostics for transported coordinate recovery, rather than as a general-purpose performance-dominance claim for autoregressive modeling.

Contributions. First, we formulate SHiPPO as a transported online-projection memory prior for autoregressive sequence models with transformed bindings. Second, we prove that, conditional on any realized right-transport path, SHiPPO is ordinary HiPPO in a tied moving channel frame and obeys Sylvester coefficient dynamics. Third, we derive a restricted scan-compatible selective realization with group-local right transport and exact block-affine scan. Fourth, we introduce Transport-MQAR and a controller-suffix intervention as diagnostics for transported coordinate recovery and dependence on a learned right-action pathway.

2. SHiPPO: Pathwise Transported Projection Memory

SHiPPO (*Sylvester HiPPO*) is the transported analogue of HiPPO-style online projection memory (Gu et al., 2020; 2023). The point of the construction is not to postulate a more expressive matrix recurrence, but to ask when a matrix-valued state can still be interpreted as coefficients of an online approximation problem. Throughout this section, fix an admissible right-generator path $A_R \in L^1([0,T];\mathbb{R}^{d\times d})$. All identities below are conditional on this fixed or realized path. If A_R is produced by a causal controller in a trainable autoregressive model, the full input–output map may be nonlinear, but the projection statement remains *pathwise*: after conditioning on the realized controller trajectory, the inner optimization is only over the coefficient matrix C, not over the transport path.

Setup and ordinary online projection. Let $f : [0,\infty) \to \mathbb{R}^d$ be a d-channel signal. For each $t > 0$, let μ_t be a measure on $[0,t]$, and let $\Phi_t : [0,t] \to \mathbb{R}^N$ be a basis with invertible Gram matrix

$$G(t) := \int_0^t \Phi_t(\tau)\Phi_t(\tau)^\top \, d\mu_t(\tau).$$

When $d\mu_t(\tau) = w_t(\tau)d\tau$, define $\psi(t,\tau) = w_t(\tau)\Phi_t(\tau)$, and assume the usual HiPPO closure condition

$$\partial_t \psi(t,\tau) = A_L(t)\psi(t,\tau) \quad (\tau < t),$$
$$B_L(t) := \psi(t,t).$$

Ordinary vector-valued HiPPO approximates $f|_{[0,t]}$ by $C^\top \Phi_t(\tau)$. Its coefficient matrix $C_H(t)$ satisfies

$$G(t)C_H(t) = \int_0^t \Phi_t(\tau) f(\tau)^\top \, d\mu_t(\tau),$$

and, in the orthonormalized case $G(t) = I_N$, obeys the one-sided coefficient ODE

$$\dot{C}_H(t) = A_L(t)C_H(t) + B_L(t)f(t)^\top.$$

Appendix A.2 recalls the variational derivation and the non-orthonormal form.

Transported approximation problem. Let $P(t,\tau) \in GL(d)$ be the state transition associated with the chosen right generator,

$$\partial_t P(t,\tau) = P(t,\tau)A_R(t), \qquad P(\tau,\tau) = I_d,$$

which exists and is invertible under the integrability assumption on A_R (Coddington & Levinson, 1955). Define the transported channel metric

$$M_P(t,\tau) := P(t,\tau)P(t,\tau)^\top,$$
$$\|u\|^2_{M_P(t,\tau)} := u^\top M_P(t,\tau)u.$$

Definition 2.1 (SHiPPO online approximation problem)**.** For each $t > 0$, define the transported approximation family

$$\mathcal{G}_t^{\mathrm{SH}} = \{\, \tau \mapsto P(t,\tau)^{-\top} C^\top \Phi_t(\tau) : C \in \mathbb{R}^{N\times d} \,\}.$$

Define

$$r_t(C;\tau) := f(\tau) - P(t,\tau)^{-\top} C^\top \Phi_t(\tau).$$

The SHiPPO coefficient matrix is

$$C_S(t) := \operatorname*{arg\,min}_{C\in\mathbb{R}^{N\times d}} J_t^{\mathrm{SH}}(C),$$
$$J_t^{\mathrm{SH}}(C) := \int_0^t \|r_t(C;\tau)\|^2_{M_P(t,\tau)} \, d\mu_t(\tau).$$

We write $C_S(t) = \mathrm{shippo}_t(f)$.

The approximation family and the channel metric are transported together. This coupling is the source of the finite coefficient closure: changing only the metric while keeping the ordinary HiPPO family generally introduces a τ-dependent linear operator on C, rather than the Gram matrix $G(t)$ appearing in ordinary HiPPO. Appendix A.6 gives the stationary calculation and the τ-independent special case.

Conjugacy and Sylvester dynamics. For fixed t, define $(\mathcal{T}_t u)(\tau) := P(t,\tau)^\top u(\tau)$. Then $\mathcal{T}_t$ is an isometry from the SHiPPO metric to the Euclidean HiPPO metric and maps $\mathcal{G}_t^{\mathrm{SH}}$ bijectively onto the ordinary HiPPO family:

$$\int_0^t u^\top M_P v \, d\mu_t = \int_0^t (\mathcal{T}_t u)^\top (\mathcal{T}_t v) \, d\mu_t,$$
$$\mathrm{shippo}_t(f) = \mathrm{hippo}_t(\mathcal{T}_t f).$$

where the arguments (t,τ) are suppressed inside the integrals. The final equality is an identity of coefficient matrices; Appendix A.4 gives the full argument.

Theorem 2.2 (Normal equation and Sylvester coefficient dynamics)**.** *The SHiPPO coefficient matrix satisfies*

$$G(t)C_S(t) = \int_0^t \Phi_t(\tau) f(\tau)^\top P(t,\tau) \, d\mu_t(\tau).$$

If, in addition, $G(t) = I_N$, $d\mu_t(\tau) = w_t(\tau)d\tau$, the HiPPO closure condition above holds, and the usual Leibniz-rule regularity assumptions are satisfied, then, at differentiability times,

$$C_S(t) = \int_0^t \psi(t,\tau) f(\tau)^\top P(t,\tau) \, d\tau$$

and

$$\dot{C}_S(t) = A_L(t)C_S(t) + B_L(t)f(t)^\top + C_S(t)A_R(t).$$

Proof sketch. By the conjugacy above, $C_S(t)$ is the ordinary HiPPO coefficient matrix of $(\mathcal{T}_t f)(\tau) = P(t,\tau)^\top f(\tau)$, which yields the normal equation. In the orthonormalized case, differentiate the integral representation using Leibniz' rule, $\partial_t \psi = A_L \psi$, $\partial_t P = PA_R$, and $P(t,t) = I_d$. The boundary term gives $B_L f^\top$, while the interior derivatives give $A_L C_S$ and $C_S A_R$. Full first-variation and regularity details are in Appendix A.5. □

The ODE in Theorem 2.2 is a differential Sylvester equation (Behr et al., 2019; Simoncini, 2016). Its interpretation is more specific than a generic two-sided recurrence: the left operators A_L and B_L are inherited from the online approximation problem, whereas A_R is an external path that transports channel coordinates. Thus any closed one-sided projection-memory equation can be lifted pathwise by adding the right-action term $C_S A_R$, while architectural restrictions on A_R are introduced only for the scan-compatible realization of Section 3.

Moving-frame interpretation and non-reduction. SHiPPO reduces exactly to ordinary HiPPO when $A_R \equiv 0$, equivalently $P(t,\tau) \equiv I_d$. For a nontrivial path, let V solve $\dot{V}(t) = V(t)A_R(t)$, $V(0) = I_d$. Then $P(t,\tau) = V(\tau)^{-1}V(t)$, and SHiPPO factors as

$$\mathrm{shippo}_t(f) = \mathrm{hippo}_t\big(\tau \mapsto V(\tau)^{-\top} f(\tau)\big)V(t).$$

The history encoder, coefficient decoder, metric, and Sylvester gauge term are therefore tied by the same moving frame. This is not a static channel mixer, not an arbitrary encoder–decoder factorization, and not a claim that every matrix-valued recurrence has projection semantics. When A_R is input-dependent, the factorization remains valid pathwise for the realized trajectory, but it does not yield a fixed input-independent encoder–HiPPO–decoder reduction. Appendix B provides the reduction, identity-metric, and moving-frame details.

The operator-level SHiPPO statement is independent of how the realized right action is represented in a discrete layer. A single matrix exponential, a split product of factor exponentials, or another controller-compatible invertible map may all define realized right-transport paths. The scan-compatible cells below are therefore realization-level choices, not additional assumptions in the abstract transported projection problem.

3. Scan-Compatible Realization

Section 2 defines SHiPPO as an operator-level transported projection memory. It does not, by itself, prescribe a lightweight selective layer. In an autoregressive generative model, such a layer should support input-dependent transitions, parallel training over long contexts, and recurrent decoding, as in modern selective SSMs and structured state-space duality (Gu & Dao, 2024; Dao & Gu, 2024; Lahoti et al., 2026). We therefore study a restricted scan-compatible realization. The restrictions below are computational choices, not part of the abstract SHiPPO definition: channels are partitioned into small transport groups, the left dynamics are diagonal in the state dimension and tied within each group, and the right transport is group-local and controller-compatible.

Let the selective branch have width D, partitioned into G groups of width P, so $D = GP$. We write one group and omit the group index. The input is $x_t \in \mathbb{R}^P$, the memory is $H_t \in \mathbb{R}^{N\times P}$, and the right action is $R_t \in \mathrm{GL}(P)$. The continuous-time restricted lift is

$$\dot{H}(t) = \mathrm{Diag}(a(t))H(t) + b(t)x(t)^\top + H(t)A_R(t), \quad (1)$$

where $a(t), b(t) \in \mathbb{R}^N$ and $A_R(t) \in \mathbb{R}^{P\times P}$. The left term stores temporal coefficients; the right term transports channel coordinates inside the memory state. The factorized source $b(t)x(t)^\top$ is the lightweight selective-cell instance, but the scan algebra below also allows a general additive source $U(t) \in \mathbb{R}^{N\times P}$. A fully channelwise lift with row- and channel-specific left decays does not generally close in the small summary algebra below once R_t is non-diagonal; Appendix C.1 gives the two-step obstruction.

Controller-compatible block-affine scan. The right transport may be token-dependent, but exact finite scan requires it to be fixed with respect to the main memory variable being scanned. We encode this by a precomputed causal controller path $\xi_{1:T}$, obtained from the input or from an auxiliary causal module independent of the main memory recurrence.

Definition 3.1 (Controller-compatible right transport)**.** A right action $R_t \in \mathrm{GL}(P)$ is controller-compatible if, conditional on a precomputed causal controller path $\xi_{1:T}$,

$$R_t = R(\xi_t)$$

and R_t is independent of H_{t-1}. A transport of the form $R_t = R(\xi_t, H_{t-1})$ is state-coupled and is not controller-compatible unless that dependence is degenerate.

Conditional on such a controller path, the discrete recurrence has the affine two-sided form

$$H_t = L_t H_{t-1} R_t + U_t, \quad (2)$$

with $L_t \in \mathbb{R}^{N\times N}$, $R_t \in \mathrm{GL}(P)$, and $U_t \in \mathbb{R}^{N\times P}$ fixed for the step.

Proposition 3.2 (Exact block-affine scan)**.** *The summaries of two consecutive steps compose as*

$$\begin{aligned}(L_2, R_2, U_2) \star (L_1, R_1, U_1) \\ = (L_2L_1,\ R_1R_2,\ L_2U_1R_2 + U_2).\end{aligned} \quad (3)$$

Thus Eq. 2 admits an exact associative prefix scan (Blelloch, 1990; Martin & Cundy, 2018). If R_t depends directly on H_{t-1}, the step map is generically nonlinear in H_{t-1}, so this finite affine summary algebra is not closed without augmenting the state or imposing special degeneracies.

The scan in Proposition 3.2 is exact for the implemented discrete recurrence. It is not the original elementwise selective-scan algebra of Mamba: the one-channel case $P = 1$ recovers the scalar right-action limit, whereas nontrivial transport uses a group-local block-affine scan.

Collapse under reducible right actions. Controller compatibility preserves computation but not necessarily nontrivial transport. A token-dependent controller can still collapse to independent banks if all right generators preserve a fixed channel decomposition.

Proposition 3.3 (Fixed simultaneous-reducibility collapse)**.** *Suppose the right-generator basis satisfies $G_m = Q\Lambda_m Q^{-1}$ for a fixed $Q \in \mathrm{GL}(P)$, where every Λ_m is diagonal or block-diagonal with the same nontrivial block partition. For any $A_{R,t} = \sum_m \rho_{t,m} G_m$, every dense exponential $R_t = \exp(\Delta_t A_{R,t})$ and every fixed-order split product formed from these generators is diagonal or block-diagonal in the same basis. Under the static change of variables $\widetilde{H}_t = H_t Q$, the recurrence $H_t = L_t H_{t-1} R_t + U_t$*

decomposes into independent scalar or blockwise transported banks. If $U_t = b_t x_t^\top$, then $U_t Q = b_t (Q^\top x_t)^\top$, so the same basis change can be absorbed into the group input coordinates.

This is a sufficient degeneracy criterion, not a full classification. It rules out the case where apparently adaptive right transport only varies coefficients inside a simultaneously reducible generator family. Noncommutativity alone is not enough: noncommuting matrices may still share a fixed block decomposition. For genuinely channel-interacting transported memory, the implemented right-action family should avoid such fixed common reductions.

Split-flow transport and discrete cell. In experiments we instantiate the right generator with a small structured library

$$\begin{aligned} A_{R,t} = &-\operatorname{Diag}(d_t) + \sum_m \theta_{t,m}\Omega_m \\ &+ \sum_n \eta_{t,n} N_n + \sum_\ell \zeta_{t,\ell} p_\ell q_\ell^\top . \end{aligned} \tag{4}$$

where $d_t \geq 0$, $\Omega_m^\top = -\Omega_m$, and $N_n^2 = 0$. The diagonal part is dissipative, skew terms generate rotations, nilpotent terms generate shears, and optional low-rank terms add further channel interaction. A dense backend uses the matrix exponential $R_t = \exp(\Delta_t A_{R,t})$, in the standard matrix-function sense (Higham, 2008). A structured backend may instead define R_t as a fixed-order Lie–Trotter product of cheap factor exponentials (Hairer et al., 2006). The scan is exact after the chosen discrete right action has been fixed; approximation error arises only if one interprets a split backend as an approximation to the dense frozen flow.

For a step-frozen right generator $A_{R,t}$, the single matrix exponential $R_t = \exp(\Delta_t A_{R,t})$ is the canonical homogeneous right-flow discretization of Eq. 1. A split backend instead forms R_t as a fixed-order product of exponentials of generator factors. Thus both backends use exponentials, and both may yield dense right actions; they differ in whether the step action is one exponential of an assembled generator or a product of factor exponentials.

With $L_t = \exp(\Delta_t \operatorname{Diag}(a_t))$, raw source $U_t = b_t x_t^\top$, and $\lambda_t \in [0, 1]$, the implemented update is

$$\begin{aligned} \widehat{U}_t &= (1-\lambda_t)\Delta_t L_t U_{t-1} R_t + \lambda_t \Delta_t U_t, \\ H_t &= L_t H_{t-1} R_t + \widehat{U}_t. \end{aligned} \tag{5}$$

For the dense backend, Eq. 5 is a two-point exponential source discretization of Eq. 1 under step-frozen a_t and $A_{R,t}$. For a split backend, it defines the exact discrete recurrence for the implemented split action. If $A_{R,t} \equiv 0$, then $R_t = I_P$ and the cell reduces to the corresponding one-sided diagonal-left selective update; with $P = 1$ and $\lambda_t = 1$, this is the usual one-channel exponential-Euler selective update. Appendix C gives the proofs, closed-form factor actions, and source-quadrature details.

4. Transport-MQAR Diagnostic

We evaluate scan-compatible SHiPPO realizations using Transport-MQAR as a diagnostic of transported-memory priors. The goal is to probe several explanations for extrapolative recall in autoregressive models: fixed-coordinate memorization, static-basis flexibility, standard sequence-model capacity, the usefulness of a learned right-action pathway, and sensitivity to the chosen right-action realization. Our main testbed is Transport-MQAR, a finite-field variant of multi-query associative recall (MQAR), a common diagnostic for recall in efficient sequence models (Arora et al., 2024; Huang et al., 2025). Unlike standard associative recall, Transport-MQAR inserts causal operations between bindings and queries, so a model must recover a value in the current transported frame.

4.1. Diagnostic design and controls

Each example is generated over $\mathbb{F}_{31}$ with 256 keys and four-coordinate values. A binding event stores a key–value pair; an operation event applies an invertible right action to all stored values; and a query event asks for the currently transported value of a key. The model consumes the sequence causally and is trained with coordinate-wise cross-entropy at query positions. We report coordinate accuracy, the fraction of target coordinates predicted correctly, and exact accuracy, the fraction of queries for which all four coordinates are correct. The configured training length is 512, and final evaluations use lengths $128, 512, 2048, 4096$; the out-of-distribution lengths are not used for model selection. Appendices D.1–D.2 specify the generator, token layout, nonreducible operation library, hyperparameters, and evaluation protocol.

We compare StructGen-Split to controls that isolate different mechanisms. GRU and Transformer are same-width sequence baselines (Cho et al., 2014; Vaswani et al., 2017). Free enc/dec adds static input/output basis flexibility around a no-right SHiPPO-style memory. No-right preserves the matrix-state geometry but fixes $R_t = I$, so it is a right-action pathway ablation rather than a parameter-matched ablation.

Within the same scan-compatible SHiPPO-style family, DirectGen-SingleExp provides a direct-generator single-exponential right-action realization. It keeps the group-local matrix-state recurrence

$$H_{t,g} = L_{t,g} H_{t-1,g} R_{t,g} + \widehat{U}_{t,g},$$

but replaces StructGen-Split's structured generator library

and split product by a directly emitted group-local generator $A_{t,g} \in \mathbb{R}^{P_g \times P_g}$ and a single matrix exponential,

$$R_{t,g} = \exp(\Delta_{t,g} A_{t,g}).$$

This intra-family control tests whether the structured generator library and split-product backend provide a useful bias beyond a direct group-local right-action parameterization. Appendix D.3 gives the exact model geometry, and Appendix D.6 reports matched-capacity stress tests.

4.2. Main diagnostic result

Table 1 has two main diagnostic implications. First, StructGen-Split improves coordinate-wise transported recovery over no-right and static-basis controls at both the configured length and the longest extrapolation length. At length 4096, for example, coordinate accuracy rises from 0.102 for no-right to 0.110 for StructGen-Split. This is consistent with the right-action pathway helping recover coordinates in the transported frame.

Second, the results leave the choice of right-action realization open. DirectGen-SingleExp, which directly emits a group-local right generator and applies a single matrix exponential inside the same SHiPPO-style recurrence, achieves the strongest exact accuracy in the primary comparison. The matched DirectGen-SingleExp stress test in Appendix D.6 reaches Coord@4096$=0.110$ and Exact@4096$=0.034$, making it competitive with StructGen-Split in raw diagnostic accuracy. Thus the diagnostic supports the usefulness of right transport while indicating that the structured-generator split backend is one realization choice rather than a uniquely preferred discretization.

4.3. Controller-suffix intervention

To test whether the trained StructGen-Split model uses the additional controller coordinates associated with split-flow right transport, we perform an evaluation-time counterfactual. In the reported configuration, the StructGen-Split controller emits 5248 scalars per token, whereas the no-right controller emits 4224. We split

$$\theta_t = (\theta_t^{\text{shared}}, \theta_t^{\text{extra}}), \qquad \theta_t^{\text{extra}} \in \mathbb{R}^{1024}, \tag{6}$$

and zero θ_t^{extra} at evaluation time while keeping all trained weights fixed. No retraining is performed.

Table 2 shows that zeroing the suffix degrades both coordinate and exact accuracy. At length 4096, exact accuracy drops from 0.0329 to 0.0176. This intervention indicates that the trained predictor depends on the additional controller coordinates associated with the right-action pathway. It does not, by itself, identify the learned transport geometry or prove that the model implements the intended noncommutative mechanism. Appendix D.7 reports controller norms and full-length counterfactuals.

4.4. Empirical scope

The empirical evidence in this paper is synthetic and diagnostic. Transport-MQAR isolates transported coordinate recovery under length extrapolation; the byte-level FineWeb-Edu training, RULER/lm-eval adapters, and backend checks (Penedo et al., 2024; Hsieh et al., 2024; Biderman et al., 2024) are reported in Appendices D.10–D.12 only as stability and readiness notes, not as benchmark evidence. The DirectGen-SingleExp comparison should be read as an intra-family comparison between right-action realizations. Because DirectGen-SingleExp differs from StructGen-Split in both the generator family and the backend, the current results do not isolate whether the observed differences come from generator structure, split factorization, or their interaction. The main empirical claim is restricted to Table 1, Table 2, and the supporting Transport-MQAR diagnostics in Appendix D.

5. Limitations and Questions for Discussion

Technical scope. The strongest claim of SHiPPO is the operator-level claim in Section 2: conditional on a fixed or realized right-transport path, the state is an online projection coefficient matrix in a tied moving channel frame. This does not imply that arbitrary matrix-valued recurrences or independent encoder–decoder wrappers around HiPPO have projection-memory semantics. In SHiPPO, the metric, approximation family, history encoder, coefficient decoder, and Sylvester gauge term are tied by the same transport path. The selective cell in Section 3 is a restricted computational realization: its group-tied diagonal left dynamics, controller-compatible group-local right actions, and chosen right-action parameterization are introduced to preserve exact block-affine scan and recurrent decoding. Thus Section 3 should be read as one scan-compatible realization of the operator-level SHiPPO prior, not as a characterization of all possible transported-projection layers.

The StructGen-Split cell is not the operator-level SHiPPO prior itself. For a chosen step-frozen right generator, a single matrix exponential is the canonical frozen-flow right-action discretization of the restricted ODE, whereas the split backend forms the right action as a product of exponentials of structured generator factors. Thus the current StructGen-Split realization imposes both a structured generator family and a split backend.

Empirical scope. Our empirical evidence is synthetic and diagnostic. Transport-MQAR isolates transported coordinate recovery under length extrapolation. The main empirical result is deliberately modest: StructGen-Split improves coordinate accuracy over structural controls in Table 1, and the controller-suffix intervention in Table 2 shows that the trained predictor depends on the right-action controller path-

Table 1. Primary Transport-MQAR diagnostic. "StructGen-Split" denotes the SHiPPO-style realization with a structured right-generator library and a split product of factor exponentials. "DirectGen-SingleExp" denotes the intra-family realization with a directly emitted group-local right generator and a single matrix-exponential right action. Means over $n = 3$ seeds are shown; full length sweeps and standard deviations are in Appendix D.5.

Model	Params	Coord@512	Coord@4096	Exact@512	Exact@4096
GRU	0.50M	0.115	0.095	0.024	0.016
Transformer	0.89M	0.099	0.042	0.017	0.002
Free enc/dec	5.11M	0.119	0.092	0.021	0.009
No-right	4.98M	0.124	0.102	0.024	0.016
DirectGen-SingleExp	1.52M	0.125	0.103	0.047	0.036
StructGen-Split	6.03M	0.135	0.110	0.044	0.033

Table 2. Evaluation-time controller-suffix intervention for StructGen-Split. "Change" is zeroed minus normal; means over $n = 3$ seeds are shown.

Setting	Coord@512	Exact@512	Coord@4096	Exact@4096
StructGen-Split	0.1346	0.0439	0.1104	0.0329
Zeroed suffix	0.1250	0.0239	0.1044	0.0176
Change	-0.0096	-0.0200	-0.0060	-0.0154

way. These results do not identify the learned transport geometry or prove a noncommutative algorithm inside the network. DirectGen-SingleExp is stronger on exact accuracy in the primary comparison and competitive under capacity matching, indicating that Transport-MQAR is sensitive not only to the presence of right transport but also to how the right generator and discrete right action are parameterized. This is consistent with recent work showing that recall, state tracking, input selectivity, and length generalization can depend sharply on the chosen recurrent structure and diagnostic task (Merrill et al., 2024; Terzić et al., 2025a; Huang et al., 2025). The byte-level language-modeling and external-benchmark adapters in Appendix D.10 are readiness checks only.

Future diagnostic variants could stratify examples by operation depth, held-out operation words, and finite-training-set reuse, and future realization studies could isolate the generator family from the discrete right-action backend under matched capacity.

Questions for discussion. These results raise three broader questions for transported memory priors in generative modeling. First, to what extent do pathwise projection semantics persist in deep stacks with residual streams, gating, normalization, and learned controllers, and when should they instead be interpreted as an architectural bias rather than an exact memory semantics? Second, how far can transported-memory priors move beyond autoregressive recall diagnostics, for example to latent-variable, flow, or diffusion models where the relevant "history" is not simply a token prefix? Third, what real generative tasks would most directly test whether models generalize by updating structured bindings rather than memorizing fixed-coordinate associations?

SHiPPO should therefore be read as a structured memory prior, with Transport-MQAR and the controller-suffix intervention providing a diagnostic protocol for transported generalization. Together, they expose a concrete mechanism and a controlled success/failure pattern, rather than closing the question of which recurrent architecture is best for deep generative models.

Acknowledgements

This work was supported by JSPS KAKENHI Grant Number JP26K21295. This research used resources of the Argonne Leadership Computing Facility under ALCF Allocation ID 15652, 15654 and resources of the Oak Ridge Leadership Computing Facility under OLCF Project ID CSC704 through Director's Discretionary allocation awards.

Impact Statement

This paper presents foundational work on recurrent memory priors for autoregressive sequence models. The empirical evaluation is primarily synthetic and diagnostic, and we do not release a large pretrained generative model or claim deployment-ready language-modeling performance. Potential positive impacts include better tools for understanding memorization, structured generalization, and memory mechanisms in generative models. Potential negative impacts are indirect and similar to those of improved sequence-modeling methods more broadly, including possible use in more capable generative systems; we therefore emphasize diagnostic scope, transparent limitations, and reproducibility.

References

Arora, S., Eyuboglu, S., Timalsina, A., Johnson, I., Poli, M., Zou, J., Rudra, A., and Ré, C. Zoology: Measuring and improving recall in efficient language models. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=LY3ukUANko.

Beck, M., Pöppel, K., Spanring, M., Auer, A., Prudnikova, O., Kopp, M., Klambauer, G., Brandstetter, J., and Hochreiter, S. xLSTM: Extended long short-term memory. In *Advances in Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=ARAxPPIAhq.

Behr, M., Benner, P., and Heiland, J. Solution formulas for differential sylvester and lyapunov equations. *Calcolo*, 56(51), 2019. doi: 10.1007/s10092-019-0348-x.

Biderman, S., Schoelkopf, H., Sutawika, L., Gao, L., Tow, J., Abbasi, B., Aji, A. F., Ammanamanchi, P. S., Black, S., Clive, J., DiPofi, A., Etxaniz, J., Fattori, B., Forde, J. Z., Foster, C., Hsu, J., Jaiswal, M., Lee, W. Y., Li, H., Lovering, C., Muennighoff, N., Pavlick, E., Phang, J., Skowron, A., Tan, S., Tang, X., Wang, K. A., Winata, G. I., Yvon, F., and Zou, A. Lessons from the trenches on reproducible evaluation of language models, 2024. URL https://arxiv.org/abs/2405.14782.

Blelloch, G. E. Prefix sums and their applications. Technical Report CMU-CS-90-190, School of Computer Science, Carnegie Mellon University, 1990. URL https://www.cs.cmu.edu/~scandal/papers/CMU-CS-90-190.html.

Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pp. 1724–1734, 2014. doi: 10.3115/v1/D14-1179. URL https://aclanthology.org/D14-1179/.

Clark, P., Cowhey, I., Etzioni, O., Khot, T., Sabharwal, A., Schoenick, C., and Tafjord, O. Think you have solved question answering? try ARC, the AI2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018. URL https://arxiv.org/abs/1803.05457.

Coddington, E. A. and Levinson, N. *Theory of Ordinary Differential Equations*. McGraw-Hill, New York, 1955.

Dao, T. and Gu, A. Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 10041–10071. PMLR, 2024. URL https://proceedings.mlr.press/v235/dao24a.html.

EleutherAI. Language model evaluation harness. GitHub repository, 2024. URL https://github.com/EleutherAI/lm-evaluation-harness. MIT license.

Fortuin, V. Priors in bayesian deep learning: A review. *International Statistical Review*, 90(3):563–591, 2022. doi: 10.1111/insr.12502.

Fu, D. Y., Dao, T., Saab, K. K., Thomas, A. W., Rudra, A., and Ré, C. Hungry hungry hippos: Towards language modeling with state space models. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=COZDy0WYGg.

Goffinet, J., Hanks, C., and Carlson, D. E. HiPPO Zoo: Explicit memory mechanisms for interpretable state space models, 2026. URL https://arxiv.org/abs/2602.21340.

Gordon, D. F. and desJardins, M. Evaluation and selection of biases in machine learning. *Machine Learning*, 20 (1–2):5–22, 1995. doi: 10.1023/A:1022630017346.

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

Gu, A., Dao, T., Ermon, S., Rudra, A., and Ré, C. HiPPO: Recurrent memory with optimal polynomial projections. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/102f0bb6efb3a6128a3c750dd16729be-Abstract.html.

Gu, A., Goel, K., and Ré, C. Efficiently modeling long sequences with structured state spaces. In *International Conference on Learning Representations*, 2022a. URL https://openreview.net/forum?id=uYLFoz1vlAC.

Gu, A., Gupta, A., Goel, K., and Ré, C. On the parameterization and initialization of diagonal state space models. In *Advances in Neural Information Processing Systems*, 2022b. URL https://papers.nips.cc/paper_files/paper/2022/hash/e9a32fade47b906de908431991440f7c-Abstract-Conference.html.

Gu, A., Johnson, I., Timalsina, A., Rudra, A., and Ré, C. How to train your HiPPO: State space models with generalized orthogonal basis projections. In *International*

Conference on Learning Representations, 2023. URL https://openreview.net/forum?id=klK17OQ3KB.

Gupta, A., Gu, A., and Berant, J. Diagonal state spaces are as effective as structured state spaces. In *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=RjS0j6tsSrf.

Hairer, E., Lubich, C., and Wanner, G. *Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations*. Springer, 2 edition, 2006. doi: 10.1007/3-540-30666-8.

Higham, N. J. *Functions of Matrices: Theory and Computation*. SIAM, 2008. doi: 10.1137/1.9780898717778.

Hsieh, C.-P., Sun, S., Kriman, S., Acharya, S., Rekesh, D., Jia, F., Zhang, Y., and Ginsburg, B. RULER: What's the real context size of your long-context language models? In *Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=kIoBbc76Sy.

Huang, N. T., Sarabia, M., Moudgil, A., Rodriguez, P., Zappella, L., and Danieli, F. Understanding input selectivity in Mamba: Impact on approximation power, memorization, and associative recall capacity. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 25693–25727. PMLR, 2025. URL https://proceedings.mlr.press/v267/huang25ab.html.

Hugging Face FineWeb Team. FineWeb-Edu: Dataset card. Hugging Face Datasets, 2024. URL https://huggingface.co/datasets/HuggingFaceFW/fineweb-edu. Released under ODC-By v1.0; subject to CommonCrawl terms of use.

Kimi Team. Kimi Linear: An expressive, efficient attention architecture, 2025. URL https://arxiv.org/abs/2510.26692.

Lahoti, A., Li, K. Y., Chen, B., Wang, C., Bick, A., Kolter, J. Z., Dao, T., and Gu, A. Mamba-3: Improved sequence modeling using state space principles. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=HwCvaJOiCj.

Liu, B., Wang, R., Wu, L., Feng, Y., Stone, P., and Liu, Q. Longhorn: State space models are amortized online learners. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=8jOqCcLzeO.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Martin, E. and Cundy, C. Parallelizing linear recurrent neural nets over sequence length. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=HyUNwulC-.

Merrill, W., Petty, J., and Sabharwal, A. The illusion of state in state-space models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*. PMLR, 2024. URL https://proceedings.mlr.press/v235/merrill24a.html.

Mishra, M., Tan, S., Stoica, I., Gonzalez, J., and Dao, T. M^2RNN: Non-linear RNNs with matrix-valued states for scalable language modeling, 2026. URL https://arxiv.org/abs/2603.14360.

Mitchell, T. M. The need for biases in learning generalizations. Technical Report CBM-TR-117, Department of Computer Science, Rutgers University, 1980. URL https://www.cs.cmu.edu/~tom/pubs/NeedForBias_1980.pdf.

Movahedi, S., Sarnthein, F., Cirone, N. M., and Orvieto, A. Fixed-point RNNs: From diagonal to dense in a few iterations, 2025. URL https://arxiv.org/abs/2503.10799.

NVIDIA. RULER: Source code repository. GitHub repository, 2024. URL https://github.com/NVIDIA/RULER. Apache-2.0 license.

Penedo, G., Kydlíček, H., Ben Allal, L., Lozhkov, A., Mitchell, M., Raffel, C., von Werra, L., and Wolf, T. The FineWeb datasets: Decanting the web for the finest text data at scale. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, 2024. URL https://arxiv.org/abs/2406.17557.

Qin, Z., Yang, S., Sun, W., Shen, X., Li, D., Sun, W., and Zhong, Y. HGRN2: Gated linear RNNs with state expansion. In *The First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=y6SqbJfCSk.

Sarrof, Y., Veitsman, Y., and Hahn, M. The expressive capacity of state space models: A formal language perspective. In *Advances in Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=eV5YIrJPdy.

Schlag, I., Irie, K., and Schmidhuber, J. Linear transformers are secretly fast weight programmers. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 9355–9366. PMLR, 2021. URL https://proceedings.mlr.press/v139/schlag21a.html.

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

Siems, J., Carstensen, T., Zela, A., Hutter, F., Pontil, M., and Grazzi, R. DeltaProduct: Increasing the expressivity of DeltaNet through products of householders, 2025. URL https://arxiv.org/abs/2502.10297.

Simoncini, V. Computational methods for linear matrix equations. *SIAM Review*, 58(3):377–441, 2016. doi: 10.1137/130912839.

Smith, J. T. H., Warrington, A., and Linderman, S. W. Simplified state space layers for sequence modeling. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=Ai8Hw3AXqks.

Sun, Y., Li, X., Dalal, K., Xu, J., Vikram, A., Zhang, G., Dubois, Y., Chen, X., Wang, X., Koyejo, S., Hashimoto, T., and Guestrin, C. Learning to (Learn at test time): RNNs with expressive hidden states. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 57503–57522. PMLR, 2025. URL https://proceedings.mlr.press/v267/sun25h.html.

Terzić, A., Hersche, M., Camposampiero, G., Hofmann, T., Sebastian, A., and Rahimi, A. On the expressiveness and length generalization of selective state-space models on regular languages. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 2025a. URL https://ojs.aaai.org/index.php/AAAI/article/view/34301.

Terzić, A., Menet, N., Hersche, M., Hofmann, T., and Rahimi, A. Structured sparse transition matrices to enable state tracking in state-space models. In *Advances in Neural Information Processing Systems*, 2025b. URL https://openreview.net/forum?id=RDbuSCWhad.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://papers.neurips.cc/paper/7181-attention-is-all-you-need.

Voelker, A. R., Kajić, I., and Eliasmith, C. Legendre memory units: Continuous-time representation in recurrent neural networks. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/952285b9b7e7a1be5aa7849f32ffff05-Abstract.html.

von Oswald, J., Scherrer, N., Kobayashi, S., Versari, L., Yang, S., Schlegel, M., Maile, K., Schimpf, Y., Sieberling, O., Meulemans, A., Lajoie, G., Saurous, R. A., Frenkel, C., Pascanu, R., Agüera y Arcas, B., and Sacramento, J. MesaNet: Sequence modeling by locally optimal test-time training. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=xa3OnTb6c3.

Yang, S., Wang, B., Shen, Y., Panda, R., and Kim, Y. Gated linear attention transformers with hardware-efficient training. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 56501–56523. PMLR, 2024a. URL https://proceedings.mlr.press/v235/yang24ab.html.

Yang, S., Wang, B., Zhang, Y., Shen, Y., and Kim, Y. Parallelizing linear transformers with the delta rule over sequence length. In *Advances in Neural Information Processing Systems*, 2024b. URL https://openreview.net/forum?id=y8Rm4VNRPH.

Yang, S., Kautz, J., and Hatamizadeh, A. Gated delta networks: Improving Mamba2 with delta rule. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=r8H7xhYPwz.

A. Derivations for SHiPPO

This appendix supports the operator-level claims in Section 2. It recalls the ordinary vector-valued HiPPO variational equations, derives the SHiPPO normal equation directly, proves the transport-isometry conjugacy and Sylvester dynamics, and explains why modifying only the channel metric does not generally preserve finite HiPPO-style closure. The derivations are pathwise: when the right generator is produced by a causal controller, all statements below are conditioned on the realized controller trajectory.

A.1. Notation, assumptions, and pathwise semantics

Signals are column vectors in $\mathbb{R}^d$. The basis $\Phi_t(\tau) \in \mathbb{R}^N$ is also a column vector, and coefficient matrices are $C \in \mathbb{R}^{N\times d}$, so that $C^\top \Phi_t(\tau) \in \mathbb{R}^d$. We use the Frobenius inner product $\langle A, B\rangle_F = \mathrm{tr}(A^\top B)$. For each t, let μ_t be a measure on $[0, t]$, and define

$$G(t) = \int_0^t \Phi_t(\tau)\Phi_t(\tau)^\top \, d\mu_t(\tau). \tag{7}$$

We assume that $G(t)$ is invertible. When $d\mu_t(\tau) = w_t(\tau)d\tau$, write $\psi(t, \tau) = w_t(\tau)\Phi_t(\tau)$, and assume the usual HiPPO closure condition (Gu et al., 2020; 2023)

$$\partial_t \psi(t, \tau) = A_L(t)\psi(t, \tau) \quad (\tau < t), \qquad B_L(t) = \psi(t, t). \tag{8}$$

We also assume the regularity needed to exchange first variations and integrals and to apply Leibniz' rule. Under weaker hypotheses, the displayed ODEs are understood at differentiability times, or almost everywhere.

For SHiPPO, fix an admissible right-generator path $A_R \in L^1([0, T]; \mathbb{R}^{d\times d})$. Let $P(t, \tau)$ be the right state transition

$$\partial_t P(t, \tau) = P(t, \tau)A_R(t), \qquad P(\tau, \tau) = I_d. \tag{9}$$

Standard linear ODE theory gives $P(t, \tau) \in \mathrm{GL}(d)$ for $0 \le \tau \le t \le T$ (Coddington & Levinson, 1955). SHiPPO is defined relative to this chosen path. The inner projection minimizes over C only; learning or parameterizing A_R is an outer modeling choice.

A.2. Ordinary vector-valued HiPPO

For ordinary HiPPO, the approximation family is $\{\tau \mapsto C^\top \Phi_t(\tau) : C \in \mathbb{R}^{N\times d}\}$, and the objective is

$$J_t^H(C) = \int_0^t \|f(\tau) - C^\top \Phi_t(\tau)\|_2^2 \, d\mu_t(\tau). \tag{10}$$

Proposition A.1 (Ordinary normal equation)**.** *Any minimizer $C_H(t)$ of Eq. 10 satisfies*

$$G(t)C_H(t) = \int_0^t \Phi_t(\tau)f(\tau)^\top \, d\mu_t(\tau). \tag{11}$$

If $G(t)$ is invertible, the minimizer is unique.

Proof. Fix t, abbreviate $\Phi = \Phi_t$, and set $e(\tau; C) = f(\tau) - C^\top \Phi(\tau)$. For a perturbation $C + \varepsilon\Delta$, we have $\delta e = -\Delta^\top \Phi$. Hence

$$\begin{aligned}\delta J_t^H(C; \Delta) &= 2\int_0^t (\delta e)^\top e \, d\mu_t = -2\int_0^t (\Delta^\top \Phi)^\top e \, d\mu_t \\ &= -2\,\mathrm{tr}\left[\Delta^\top \int_0^t \Phi(\tau)e(\tau; C)^\top \, d\mu_t(\tau)\right].\end{aligned}$$

Stationarity for all Δ gives $\int_0^t \Phi e^\top d\mu_t = 0$. Expanding $e^\top = f^\top - \Phi^\top C$ yields Eq. 11. Since the quadratic has Hessian $G(t) \otimes I_d$, invertibility of $G(t)$ gives strict convexity and uniqueness. □

Proposition A.2 (Ordinary coefficient dynamics)**.** *Assume $G(t) = I_N$, $d\mu_t(\tau) = w_t(\tau)d\tau$, and Eq. 8. Then*

$$C_H(t) = \int_0^t \psi(t,\tau) f(\tau)^\top \, d\tau, \tag{12}$$

and

$$\dot{C}_H(t) = A_L(t) C_H(t) + B_L(t) f(t)^\top. \tag{13}$$

Proof. With $G(t) = I_N$, Proposition A.1 gives Eq. 12. Differentiating by Leibniz' rule,

$$\begin{aligned}\dot{C}_H(t) &= \psi(t,t) f(t)^\top + \int_0^t \partial_t \psi(t,\tau) f(\tau)^\top \, d\tau \\ &= B_L(t) f(t)^\top + A_L(t) \int_0^t \psi(t,\tau) f(\tau)^\top \, d\tau,\end{aligned}$$

which is Eq. 13. □

If $G(t) \neq I_N$, define $b_H(t) = \int_0^t \Phi_t(\tau) f(\tau)^\top d\mu_t(\tau)$. Then $C_H(t) = G(t)^{-1} b_H(t)$ and

$$\dot{C}_H(t) = G(t)^{-1} \dot{b}_H(t) - G(t)^{-1} \dot{G}(t) C_H(t), \tag{14}$$

whenever the derivatives exist. The main text uses the orthonormalized form because it exposes the clean HiPPO coefficient ODE.

A.3. Stationarity of the SHiPPO objective

For the path $P(t,\tau)$, define the transported metric

$$M_P(t,\tau) = P(t,\tau) P(t,\tau)^\top. \tag{15}$$

The SHiPPO objective at time t is

$$J_t^{SH}(C) = \int_0^t \left\| f(\tau) - P(t,\tau)^{-\top} C^\top \Phi_t(\tau) \right\|^2_{M_P(t,\tau)} \, d\mu_t(\tau). \tag{16}$$

Proposition A.3 (SHiPPO normal equation)**.** *Any minimizer $C_S(t)$ of Eq. 16 satisfies*

$$G(t) C_S(t) = \int_0^t \Phi_t(\tau) f(\tau)^\top P(t,\tau) \, d\mu_t(\tau). \tag{17}$$

If $G(t)$ is invertible, the minimizer is unique.

Proof. Fix t, and abbreviate $P = P(t,\tau)$, $M_P = PP^\top$, and $\Phi = \Phi_t(\tau)$. Let $e_S(\tau; C) = f(\tau) - P^{-\top} C^\top \Phi$. Under $C + \varepsilon\Delta$, $\delta e_S = -P^{-\top} \Delta^\top \Phi$. Therefore

$$\begin{aligned}\delta J_t^{SH}(C;\Delta) &= 2 \int_0^t (\delta e_S)^\top M_P e_S \, d\mu_t \\ &= -2 \int_0^t (\Delta^\top \Phi)^\top P^{-1} M_P e_S \, d\mu_t \\ &= -2 \int_0^t (\Delta^\top \Phi)^\top P^\top e_S \, d\mu_t \\ &= -2 \operatorname{tr}\left[\Delta^\top \int_0^t \Phi(\tau) e_S(\tau; C)^\top P(t,\tau) \, d\mu_t(\tau) \right].\end{aligned}$$

Stationarity for all Δ gives $\int_0^t \Phi e_S^\top P \, d\mu_t = 0$. Since

$$e_S(\tau; C)^\top P(t,\tau) = f(\tau)^\top P(t,\tau) - \Phi_t(\tau)^\top C,$$

we obtain Eq. 17. Strict convexity follows from invertibility of $G(t)$ and $P(t,\tau) \in \mathrm{GL}(d)$, so the minimizer is unique. □

A.4. Transport isometry and conjugacy to ordinary HiPPO

For fixed t, define

$$(\mathcal{T}_t u)(\tau) = P(t,\tau)^\top u(\tau), \qquad (\mathcal{T}_t^{-1} u)(\tau) = P(t,\tau)^{-\top} u(\tau). \tag{18}$$

Let $\langle u, v\rangle_{t,M_P} = \int_0^t u(\tau)^\top M_P(t,\tau) v(\tau)\, d\mu_t(\tau)$ and let $\langle \cdot,\cdot\rangle_{t,I}$ denote the Euclidean-channel inner product.

Proposition A.4 (Transport isometry and SHiPPO–HiPPO conjugacy)**.** *For all admissible u, v,*

$$\langle u, v\rangle_{t,M_P} = \langle \mathcal{T}_t u, \mathcal{T}_t v\rangle_{t,I}. \tag{19}$$

Moreover, $\mathcal{T}_t$ maps the SHiPPO approximation family bijectively onto ordinary HiPPO's approximation family, and

$$\mathrm{shippo}_t(f) = \mathrm{hippo}_t(\mathcal{T}_t f). \tag{20}$$

Proof. The isometry follows pointwise:

$$u^\top P P^\top v = (P^\top u)^\top (P^\top v).$$

For any coefficient matrix C,

$$\mathcal{T}_t\big(P(t,\cdot)^{-\top} C^\top \Phi_t(\cdot)\big) = C^\top \Phi_t(\cdot).$$

Because $P(t,\tau)$ is invertible, this correspondence between approximation families is bijective. Minimizing $\|f-g\|_{t,M_P}$ over transported approximants g is therefore equivalent, by Eq. 19, to minimizing $\|\mathcal{T}_t f - \tilde{g}\|_{t,I}$ over ordinary HiPPO approximants $\tilde{g} = \mathcal{T}_t g$. The same coefficient matrix C indexes both families, giving Eq. 20. □

Proposition A.4 gives a conceptual proof of Proposition A.3: apply Proposition A.1 to the transformed signal $(\mathcal{T}_t f)(\tau) = P(t,\tau)^\top f(\tau)$. It also explains why the metric and approximation family must be transported together: the construction is an isometric change of frame for the online projection problem.

A.5. Proof of the Sylvester dynamics and pathwise lift

We now prove the dynamics stated in Theorem 2.2. The normal equation is Proposition A.3. Under the orthonormalized assumptions $G(t) = I_N$ and $d\mu_t(\tau) = w_t(\tau)d\tau$, it becomes

$$C_S(t) = \int_0^t \psi(t,\tau) f(\tau)^\top P(t,\tau)\, d\tau. \tag{21}$$

Differentiating Eq. 21 gives

$$\begin{aligned}\dot{C}_S(t) = \psi(t,t) f(t)^\top P(t,t) &+ \int_0^t \partial_t \psi(t,\tau) f(\tau)^\top P(t,\tau)\, d\tau \\ &+ \int_0^t \psi(t,\tau) f(\tau)^\top \partial_t P(t,\tau)\, d\tau.\end{aligned}$$

Using $P(t,t) = I_d$, $\psi(t,t) = B_L(t)$, the closure condition $\partial_t \psi = A_L \psi$, and the transport equation $\partial_t P = P A_R$, the three terms become

$$B_L(t) f(t)^\top, \qquad A_L(t) C_S(t), \qquad C_S(t) A_R(t),$$

respectively. Hence

$$\dot{C}_S(t) = A_L(t) C_S(t) + B_L(t) f(t)^\top + C_S(t) A_R(t). \tag{22}$$

This is the differential Sylvester form used in the main text (Behr et al., 2019; Simoncini, 2016).

The same calculation proves the pathwise lift statement. Any one-sided coefficient equation derived from the projection setup and satisfying

$$\dot{C}(t) = A_L(t) C(t) + B_L(t) f(t)^\top$$

can be lifted, for any admissible realized right path A_R, by replacing the ordinary normal equation with Eq. 17. The left-memory term and boundary injection are unchanged, while differentiating the right transition contributes $C_S(t) A_R(t)$. This statement is about projection-derived coefficient equations, not arbitrary matrix ODEs.

For non-orthonormal bases, define

$$b_S(t) = \int_0^t \Phi_t(\tau) f(\tau)^\top P(t,\tau)\, d\mu_t(\tau).$$

Then $C_S(t) = G(t)^{-1} b_S(t)$, and

$$\dot{C}_S(t) = G(t)^{-1}\dot{b}_S(t) - G(t)^{-1}\dot{G}(t) C_S(t). \tag{23}$$

Thus the clean Sylvester display is the orthonormalized presentation, matching the corresponding ordinary HiPPO form.

A.6. Why metric-only modification does not generally close

This subsection records the nearby construction that SHiPPO avoids. Keep the ordinary approximation family $C^\top \Phi_t(\tau)$, but replace the Euclidean channel metric by a symmetric positive definite matrix $M(t,\tau)$:

$$J_t^M(C) = \int_0^t \big(f(\tau) - C^\top \Phi_t(\tau)\big)^\top M(t,\tau) \big(f(\tau) - C^\top \Phi_t(\tau)\big)\, d\mu_t(\tau). \tag{24}$$

Proposition A.5 (Metric-only stationarity)**.** *Any minimizer $C_M(t)$ of Eq. 24 satisfies*

$$\int_0^t \Phi_t(\tau) f(\tau)^\top M(t,\tau)\, d\mu_t(\tau) = \int_0^t \Phi_t(\tau)\Phi_t(\tau)^\top C_M(t) M(t,\tau)\, d\mu_t(\tau). \tag{25}$$

Proof. Let $e(\tau; C) = f(\tau) - C^\top \Phi_t(\tau)$. Since $\delta e = -\Delta^\top \Phi_t$ and $M(t,\tau)$ is symmetric,

$$\begin{aligned}\delta J_t^M(C;\Delta) &= -2\int_0^t (\Delta^\top \Phi_t(\tau))^\top M(t,\tau) e(\tau; C)\, d\mu_t(\tau) \\ &= -2\,\mathrm{tr}\left[\Delta^\top \int_0^t \Phi_t(\tau)(M(t,\tau)e(\tau;C))^\top\, d\mu_t(\tau)\right].\end{aligned}$$

Stationarity gives $\int_0^t \Phi_t (Me)^\top d\mu_t = 0$. Expanding $e = f - C^\top \Phi_t$ yields Eq. 25. □

Equivalently, if

$$K_{M,t}[C] := \int_0^t \Phi_t(\tau)\Phi_t(\tau)^\top C M(t,\tau)\, d\mu_t(\tau), \quad b_{M,t} := \int_0^t \Phi_t(\tau) f(\tau)^\top M(t,\tau)\, d\mu_t(\tau),$$

then $C_M(t)$ solves $K_{M,t}[C_M(t)] = b_{M,t}$. If $M(t,\tau) = M(t)$ is independent of τ, this reduces to

$$G(t) C_M(t) M(t) = \left(\int_0^t \Phi_t(\tau) f(\tau)^\top d\mu_t(\tau)\right) M(t),$$

and invertibility of $M(t)$ recovers the ordinary HiPPO normal equation. When $M(t,\tau)$ genuinely depends on τ, however, $K_{M,t}$ does not factor as $C \mapsto G(t)CM(t)$. A finite closed coefficient ODE would then require solving a time-varying operator equation or augmenting the state with additional metric-dependent moments.

SHiPPO avoids this obstruction by coupling the metric $P(t,\tau)P(t,\tau)^\top$ with the transported family $P(t,\tau)^{-\top} C^\top \Phi_t(\tau)$. The factors cancel in the first variation, producing Eq. 17 and hence the Sylvester coefficient dynamics.

B. Moving-Frame Interpretation and Non-Reduction

This appendix supports the reduction and moving-frame claims at the end of Section 2. Fix an admissible path $A_R \in L^1([0,T]; \mathbb{R}^{d\times d})$. All statements are exact for this chosen path, and therefore apply pathwise when A_R is produced by a causal controller. The appendix is not about learning or optimizing A_R; it only records the algebra forced by the transported approximation problem. Since A_R is assumed integrable, differential statements hold almost everywhere.

B.1. Transport identities

For $0 \le \tau \le t \le T$, let $P(t,\tau)$ solve

$$\partial_t P(t,\tau) = P(t,\tau)A_R(t), \qquad P(\tau,\tau) = I_d. \tag{26}$$

Standard linear ODE theory gives a unique absolutely continuous transition family (Coddington & Levinson, 1955).

Lemma B.1 (Invertibility and composition)**.** *For all $0 \le \tau \le \sigma \le t \le T$, $P(t,\tau) \in GL(d)$,*

$$\partial_t P(t,\tau)^{-1} = -A_R(t)P(t,\tau)^{-1}, \tag{27}$$

for a.e. t, and

$$P(t,\tau) = P(\sigma,\tau)P(t,\sigma). \tag{28}$$

Moreover,

$$\det P(t,\tau) = \exp\left(\int_\tau^t \operatorname{tr} A_R(s)\, ds\right) \neq 0. \tag{29}$$

Proof. Let $Y(t,\tau)$ solve $\partial_t Y = -A_R(t)Y$, $Y(\tau,\tau) = I_d$. Then $\partial_t(PY) = PA_RY - PA_RY = 0$, so $P(t,\tau)Y(t,\tau) = I_d$. Since the matrices are square, $P(t,\tau)$ is invertible and $Y = P^{-1}$, which gives Eq. 27. Liouville's formula follows from Jacobi's identity applied after invertibility is established. For composition, fix $\tau \le \sigma$ and set $Q(t) = P(\sigma,\tau)P(t,\sigma)$. Then Q and $P(t,\tau)$ satisfy Eq. 26 on $[\sigma,T]$ with the same initial value $P(\sigma,\tau)$; uniqueness gives Eq. 28. □

The order in Eq. 28 reflects the right-action convention. If $H(t) = H(\tau)P(t,\tau)$, then

$$H(t) = H(\tau)P(\sigma,\tau)P(t,\sigma).$$

The earlier segment is multiplied first and the later segment second.

B.2. Exact reduction to ordinary HiPPO

Proposition B.2 (Reduction criterion)**.** *The following are equivalent:*

1. $A_R(t) = 0$ *for a.e.* $t \in [0,T]$*;*
2. $P(t,\tau) = I_d$ *for all* $0 \le \tau \le t \le T$*.*

In this case $M_P(t,\tau) = I_d$, $P(t,\tau)^{-\top}C^\top\Phi_t(\tau) = C^\top\Phi_t(\tau)$, and Definition 2.1 reduces exactly to ordinary vector-valued HiPPO (Gu et al., 2020; 2023).

Proof. If $A_R = 0$ a.e., the integral form of Eq. 26 gives $P(t,\tau) = I_d$. Conversely, if $P(t,\tau) = I_d$ for every $\tau \le t$, then taking $\tau = 0$ and differentiating gives $0 = \partial_t P(t,0) = P(t,0)A_R(t) = A_R(t)$ a.e. The final statement follows by substituting $P = I_d$ into the transported metric and approximation family. □

B.3. Identity metric is not identity transport

The transported metric is $M_P(t,\tau) = P(t,\tau)P(t,\tau)^\top$. The condition $M_P \equiv I_d$ rules out dilation but not rotation of the channel frame.

Proposition B.3 (Identity metric criterion)**.** *The following are equivalent:*

1. $M_P(t,\tau) = I_d$ *for all* $0 \le \tau \le t \le T$*;*
2. $A_R(t) + A_R(t)^\top = 0$ *for a.e.* $t \in [0,T]$*.*

Consequently, $M_P \equiv I_d$ does not imply $P \equiv I_d$ unless the skew transport is also trivial.

Proof. For fixed τ, absolute continuity gives, for a.e. $t \geq \tau$,

$$\partial_t M_P(t,\tau) = P(t,\tau)\big(A_R(t) + A_R(t)^\top\big)P(t,\tau)^\top. \tag{30}$$

If $A_R + A_R^\top = 0$ a.e., then $\partial_t M_P = 0$ and $M_P(\tau,\tau) = I_d$, so $M_P \equiv I_d$. Conversely, if $M_P(t,\tau) = I_d$ for all $\tau \leq t$, take $\tau = 0$ in Eq. 30. Since $P(t,0)$ is invertible by Lemma B.1, the middle factor must vanish a.e. □

Even under $M_P \equiv I_d$, the SHiPPO family is still transported: $\tau \mapsto P(t,\tau)^{-\top} C^\top \Phi_t(\tau)$. Equivalently, by Proposition A.4, SHiPPO is ordinary HiPPO applied to the moving-frame signal $(\mathcal{T}_t f)(\tau) = P(t,\tau)^\top f(\tau)$, not to the original signal. For instance, if

$$A_R = \omega \left(\begin{smallmatrix} 0 & -1 \\ 1 & 0 \end{smallmatrix}\right), \qquad \omega \neq 0,$$

then $A_R^\top = -A_R$, so $M_P \equiv I_2$, but

$$P(t,\tau) = \begin{pmatrix} \cos\omega(t-\tau) & -\sin\omega(t-\tau) \\ \sin\omega(t-\tau) & \cos\omega(t-\tau) \end{pmatrix}$$

is nontrivial for generic $t \neq \tau$. The Sylvester term $C_S A_R$ in Theorem 2.2 therefore remains present.

B.4. Moving-frame factorization

Let $V : [0,T] \to GL(d)$ be the fundamental matrix

$$\dot{V}(t) = V(t)A_R(t), \qquad V(0) = I_d. \tag{31}$$

Define the tied history encoder and coefficient decoder

$$(\mathcal{E}_V f)(\tau) = V(\tau)^{-\top} f(\tau), \qquad \mathcal{D}_{V,t}(C) = CV(t). \tag{32}$$

Lemma B.4 (Frame representation and fixed-mixing equivariance)**.** *For every $0 \leq \tau \leq t \leq T$,*

$$P(t,\tau) = V(\tau)^{-1}V(t), \qquad P(t,\tau)^\top = V(t)^\top V(\tau)^{-\top}. \tag{33}$$

Moreover, if $Q \in GL(d)$ is independent of τ and $(\mathcal{M}_Q u)(\tau) = Q^\top u(\tau)$, then

$$\mathrm{hippo}_t(\mathcal{M}_Q u) = \mathrm{hippo}_t(u)Q. \tag{34}$$

Proof. The first identity follows because $V(\tau)^{-1}V(t)$ satisfies Eq. 26 with initial value I_d at $t = \tau$. For Eq. 34, apply the ordinary normal equation: $G(t)C_u = \int \Phi_t u^\top d\mu_t$. Replacing u by $Q^\top u$ multiplies the right-hand side by Q on the right, and invertibility of $G(t)$ gives the claim. □

Proposition B.5 (Moving-frame factorization)**.** *For the chosen transport path,*

$$\mathrm{shippo}_t = \mathcal{D}_{V,t} \circ \mathrm{hippo}_t \circ \mathcal{E}_V. \tag{35}$$

Equivalently,

$$\mathrm{shippo}_t(f) = \mathrm{hippo}_t(\mathcal{E}_V f)V(t). \tag{36}$$

Proof. By Eq. 33, $(\mathcal{T}_t f)(\tau) = P(t,\tau)^\top f(\tau) = V(t)^\top(\mathcal{E}_V f)(\tau)$. Thus $\mathcal{T}_t = \mathcal{M}_{V(t)} \circ \mathcal{E}_V$. Combining Proposition A.4 with Eq. 34 yields

$$\mathrm{shippo}_t(f) = \mathrm{hippo}_t(\mathcal{T}_t f) = \mathrm{hippo}_t(\mathcal{E}_V f)V(t).$$

□

The encoder and decoder in Eq. 32 are tied by the same moving frame. Hence Eq. 35 is not an arbitrary free encoder–HiPPO–decoder factorization. If the transport is generated by an input-dependent controller, the same identity holds after conditioning on the realized trajectory, but the frame itself is input-dependent and therefore is not a fixed input-independent reduction.

B.5. Gauge equivalence of coefficient dynamics

Proposition B.6 (Gauge form of the Sylvester dynamics)**.** *Assume $G(t) = I_N$ and the HiPPO closure condition holds. Let $\widetilde{f}(t) = V(t)^{-\top} f(t)$, and let $\widetilde{C}$ be the ordinary HiPPO coefficient trajectory driven by $\widetilde{f}$:*

$$\dot{\widetilde{C}} = A_L \widetilde{C} + B_L \widetilde{f}^\top. \tag{37}$$

Then $C_S = \widetilde{C} V$ satisfies

$$\dot{C}_S = A_L C_S + B_L f^\top + C_S A_R. \tag{38}$$

Conversely, any solution of Eq. 38 gives an ordinary HiPPO trajectory $\widetilde{C} = C_S V^{-1}$ driven by $\widetilde{f}$.

Proof. Differentiate $C_S = \widetilde{C} V$:

$$\dot{C}_S = (A_L \widetilde{C} + B_L \widetilde{f}^\top) V + \widetilde{C} V A_R = A_L C_S + B_L (\widetilde{f}^\top V) + C_S A_R.$$

Since $\widetilde{f}^\top V = (V^{-\top} f)^\top V = f^\top$, this proves Eq. 38. Conversely, $\partial_t V^{-1} = -A_R V^{-1}$, so differentiating $\widetilde{C} = C_S V^{-1}$ cancels the $C_S A_R$ term and leaves $\dot{\widetilde{C}} = A_L \widetilde{C} + B_L f^\top V^{-1} = A_L \widetilde{C} + B_L \widetilde{f}^\top$. □

Propositions B.5 and B.6 show that the right-action term is exactly the gauge term induced by decoding ordinary HiPPO coefficients from a moving frame. They also delimit the non-reduction claim. SHiPPO is not a static mixer: if $V(t) \equiv V_0$, then $0 = \dot{V} = V_0 A_R$, hence $A_R = 0$ a.e. Nor is it a free encoder–decoder model: the encoder $V(\tau)^{-\top}$, decoder $V(t)$, metric $PP^\top$, and Sylvester term are all determined by the same transport path.

C. Proofs for the Scan-Compatible Realization

This appendix supports Section 3. It proves the direct-lift obstruction, the block-affine scan closure, the simultaneous-reducibility collapse criterion, and the discretization facts used in the scan-compatible SHiPPO cell. All results concern one transport group, so we omit the group index and write $H_t \in \mathbb{R}^{N\times P}$, $L_t \in \mathbb{R}^{N\times N}$, $R_t \in \mathrm{GL}(P)$, and $U_t \in \mathbb{R}^{N\times P}$. These group-local and controller-compatible restrictions are computational restrictions of the realization; they are not assumptions in the abstract SHiPPO projection problem.

C.1. Direct channelwise lift obstruction

A tempting direct lift of a diagonal selective recurrence would keep a separate left decay for every row and channel:

$$H_t^{\text{full}} = (\bar{A}_t \odot H_{t-1}^{\text{full}}) R_t + U_t, \qquad \bar{A}_t \in \mathbb{R}^{N\times P}. \tag{39}$$

Let $\bar{a}_{t,n} \in \mathbb{R}^P$ be row n of $\bar{A}_t$ and define $D_{t,n} = \mathrm{Diag}(\bar{a}_{t,n})$. Then row n evolves as

$$h_{t,n} = h_{t-1,n} D_{t,n} R_t + u_{t,n}. \tag{40}$$

After two steps,

$$h_{2,n} = h_{0,n} D_{1,n} R_1 D_{2,n} R_2 + u_{1,n} D_{2,n} R_2 + u_{2,n}. \tag{41}$$

For this family to close in the small summary algebra of Proposition 3.2, one would need a common right factor $R_\star$ and rowwise diagonal matrices $\widetilde{D}_n$ such that

$$D_{1,n} R_1 D_{2,n} R_2 = \widetilde{D}_n R_\star \qquad \text{for all } n. \tag{42}$$

This is not true in general. For example, take $P = 2$, $R_2 = I$, $D_{1,n} = I$,

$$R_1 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \quad D_{2,1} = I, \quad D_{2,2} = \mathrm{Diag}(1, 2).$$

Then the first rows of $R_1 D_{2,1}$ and $R_1 D_{2,2}$ are $(1, 1)$ and $(1, 2)$, which cannot both be scalar multiples of a single row of $R_\star$. Thus Eq. 42 fails. This does not make Eq. 39 invalid; exact scan could be recovered with larger rowwise $P \times P$ summaries. The group-tied left dynamics in Section 3 are introduced to keep the summary algebra lightweight.

C.2. Controller-compatible scan closure

Proof of Proposition 3.2. Conditional on a precomputed controller path, L_t, R_t, and U_t are fixed with respect to the main memory variable. Each step is therefore an affine map $F_t(H) = L_t H R_t + U_t$. For two steps,

$$F_2(F_1(H)) = L_2(L_1 H R_1 + U_1)R_2 + U_2 = (L_2 L_1)H(R_1 R_2) + L_2 U_1 R_2 + U_2.$$

Hence summaries compose as in Eq. 3. Associativity follows from associativity of function composition, giving a standard prefix scan computation (Blelloch, 1990; Martin & Cundy, 2018). □

Direct state coupling generically breaks this finite affine algebra. If $R_t = R(\xi_t, H_{t-1})$, then $F(H) = L_t H R(\xi_t, H) + U_t$ is affine only under strong degeneracies. Indeed, affinity implies $F(sH) - F(0) = s(F(H) - F(0))$, hence $L_t H R(\xi_t, sH) = L_t H R(\xi_t, H)$ for all scalars s. Whenever $L_t H$ has full column rank, right multiplication is injective, forcing $R(\xi_t, sH) = R(\xi_t, H)$ along that ray. A concrete non-affine case is

$$R(H) = \exp\big(\kappa \langle W, H\rangle_F M\big) = I_P + \kappa \langle W, H\rangle_F M,$$

where $M^2 = 0$ and $\kappa \neq 0$. The update contains the quadratic term $\kappa\langle W, H\rangle_F L_t H M$, so it cannot be represented by a finite triple (L, R, U). State-coupled transports can still be defined pathwise, but not with this exact block-affine scan unless the summary is enlarged or the coupling is degenerate.

C.3. Collapse under fixed simultaneous reduction

Proof of Proposition 3.3. Assume $G_m = Q\Lambda_m Q^{-1}$ for a fixed $Q \in \mathrm{GL}(P)$, with all Λ_m diagonal or block-diagonal in the same fixed partition. Then

$$A_{R,t} = \sum_m \rho_{t,m} G_m = Q\Big(\sum_m \rho_{t,m}\Lambda_m\Big)Q^{-1} = Q\Lambda_t Q^{-1},$$

where Λ_t has the same diagonal or block-diagonal structure. For the dense exponential, $R_t = \exp(\Delta_t A_{R,t}) = Q\exp(\Delta_t \Lambda_t)Q^{-1}$, and the middle factor preserves the same fixed blocks. For a fixed-order split product, every factor satisfies $\exp(\Delta_t \rho_{t,m} G_m) = Q\exp(\Delta_t \rho_{t,m}\Lambda_m)Q^{-1}$; the product therefore has the same form and the same block partition.

Set $\widetilde{H}_t = H_t Q$, $\widetilde{U}_t = U_t Q$, and $\widetilde{R}_t = Q^{-1} R_t Q$. Then

$$\widetilde{H}_t = L_t \widetilde{H}_{t-1}\widetilde{R}_t + \widetilde{U}_t,$$

with $\widetilde{R}_t$ diagonal or fixed-block diagonal. Thus the transformed columns split into independent scalar banks or independent fixed blocks. If $U_t = b_t x_t^\top$, then $U_t Q = b_t(Q^\top x_t)^\top$, so the static basis change can be absorbed into the group input coordinates. This proves the stated sufficient collapse criterion. □

The proposition is intentionally one-sided. It identifies a common degeneracy, but it is not a classification of all possible right-action degeneracies. Noncommutativity alone is also insufficient: noncommuting generators can still share a fixed nontrivial invariant block decomposition.

C.4. Split-flow factor actions

The dense backend uses matrix exponentials in the standard sense (Higham, 2008). A split backend implements the right action as a product of cheap invertible factors. For $H \in \mathbb{R}^{N\times P}$:

1. If $D = \mathrm{Diag}(\delta)$, then $\exp(D) = \mathrm{Diag}(e^\delta)$, so $H\exp(D)$ is column scaling.
2. If $\Omega_{ij} = e_j e_i^\top - e_i e_j^\top$, then $\Omega_{ij}^\top = -\Omega_{ij}$, and $\exp(\phi\Omega_{ij})$ rotates only columns (i, j):

$$(H_{:,i}, H_{:,j}) \mapsto (H_{:,i}, H_{:,j})\begin{pmatrix}\cos\phi & -\sin\phi \\ \sin\phi & \cos\phi\end{pmatrix}.$$

3. If $N_{ij} = e_i e_j^\top$ with $i \neq j$, then $N_{ij}^2 = 0$ and $\exp(\eta N_{ij}) = I + \eta N_{ij}$; right multiplication adds η times column i to column j.

4. If $A = uv^\top$ is rank one, then

$$\exp(suv^\top) = I + \varphi(sv^\top u)suv^\top, \qquad \varphi(\kappa) = \begin{cases} (e^\kappa - 1)/\kappa, & \kappa \neq 0, \\ 1, & \kappa = 0. \end{cases}$$

Hence $H \exp(suv^\top) = H + \varphi(sv^\top u)s(Hu)v^\top$.

The inverse of each factor is obtained by negating the scalar parameter. For small $|\kappa|$, $\varphi(\kappa)$ should be evaluated with an `expm1`-style implementation.

C.5. Lie–Trotter split products

In this subsection, $\|\cdot\|_{\mathrm{op}}$ denotes the Euclidean operator norm for right-action matrices, and $\|\cdot\|_F$ denotes the Frobenius norm for memory and source matrices.

If $A = \sum_{k=1}^K A_k$ and $\widetilde{R}(\Delta) = \prod_{k=1}^K \exp(\Delta A_k)$ in a fixed order, then standard Lie–Trotter splitting gives

$$\|\widetilde{R}(\Delta) - \exp(\Delta A)\|_{\mathrm{op}} \leq C\Delta^2 \tag{43}$$

for bounded factors and sufficiently small Δ (Hairer et al., 2006). A direct proof is obtained by expanding every factor as $I + \Delta A_k + O(\Delta^2)$; the product has first-order term $\Delta \sum_k A_k$, matching $\exp(\Delta A)$, and all remaining terms are $O(\Delta^2)$. The scan algebra is nevertheless exact for the implemented discrete recurrence after R_t—dense or split—has been fixed.

C.6. Exponential-adjusted source update

We now justify Eq. 5. Freeze a_t and $A_{R,t}$ on one interval of length Δ_t, set $D_t = \mathrm{Diag}(a_t)$, and consider

$$\dot{H}(\tau) = D_t H(\tau) + U(\tau) + H(\tau) A_{R,t}.$$

For the dense backend, variation of constants gives

$$H(t) = L_t H(t - \Delta_t) R_t + \int_0^{\Delta_t} e^{(\Delta_t - s)D_t} U(t - \Delta_t + s) e^{(\Delta_t - s)A_{R,t}}\, ds, \tag{44}$$

where $L_t = e^{\Delta_t D_t}$ and $R_t = e^{\Delta_t A_{R,t}}$. Let the Duhamel integrand be $G(s)$. Its endpoint values are $G(0) = L_t U_{t-1} R_t$ and $G(\Delta_t) = U_t$. The two-point quadrature

$$\int_0^{\Delta_t} G(s)\, ds \approx (1 - \lambda_t)\Delta_t G(0) + \lambda_t \Delta_t G(\Delta_t)$$

therefore yields exactly the source term in Eq. 5:

$$\widehat{U}_t = (1 - \lambda_t)\Delta_t L_t U_{t-1} R_t + \lambda_t \Delta_t U_t.$$

If $A_{R,t} \equiv 0$, then $R_t = I_P$, and Eq. 5 reduces to the one-sided diagonal-left selective update. With $P = 1$ and $\lambda_t = 1$, this is the usual one-channel exponential-Euler source update.

C.7. Source quadrature and split-backend replacement

The source rule above is a quadrature approximation to the dense frozen-flow Duhamel integral, not an approximation to the scan algebra. For $\lambda_t \in [0, 1]$, using the Frobenius norm on the $N \times P$ matrix-valued source integrand,

$$\left\| \int_0^{\Delta_t} G(s)\, ds - ((1 - \lambda_t)\Delta_t G(0) + \lambda_t \Delta_t G(\Delta_t)) \right\|_F \leq \frac{\Delta_t^2}{2} \sup_{s \in [0, \Delta_t]} \|G'(s)\|_F. \tag{45}$$

If $\lambda_t = 1/2$ and $G \in C^2$, the trapezoidal remainder is $O(\Delta_t^3)$; the same order holds for $\lambda_t = 1/2 + O(\Delta_t)$. If a split backend is used, the same algebraic cell is exact for the chosen split right action. Comparing it to the dense frozen flow adds the replacement error

$$\|L_t(H_{t-1} + \beta_t U_{t-1})(R_t^{\text{split}} - R_t^{\text{den}})\|_F, \qquad \beta_t = (1 - \lambda_t)\Delta_t,$$

which is controlled by Eq. 43 under bounded one-step inputs and bounded $\|L_t\|_{\text{op}}$. Thus split-flow approximation error and source-quadrature error are separate; the block-affine scan remains exact for whichever discrete right action is actually implemented.

D. Experimental Details and Additional Diagnostics

This appendix supports the empirical claims in Section 4. It records the Transport-MQAR generator, metrics, selection protocol, model geometry, full result tables, controller counterfactuals, and limited readiness checks. The experiments evaluate SHiPPO as a structured transported-memory prior; Transport-MQAR, controller counterfactuals, and supporting controls provide diagnostic evidence, while the byte-level language-modeling, adapter, and backend checks are included only as stability and readiness notes.

D.1. Transport-MQAR task, generator, and metrics

Transport-MQAR is a finite-field variant of multi-query associative recall (MQAR) (Arora et al., 2024). Examples define key–value bindings over $\mathbb{F}_{31}$, values have $p = 4$ coordinates, and operation tokens apply invertible right actions to all currently stored values. A query token asks for the current transported value of a key. Thus success cannot be explained by fixed-coordinate recall alone: the model must recover the value in the frame induced by the intervening operations.

Table 3. Transport-MQAR generator configuration for the reported primary runs.

Field	Value
Finite field	$\mathbb{F}_{31}$
Target coordinates	$p = 4$
Number of keys	256
Configured training length	512
Evaluation lengths	128, 512, 2048, 4096
Generator mode	`nonreducible`
Event probabilities	$p_{\text{op}} = 0.50,\ p_{\text{bind}} = 0.22,\ p_{\text{query}} = 0.28$
Ensure at least one query	`true`
Number of operation generators	13
Evaluation examples	640 per length and seed unless noted otherwise
Training loss	coordinate-wise cross-entropy at query positions
Reported metrics	coordinate accuracy and exact all-coordinate accuracy

Algorithmically, the generator maintains a dictionary of key–value bindings. A binding event samples a key and a value $v \in \mathbb{F}_{31}^4$, emits one key token followed by four coordinate-value tokens, and stores the value in the current frame. An operation event emits an operation token and applies the corresponding invertible matrix to every stored value. A query event emits a key-specific query token and supplies as target the current transported value associated with that key. Targets are emitted only at query positions. If a sampled sequence contains no query and `ensure_query` is enabled, the tail is overwritten by a minimal bind/query sequence for a fixed key.

Table 4. Transport-MQAR token layout for $q = 31$, $p = 4$, and 256 keys.

Field	Value
Pad token	0
Key tokens	$1, \ldots, 256$
Value-token offset	257
Value tokens	$257 + j \cdot 31 + a$, $j \in \{0, 1, 2, 3\}$, $a \in \{0, \ldots, 30\}$
Operation-token offset	381
Operation tokens	$381, \ldots, 393$
Query-token offset	394
Query tokens	$394, \ldots, 649$
Vocabulary size	650
Output coordinate classes	$4 \cdot 31 = 124$

The `nonreducible` mode uses the following 13 matrices over $\mathbb{F}_{31}$, with all entries interpreted modulo 31:

$$\texttt{rot_0_1} \begin{bmatrix} 0 & 30 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \texttt{shear_0_1} \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\texttt{shear_1_0} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \texttt{rot_1_2} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 30 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\texttt{shear_1_2} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \texttt{shear_2_1} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\texttt{rot_2_3} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 30 \\ 0 & 0 & 1 & 0 \end{bmatrix} \qquad \texttt{shear_2_3} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\texttt{shear_3_2} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \qquad \texttt{rot_0_3} \begin{bmatrix} 0 & 0 & 0 & 30 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

$$\texttt{shear_0_3} \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \texttt{shear_3_0} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}$$

$$\texttt{wrap_diag} \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 16 \end{bmatrix}$$

Coordinate accuracy is the fraction of target coordinates predicted correctly. Exact accuracy is the fraction of query targets for which all four coordinates are correct. For $q = 31$, the random coordinate baseline is $1/q \approx 0.0323$. We emphasize coordinate accuracy because it directly measures transported coordinate recovery; exact accuracy is a stricter joint criterion.

D.2. Experimental protocol and selection

Final out-of-distribution evaluation lengths are not used for model selection. For SHiPPO-family variants, hyperparameters are selected using validation data at the configured training length only. Matched-capacity controls use a separate shared stress-test budget, reported in the matched-control rows of Table 5; they are capacity stress tests rather than fully tuned architecture baselines. All main Transport-MQAR results are means and sample standard deviations over $n = 3$ independent training seeds. We report standard deviations, not confidence intervals, because $n = 3$ is too small for reliable distributional

assumptions. Training uses AdamW (Loshchilov & Hutter, 2019) with gradient clipping as specified below. The DirectGen-SingleExp rows are SHiPPO-family right-action realization controls.

Table 5. Training hyperparameters for the reported Transport-MQAR runs. For primary families, the seed column records the audited configuration seed, not the full set of random seeds used for the $n = 3$ aggregation.

Run family	Audit config seed	Optimizer	LR	WD	Batch	Steps	Eval every	Clip
GRU primary	106	AdamW	5×10^{-4}	0.01	16	5000	250	1.0
Transformer primary	107	AdamW	5×10^{-4}	0.01	16	5000	250	1.0
Free enc/dec primary	109	AdamW	5×10^{-4}	0.01	16	5000	250	1.0
No-right primary	101	AdamW	5×10^{-4}	0.01	16	5000	250	1.0
DirectGen-SingleExp primary	108	AdamW	5×10^{-4}	0.01	16	5000	250	1.0
StructGen-Split primary	105	AdamW	5×10^{-4}	0.01	16	5000	250	1.0
GRU matched	0,1,2	AdamW	10^{-3}	0.0	32	1000	100	1.0
Transformer matched	0,1,2	AdamW	10^{-3}	0.0	32	1000	100	1.0
DirectGen-SingleExp matched	0,1,2	AdamW	10^{-3}	0.0	32	1000	100	1.0

All audited Transport-MQAR configurations use dropout 0.0 and constant learning rate. We did not find an explicit scheduler, warmup, autocast, or GradScaler path in the audited diagnostic runs. Linear modules in the critical-baseline implementation use Xavier-uniform weights and zero biases; other modules follow their model-defined or framework-default initializations. Runs were executed on NVIDIA A100 40GB graphics processing units (GPUs); long evaluations used smaller batches when needed for memory. Because hardware and kernels were not optimized, wall-clock time is not used as a claim.

D.3. Model geometry and update equations

In a SHiPPO-family layer, the residual stream has width d_{model}. A block expansion factor e gives recurrent cell width $D_{\text{cell}} = ed_{\text{model}}$. The cell is partitioned into $G = D_{\text{cell}}/P_g$ groups of width P_g. Each group has state $H_{t,g} \in \mathbb{R}^{N\times P_g}$, so the per-layer recurrent state size is $GNP_g = D_{\text{cell}}N$.

Table 6. Model geometry and parameter counts in the primary Transport-MQAR comparison.

Model	d_{model}	e	D_{cell}	N	P_g	G	State	Role	Params
GRU (Cho et al., 2014)	128	–	–	–	–	–	128	same-width recurrent baseline	0.50M
Transformer (Vaswani et al., 2017)	128	–	–	–	–	–	128	same-width attention baseline	0.89M
Free enc/dec	128	2	256	32	4	64	8192	static-basis control	5.11M
No-right	128	2	256	32	4	64	8192	right-action pathway ablation	4.98M
DirectGen-SingleExp	128	1	128	32	4	32	4096	direct-generator single-exponential right-action control	1.52M
StructGen-Split	128	2	256	32	4	64	8192	structured-generator split-product right-action realization	6.03M

For each group, StructGen-Split uses

$$H_{t,g} = L_{t,g}H_{t-1,g}R_{t,g} + \widehat{U}_{t,g}, \qquad U^{\text{raw}}_{t,g} = b_{t,g}x^{\top}_{t,g}, \tag{46}$$

where $L_{t,g}$ is diagonal on the left and $R_{t,g}$ is a split-product right action built from a structured generator library. The source convention is

$$\widehat{U}_{t,g} = (1-\lambda_{t,g})\Delta_{t,g}L_{t,g}U^{\text{raw}}_{t-1,g}R_{t,g} + \lambda_{t,g}\Delta_{t,g}U^{\text{raw}}_{t,g}. \tag{47}$$

The no-right ablation fixes $R_{t,g} = I$.

DirectGen-SingleExp keeps the same group-local matrix-state form but directly emits a group-local right generator $A_{t,g} \in \mathbb{R}^{P_g\times P_g}$ from the controller and forms the right action by a single matrix exponential,

$$R^{\text{DSE}}_{t,g} = \exp(\Delta_{t,g}A_{t,g}).$$

In the audited implementation, the controller emits

$$\theta_{t,g} = (a_{t,g}, b_{t,g}, \Delta_{t,g}, \lambda_{t,g}, A_{t,g}),$$

with $a_{t,g} \leq 0$, $\Delta_{t,g} > 0$, and $\lambda_{t,g} \in [0,1]$. The raw generator is bounded entrywise and its diagonal is made dissipative. Writing

$$Z_{t,g} = \alpha \tanh(\widetilde{A}_{t,g}),$$

the implemented generator is

$$A_{t,g} = Z_{t,g} - \mathrm{Diag}\left(\mathrm{softplus}(\mathrm{diag}(Z_{t,g}))\right),$$

so the off-diagonal entries are directly emitted bounded entries, while the diagonal satisfies

$$(A_{t,g})_{ii} = -\,\mathrm{softplus}(-(Z_{t,g})_{ii}) < 0.$$

Its update is

$$H_{t,g} = L_{t,g} H_{t-1,g} R_{t,g}^{\mathrm{DSE}} + \widehat{U}_{t,g},$$

where

$$L_{t,g} = \mathrm{Diag}\left(\exp(\Delta_{t,g} a_{t,g})\right), \qquad U_{t,g}^{\mathrm{raw}} = b_{t,g} x_{t,g}^{\top},$$

and

$$\widehat{U}_{t,g} = (1-\lambda_{t,g})\Delta_{t,g} L_{t,g} U_{t-1,g}^{\mathrm{raw}} R_{t,g}^{\mathrm{DSE}} + \lambda_{t,g}\Delta_{t,g} U_{t,g}^{\mathrm{raw}}.$$

With column-vector vec, the homogeneous part is

$$\mathrm{vec}(H_{t,g}) = \left((R_{t,g}^{\mathrm{DSE}})^{\top} \otimes L_{t,g}\right) \mathrm{vec}(H_{t-1,g}),$$

so the transition is Kronecker-structured by a left diagonal factor and a group-local right action. The comparison with StructGen-Split changes both the right-generator family and the discrete right-action backend. The free encoder/decoder baseline wraps a no-right memory with static MLP input/output transformations. It tests static basis flexibility, not pathwise transport of previously written memory. GRU and Transformer controls are same-width sequence baselines and do not implement an $N \times P_g$ transported coefficient state.

D.4. Implementation conventions

The scan-compatible recurrence is exact for the chosen discrete factors. A step summary (L_t, R_t, V_t) acts as

$$H_t = L_t H_{t-1} R_t + V_t, \tag{48}$$

and summaries compose associatively as

$$(L_2, R_2, V_2) \star (L_1, R_1, V_1) = \left(L_2 L_1,\ R_1 R_2,\ L_2 V_1 R_2 + V_2\right). \tag{49}$$

Forward scanning computes prefixes of this associative operation, while recurrent decoding applies the same recurrence serially. Dense right actions use $R_t = \exp(\Delta_t A_{R,t})$; split-flow actions use a fixed-order product of structured factors. In either case, the scan is exact for the implemented discrete action. Controller outputs satisfy $\Delta_{t,g} > 0$ and $\lambda_{t,g} \in [0,1]$. Direct dependence of $R_{t,g}$ on the scanned main memory $H_{t-1,g}$ is excluded in these experiments, matching the controller-compatible condition in Section 3. Here, "single exponential" refers to one matrix exponential of the assembled step generator. This should be distinguished from the fact that split-flow also uses exponentials: split forms R_t as a fixed-order product of exponentials of structured factors. A structured-generator single-exponential backend and DirectGen-SingleExp are different variants, because DirectGen-SingleExp also changes the generator family to a directly emitted group-local generator.

D.5. Full primary Transport-MQAR results

Tables 7 and 8 report the full length sweep behind Table 1. The main-text interpretation is based on these tables: StructGen-Split improves coordinate accuracy relative to no-right and static-basis ablations, while DirectGen-SingleExp remains stronger on exact accuracy. Thus the results support the usefulness of a right-action pathway while leaving the preferred discrete right-action realization open.

Table 7. Primary Transport-MQAR coordinate accuracy. Mean ± sample standard deviation over $n = 3$ seeds.

Model	128	512	2048	4096
GRU	0.1841 ± 0.0012	0.1150 ± 0.0021	0.0982 ± 0.0011	0.0952 ± 0.0017
Transformer	0.2357 ± 0.0145	0.0986 ± 0.0099	0.0506 ± 0.0032	0.0416 ± 0.0012
Free enc/dec	0.1897 ± 0.0030	0.1194 ± 0.0003	0.0990 ± 0.0041	0.0916 ± 0.0100
No-right	0.1950 ± 0.0030	0.1235 ± 0.0008	0.1053 ± 0.0004	0.1019 ± 0.0000
DirectGen-SingleExp	0.1966 ± 0.0041	0.1248 ± 0.0009	0.1066 ± 0.0006	0.1032 ± 0.0007
StructGen-Split	0.2115 ± 0.0028	0.1346 ± 0.0007	0.1140 ± 0.0002	0.1104 ± 0.0010

Table 8. Primary Transport-MQAR exact accuracy. Mean ± sample standard deviation over $n = 3$ seeds.

Model	128	512	2048	4096
GRU	0.0535 ± 0.0021	0.0239 ± 0.0026	0.0167 ± 0.0021	0.0159 ± 0.0024
Transformer	0.0648 ± 0.0120	0.0166 ± 0.0035	0.0043 ± 0.0010	0.0021 ± 0.0006
Free enc/dec	0.0446 ± 0.0023	0.0207 ± 0.0003	0.0120 ± 0.0023	0.0094 ± 0.0042
No-right	0.0501 ± 0.0022	0.0242 ± 0.0001	0.0170 ± 0.0004	0.0158 ± 0.0003
DirectGen-SingleExp	0.0851 ± 0.0020	0.0468 ± 0.0017	0.0375 ± 0.0019	0.0355 ± 0.0005
StructGen-Split	0.0795 ± 0.0021	0.0439 ± 0.0016	0.0342 ± 0.0018	0.0329 ± 0.0013

D.6. Capacity stress tests

Matched-capacity stress tests probe whether the diagnostic picture changes under a separate shared stress-test budget, reported in the matched-control rows of Table 5. They are capacity checks rather than the primary same-width comparison. The matched GRU and Transformer do not close the long-length coordinate gap to StructGen-Split. The matched DirectGen-SingleExp control is competitive with StructGen-Split, reinforcing the main-text scope: the structured-generator split-product realization is not claimed to dominate direct-generator single-exponential right transport.

Table 9. Matched-capacity Transport-MQAR coordinate accuracy. Mean ± sample standard deviation over $n = 3$ seeds.

Model	Params	128	512	2048	4096
GRU matched	6.45M	0.1937 ± 0.0036	0.1215 ± 0.0010	0.1042 ± 0.0003	0.1013 ± 0.0007
Transformer matched	5.02M	0.1208 ± 0.0063	0.0562 ± 0.0031	0.0386 ± 0.0004	0.0353 ± 0.0001
DirectGen-SingleExp matched	5.84M	0.2081 ± 0.0056	0.1326 ± 0.0005	0.1136 ± 0.0004	0.1102 ± 0.0002

Table 10. Matched-capacity Transport-MQAR exact accuracy. Mean ± sample standard deviation over $n = 3$ seeds.

Model	Params	128	512	2048	4096
GRU matched	6.45M	0.0535 ± 0.0013	0.0261 ± 0.0012	0.0193 ± 0.0008	0.0181 ± 0.0009
Transformer matched	5.02M	0.0193 ± 0.0029	0.0046 ± 0.0007	0.0011 ± 0.0001	0.0006 ± 0.0001
DirectGen-SingleExp matched	5.84M	0.0823 ± 0.0025	0.0450 ± 0.0012	0.0357 ± 0.0008	0.0341 ± 0.0006

D.7. Learned-controller audit and counterfactuals

The StructGen-Split and no-right cells share corresponding state-dict keys, but their controller output dimensions differ. In the reported configuration, the StructGen-Split controller emits 5248 scalars per token, whereas no-right emits 4224. The additional 1024-dimensional suffix consists of StructGen-Split right-generator controller coordinates absent from the no-right cell.

Table 11. Controller-head audit for StructGen-Split checkpoints.

Checkpoint	Shared abs. mean	Extra abs. mean	Shared norm	Extra norm	Extra dim
seed 0	0.032498	0.030625	44.787	20.751	1024
seed 1	0.032236	0.031719	44.761	21.472	1024
seed 2	0.031746	0.032846	43.717	22.625	1024

Across seeds, the suffix is not driven to zero. We therefore run an evaluation counterfactual that zeros this suffix while keeping all other trained weights fixed. No retraining is performed.

Table 12. Evaluation-time counterfactual for the StructGen-Split controller suffix. Means over three seeds.

Length	Coord. normal	Coord. zeroed	Δ coord.	Exact normal	Exact zeroed	Δ exact
128	0.2115	0.1936	-0.0179	0.0795	0.0436	-0.0359
512	0.1346	0.1250	-0.0096	0.0439	0.0239	-0.0200
2048	0.1140	0.1074	-0.0066	0.0342	0.0185	-0.0157
4096	0.1104	0.1044	-0.0060	0.0329	0.0176	-0.0154

This counterfactual shows dependence on the additional right-action controller coordinates. It does not identify the learned transport geometry or prove that the model implements the intended noncommutative mechanism.

D.8. Group-local transport tradeoff

The scan-compatible realization is group-local: $R_{t,g}$ transports memory coordinates only within group g. Increasing or decreasing P_g changes the locality of the right action, the number of groups, controller output dimension, parameter count, and optimization behavior. We performed exploratory seed-0 pilot sweeps over P_g, but these used separate pilot configurations from the primary $n = 3$ comparison in Tables 6–8. We therefore treat the pilot sweep as exploratory and do not use it as quantitative evidence.

D.9. State-tracking stress tests

State tracking is a harder stress test. The nonreducible state-tracking results in Table 13 are near chance and are included only as a documented failure mode under the current budget; they are not evidence for the main transported-recall claim.

Table 13. Nonreducible state-tracking stress-test summary.

Model	Coord@128	Coord@512	Coord@2048
StructGen-Split	0.044 ± 0.002	0.035 ± 0.000	0.033 ± 0.000
DirectGen-SingleExp	0.034 ± 0.000	0.032 ± 0.000	0.032 ± 0.000
No-right	0.045 ± 0.002	0.034 ± 0.000	0.033 ± 0.000

D.10. FineWeb-Edu byte-level language-model stability pilot

We also ran a small byte-level language-model (byte-LM) stability pilot on FineWeb-Edu (Penedo et al., 2024). A 21.6M-parameter SHiPPO byte-LM was trained for 1000 steps, corresponding to 4.096M training tokens. Training was NaN-free. The saved checkpoint obtained independent evaluation negative log-likelihood (NLL) 2.396, perplexity (PPL) 10.98, and token accuracy 0.337 over 204,800 tokens. This is a stability check, not a language-modeling benchmark.

Table 14. FineWeb-Edu 10k byte-level language-model stability pilot.

Run	Params	Train tokens	Train loss	Eval NLL	Eval PPL	Eval acc.
SHiPPO byte-LM	21.6M	4.096M	2.358	2.396	10.98	0.337

D.11. RULER and lm-eval readiness

The RULER adapter (Hsieh et al., 2024) supports prediction and contrastive scoring for the implemented SHiPPO byte-LM interface. The lm-evaluation-harness adapter (Biderman et al., 2024) registers the model under `shippo_byte_lm` and passes a small `arc_easy` smoke test based on the AI2 Reasoning Challenge (AI2-ARC) (Clark et al., 2018) with `--limit 2`. These are adapter readiness checks, not benchmark results.

Table 15. RULER and lm-eval readiness checks. These are smoke tests, not benchmark results.

Check	Status
RULER JSON Lines (JSONL) generation for NIAH/VT	pass
SHiPPO JSON Lines (JSONL) prediction adapter	pass
Continuation log-likelihood scorer	pass
Contrastive scoring path	pass
lm-eval model registration	pass
`arc_easy`, `--limit 2` command-line interface (CLI) smoke test	pass

D.12. Production backend and latency status

The current backend is a reference implementation of group-local block-affine scan. Fused kernels and matched wall-clock comparisons against optimized baselines are left for future work; the empirical evidence concerns transported-memory diagnostics, not production throughput.

D.13. Reproducibility, assets, and anonymization

This paper records the algorithmic task specification, model geometry, training hyperparameters, evaluation lengths, metric definitions, seed aggregation, and result tables needed to interpret the diagnostic claims. The experiments were run using an internal research codebase that also contains infrastructure for broader experiments. Because this codebase includes private paths, machine-specific scripts, and main-experiment infrastructure, we do not package it as a standalone code release. The empirical claims above are therefore tied to the task specification, model equations, hyperparameters, and tables in this appendix, rather than to an external artifact.

Table 16. Existing assets used in stability and readiness checks.

Asset	Use	License / terms
FineWeb-Edu (Hugging Face FineWeb Team, 2024)	byte-LM stability pilot	ODC-By v1.0; CommonCrawl terms
RULER (NVIDIA, 2024; Hsieh et al., 2024)	adapter smoke checks	Apache-2.0
lm-evaluation-harness (EleutherAI, 2024; Biderman et al., 2024)	evaluation adapter smoke check	MIT
ARC-Easy / AI2-ARC (Clark et al., 2018)	lm-eval smoke task	CC-BY-SA-4.0

E. Additional Related Work and Positioning

This appendix expands the main-text positioning by asking where a sequence model obtains its memory structure: from an online approximation objective, a selective state-space parameterization, a diagnostic for recall or state tracking, or a generic matrix-valued memory. The key distinction for SHiPPO is not the mere use of a matrix state or non-diagonal transition. It is that the two-sided dynamics are derived from a pathwise transported online-projection problem, and the scan-compatible cell of Section 3 is only one restricted implementation of that prior.

E.1. Online projection and memory priors

The closest predecessors are recurrent memories with explicit history-compression semantics. The Legendre Memory Unit represents a sliding history window in a Legendre basis, while HiPPO derives online polynomial projection dynamics for compressing the revealed signal (Voelker et al., 2019; Gu et al., 2020). Generalized orthogonal-basis views clarify how HiPPO-style coefficient dynamics connect to trainable state-space model (SSM) layers (Gu et al., 2023). S4 incorporates HiPPO-derived structure into deep state-space layers; Diagonal State Spaces (DSS) and S4D show how diagonal parameterization and initialization can retain much of the practical benefit (Gu et al., 2022a; Gupta et al., 2022; Gu et al., 2022b). Recent HiPPO-Zoo work revisits polynomial memories to expose adaptive, associative, and multiscale mechanisms in an interpretable basis (Goffinet et al., 2026). SHiPPO is aligned with this objective-derived tradition, but the modification is not a new basis or initialization: it transports the channel frame and metric of the approximation problem, yielding a Sylvester right-action term.

E.2. Selective state-space models and recurrent generative backbones

Modern autoregressive sequence models often use efficient recurrent or linear-time blocks as alternatives or complements to attention. S5 moves from many single-input single-output (SISO) systems to a multi-input multi-output (MIMO) SSM layer, H3 introduces multiplicative SSM-based interactions for language modeling, Mamba makes SSM parameters input-dependent, and structured state-space duality (SSD)/Mamba-2 exposes the structured semiseparable algebra behind selective recurrences (Smith et al., 2023; Fu et al., 2023; Gu & Dao, 2024; Dao & Gu, 2024). Mamba-3 continues this direction with improved discretization, complex dynamics, and MIMO updates (Lahoti et al., 2026). Gated Linear Attention (GLA), DeltaNet, Gated DeltaNet, HGRN2, and Kimi Linear combine gates, delta-style memory updates, state expansion, or chunkwise parallel algorithms for efficient language modeling (Yang et al., 2024a;b; 2025; Qin et al., 2024; Kimi Team, 2025). SHiPPO does not claim that selectivity or channel interaction is new. Its narrower question is when channel interaction can be interpreted as pathwise transport of memory coordinates, rather than as an untied mixer, gate, or controller.

E.3. Associative recall, state tracking, and length generalization

Associative recall and state-tracking diagnostics are useful because they expose differences that aggregate perplexity can hide. MQAR-style tasks in Zoology probe recall behavior of efficient language-model backbones, and recent work analyzes how Mamba's input selectivity affects approximation, memorization, and associative-recall capacity (Arora et al., 2024; Huang et al., 2025). Formal-language and state-tracking analyses show that SSMs, selective SSMs, and diagonal or gated linear recurrences have architecture-dependent strengths and failure modes (Merrill et al., 2024; Sarrof et al., 2024; Terzić et al., 2025a; Shakerinava et al., 2026). Structured-transition responses enrich the transition family while trying to preserve efficient recurrence, for example with sparse transition structure, Householder products, or fixed-point routes from diagonal to dense updates (Terzić et al., 2025b; Siems et al., 2025; Movahedi et al., 2025). Transport-MQAR is intended as a diagnostic in this spirit: it separates fixed-coordinate recall from coordinate recovery after a causal sequence of right actions. It should not be read as proof of general reasoning or as a substitute for natural generative benchmarks.

E.4. Matrix memories, fast weights, and objective-derived updates

Many efficient sequence models use matrix-valued states or outer-product updates. Linear attention can be viewed as fast-weight programming; DeltaNet, GLA, and Gated DeltaNet use delta rules or gates to control finite-state associative memories (Schlag et al., 2021; Yang et al., 2024b;a; 2025). xLSTM/mLSTM and HGRN2 increase recurrent capacity through exponential gating, covariance-style matrix memory, or state expansion (Beck et al., 2024; Qin et al., 2024). Test-time training (TTT), Longhorn, and MesaNet derive recurrent updates from online learning, amortized online learning, or locally optimal in-context regression objectives, and M^2RNN revisits nonlinear recurrent neural networks (RNNs) with matrix-valued states at language-modeling scale (Sun et al., 2025; Liu et al., 2025; von Oswald et al., 2026; Mishra et al.,

Table 17. Positioning by where memory dynamics, channel interaction, or matrix-valued structure enters. SHiPPO's central distinction is the source of the right-action term: it is induced by a transported online-projection problem.

Family	Where structure enters	SHiPPO distinction
Online projection memories	Basis choice and online approximation objective	Transports the projection family and channel metric, producing Sylvester coefficient dynamics
Selective SSMs and linear-time language models (LMs)	Token-dependent parameters, structured scans, discretization, channel interactions, and gates	Uses selectivity only after choosing a transported projection-memory ODE
Recall and state-tracking diagnostics	Synthetic tasks probing retrieval, finite-state updates, and length generalization	Tests transported coordinate recovery rather than fixed-coordinate recall alone
Fast-weight and matrix-memory models	Associative-memory matrices, gates, delta rules, or state expansion	Matrix axes have projection-coefficient and moving-channel-frame semantics
Structured-transition models	Sparse, product, unitary, low-rank, or dense transitions for expressivity and efficient recurrence	Reducible right transports collapse to static mixing plus independent banks

2026). SHiPPO is close to this literature in state shape and objective-derived spirit, but not in semantics. The left axis is an online approximation-coefficient axis; the right axis is a transported channel frame; and the two-sided update is the discretized form of a transported projection equation.

E.5. Summary of distinctions

Table 17 summarizes the intended positioning. The collapse result in Appendix C gives an additional algebraic caution: if the right-generator family is simultaneously reducible in a fixed basis, the apparent transport collapses to static mixing followed by independent scalar or fixed-block banks. This is why SHiPPO emphasizes transported projection semantics and non-reducible right-action families, rather than dense channel mixing alone.