FROM TRACE INVARIANCE TO CERTIFIED EXECUTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Independent update streams may interleave, but corrections within a stream must remain ordered. Trace invariance captures this symmetry; certified execution additionally requires reusable semantic transitions. We separate three questions: what observations identify, what a learner recovers, and what its rollout preserves. For finite permutation primitives, we characterize exactly the ambiguity left by terminal composition labels: one operation order leaves a permutation gauge, while both orders restrict it to a common centralizer. For stochastic learners, terminal target probabilities uniformly above one half in both orders guarantee correct row-wise projection when the centralizer is trivial. We then characterize when decoded recurrent states admit a finite transition audit, and how known product structure reduces its worst-case size independently of the number of streams. Controlled synthetic PatchWorld experiments demonstrate the benefits of this separation. Product-state recurrence attains perfect joint accuracy with 2,048 training programs, and intermediate supervision raises held-out-composition accuracy to 0.9118. Projecting the same learned operator scores raises length-16 joint accuracy from 9.6×10^{-5} under soft rollout to 1.0 across seven seeds. These results connect identifiable primitives, semantic-state structure, and discrete projection to finitely verifiable compositional execution.

1 INTRODUCTION

A destination may be set to Paris, a budget recorded as low, and the destination subsequently corrected to Rome. Moving the budget event should not change the resulting state; reversing the destination updates should. Such histories are words modulo swaps of independent events: *traces* in the sense of Mazurkiewicz (1987); Diekert & Rozenberg (1995). Treating the history as a set discards necessary order. Treating every linearization as a distinct sequence retains accidental scheduling. Yet removing this nuisance variation is only the first step toward reliable execution.

The more consequential question is whether a learned update is a function of the *semantic state*, rather than of the history used to reach that state. A recurrent model can predict intermediate states accurately on training histories without implementing one reusable transition per state and primitive. Conversely, a learner can recover the correct next-state argmax in every row, yet lose correctness when its execution propagates distributions rather than those discrete successors. Finally, poor recovery need not mean the training information was insufficient. These distinctions motivate an analysis of **identification, recovery, and execution** as separate objects (fig. 1).

We study a deliberately explicit setting: typed events routed to independent keys, a known finite semantic state space, and local primitives shared across keys. Our principal identification result characterizes the ambiguity in *terminal-only* supervision of two primitive families. Observing one composition order leaves an arbitrary relabeling between the families. Observing both orders eliminates exactly the relabelings that fail to commute with the observed reverse compositions. For PatchWorld arithmetic, the surviving relabeling is necessarily the identity. A quantitative extension shows that sufficiently confident terminal predictions force correct projected primitives even without intermediate labels.

Our execution analysis makes the architectural premise explicit: decoded states must be *closed under updates*. Only then does one semantic state–event row denote one behavior across all histories. Under this premise, reachable-row correctness is equivalent to exact execution of every finite prefix. Known keywise factorization and sharing reduce the point-audit size from an exponential joint table to one

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

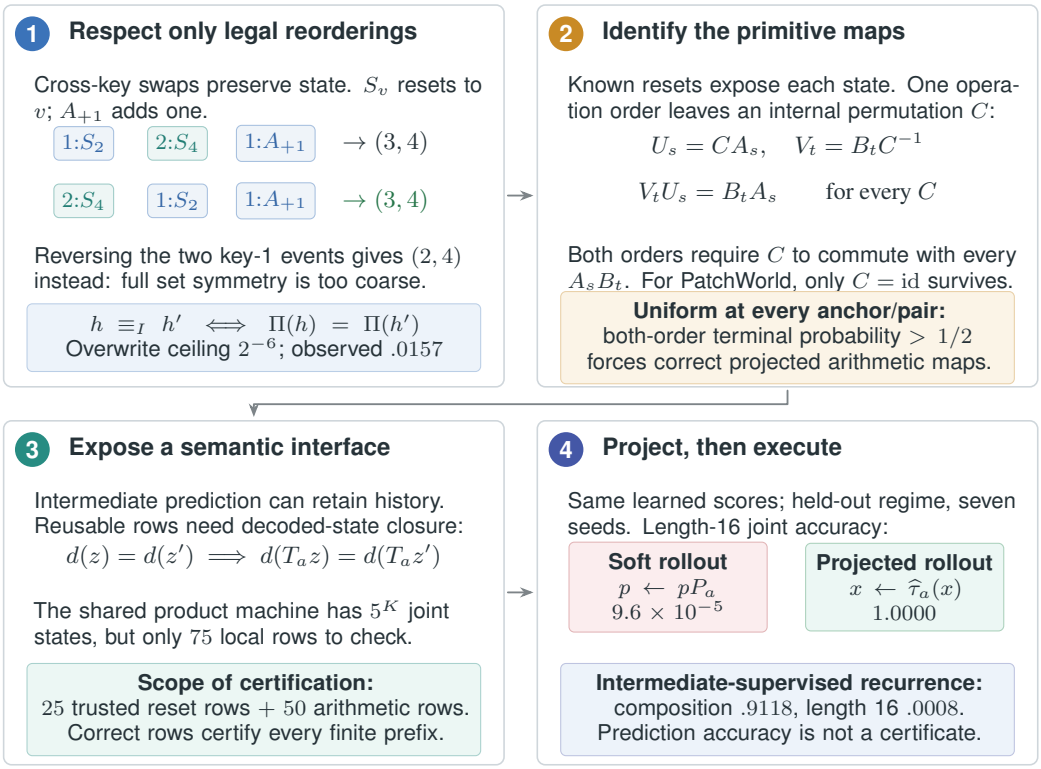

Figure 1: **From trace invariance to certified execution.** Legal cross-key swaps define the trace quotient; semantic continuation classes define a smaller state space. Terminal observations identify primitive maps only relative to their support and structural assumptions: one composition order leaves an internal permutation, while both orders can remove it. A discrete semantic state interface then makes a finite row audit meaningful. The numerical callouts are observed PatchWorld results; the centralizer and confidence statements are proved conditions, not additional experiments. The soft/projected comparison changes no learned scores.

local table. A separate two-state construction shows why arbitrarily small stochastic row errors can eventually reverse the decoded answer even when every row argmax is correct.

The empirical contribution is a controlled test of these distinctions. A matched study comprises 189 single-A100 runs with seven evaluation seeds, complemented by two studies of symmetry and transition learning. Product recurrence substantially improves low-data learning; canonicalized global recurrence nearly solves the tested finite key counts with sufficient data. Intermediate labels strongly improve short composition, while projected operators preserve execution across the tested lengths. Most decisively, projecting *the same learned operator scores* changes length-16 joint accuracy from 9.6×10^{-5} to 1.0. These comparisons isolate the roles of state organization, supervision, and execution in synthetic PatchWorld.

2 The Semantic Object: Traces and Continuation States

Write $[n] = \{1, \ldots, n\}$. Let $K \geq 1$, let $\mathcal{A}$ be a finite nonempty local event alphabet, $A = |\mathcal{A}|$, and let $\Sigma = [K] \times \mathcal{A}$. Histories lie in Σ^*. An event $e = (k, a)$ addresses key k; independence is

$$(e, e') \in I \quad \iff \quad k(e) \neq k(e'). \tag{1}$$

Write $h \equiv_I h'$ when adjacent swaps of independent events transform h into h', and $\pi_k(h)$ for the ordered local events at key k. This is a *declared semantic assumption*: cross-key effects or incorrect routing invalidate it.

Proposition 1 (Exact quotient and excessive symmetry). *$h \equiv_I h'$ iff $\Pi(h) = \Pi(h')$, where $\Pi(h) = (\pi_1(h), \ldots, \pi_K(h))$. Every trace-invariant function factors through Π. Stable sorting by key is an exact canonical representative.*

Proposition 2 (Overwrite ceiling). *If each key receives two distinct overwrites in independently uniform local order, every randomized predictor depending only on the event multiset has joint accuracy at most 2^{-K}.*

The first statement follows by stable sorting; the second by 2^K equiprobable final ledgers per multiset. Neither is an optimization bound.

Theorem 3 (Reachable-state commutation). *Let $F(h)$ be a deterministic recurrent state obtained by composing updates T_e from z_0, and let $\mathcal{R} = \{F(h) : h \in \Sigma^*\}$. Then F is trace-invariant iff $T_b T_a(z) = T_a T_b(z)$ for every $z \in \mathcal{R}$ and $(a, b) \in I$.*

This specializes the reachable-state commutation criterion of Cohen-Karlik et al. (2020, Theorem 5) to the declared independent pairs. It characterizes invariance of the *state representation*. Invariance of a noninjective output alone does not imply hidden-state commutation. An approximate Lipschitz bound appears in proposition 14.

Trace equivalence removes schedules but generally retains an unbounded local history. To identify memory sufficient for the task, let $g_k : \mathcal{A}^* \to \mathcal{Y}_k$ be the local target and define classical continuation equivalence (Nerode, 1958):

$$u \sim_{g_k} v \quad \Longleftrightarrow \quad g_k(uw) = g_k(vw) \text{ for all } w \in \mathcal{A}^*. \tag{2}$$

Let $Q_k = \mathcal{A}^*/\!\sim_{g_k}$ have finite size r_k, and $G(h) = (g_k(\pi_k(h)))_{k=1}^K$.

Theorem 4 (Canonical product quotient). *The continuation quotient of G is exactly $Q_1 \times \cdots \times Q_K$, reached by $\Phi(h) = ([\pi_k(h)]_{\sim_{g_k}})_k$. It has $\prod_k r_k$ states. Every reachable deterministic Moore machine computing G has a unique surjective initial-state-, transition-, and output-preserving map onto this product machine.*

Corollary 5 (PatchWorld state cardinality). *For a five-valued local state with resets to every value and the state itself as output, the local continuation index is five. The minimal K-key executor has 5^K discrete states.*

This classical minimality statement bounds semantic state count: a binary encoding needs $\lceil K \log_2 5 \rceil$ bits. Neural hidden width is a separate resource, evaluated by our gated recurrent unit (GRU) scaling controls.

3 What terminal supervision identifies

Consider a known semantic state set X of size $R \geq 2$, initial state x_0, and deterministic primitives $\tau_a : X \to X$. *Identification* asks whether all maps consistent with the stipulated observations induce the same execution; *recovery* asks what a particular training procedure returns. The distinction depends on the observation model.

Theorem 6 (Row observations only). *Suppose the only constraints fix the true successors on a row set $\mathcal{C} \subseteq X \times \mathcal{A}$, with no other parameter ties or compositional constraints. Exactly $R^{RA-|\mathcal{C}|}$ deterministic tables are compatible. They all execute every word correctly from x_0 iff $\mathcal{E}_\tau \subseteq \mathcal{C}$, where $\mathcal{E}_\tau$ contains every state–event row reachable under τ.*

Composed terminal labels are *not row-only observations*: they can jointly constrain otherwise unobserved primitive rows.

Theorem 7 (Composition-order identification). *Let $\{A_s\}_{s \in S}$ and $\{B_t\}_{t \in T}$ be nonempty families of permutations of X. Assume trusted reset/access operations expose every labeled $x \in X$. Let $U_s, V_t : X \to X$ be arbitrary deterministic candidate primitives. Composition acts right to left, and $\mathrm{Sym}(X)$ is the permutation group of X.*

One order. *Exact terminal labels for every x, s, t impose $V_t U_s = B_t A_s$. The compatible candidates are exactly*

$$U_s = C A_s, \qquad V_t = B_t C^{-1}, \qquad C \in \mathrm{Sym}(X). \tag{3}$$

Both orders. *Adding $U_s V_t = A_s B_t$ restricts C exactly to*

$$\mathcal{Z} = \{C \in \mathrm{Sym}(X) : C(A_s B_t) = (A_s B_t)C \ \forall s, t\}. \tag{4}$$

Thus both orders identify every primitive uniquely iff this common centralizer is trivial.

Proof of theorem 7. First suppose $V_t U_s = B_t A_s$ for all s, t. A composition $V_t U_s$ that is bijective implies that U_s is injective and V_t is surjective. Since each is an endomorphism of the same finite set, both are bijective. This holds for every s, t because both index sets are nonempty.

Fix $s_0 \in S$ and let $C = U_{s_0} A_{s_0}^{-1} \in \mathrm{Sym}(X)$. For every t,

$$V_t = B_t A_{s_0} U_{s_0}^{-1} = B_t C^{-1}.$$

Fixing any t_0 and using its equation for any s gives

$$U_s = V_{t_0}^{-1} B_{t_0} A_s = C A_s.$$

Conversely, these choices satisfy $(B_t C^{-1})(C A_s) = B_t A_s$ for every $C \in \mathrm{Sym}(X)$. Distinct C give distinct candidate families because $C A_{s_0}$ determines C. Thus there are exactly $R!$ compatible families for complete one-order observations.

Adding the reverse order requires

$$U_s V_t = C A_s B_t C^{-1} = A_s B_t \quad \text{for all } s, t.$$

This is equivalent to $C \in \mathcal{Z}$. The identity is always in $\mathcal{Z}$, and the injectivity just shown implies that the compatible family is unique precisely when $\mathcal{Z} = \{\mathrm{id}\}$. □

This is an internal *factorization ambiguity*, not a harmless relabeling of the observed input and output states, which are already anchored. It can change intermediate states and unseen compositions. Intermediate labels can fix this ambiguity directly, but the theorem shows that they are not the only way to remove it.

Theorem 8 (A terminal-confidence guarantee for projection)**.** *Replace U_s, V_t by row-stochastic kernels on X, with trusted point-mass resets. Suppose every terminal target in both orders has probability at least $1 - \varepsilon$, uniformly over x, s, t, where $0 \le \varepsilon < 1/2$. Row-wise argmax projections are uniquely defined permutations satisfying eq. (3) for some $C \in \mathcal{Z}$. If $\mathcal{Z} = \{\mathrm{id}\}$, all projected primitives equal the true primitives and execute every finite word exactly.*

Why confidence matters. Each order forces the second kernel to contain a distinct majority row for every target. Together, the two orders force both kernels to have permutation-valued majority maps. A cycle argument shows that any permutation-valued majority map of their product must compose these maps. Applying theorem 7 completes the argument. The condition is a uniform probability guarantee, stronger than terminal classification accuracy; the experiments below evaluate recovered rows and rollout accuracy rather than this confidence condition.

Corollary 9 (PatchWorld arithmetic is identifiable from both orders)**.** *On $X = \mathbb{Z}_5$, let $A_s(x) = x + s$ for $s \in \{1, 2, 3, 4\}$ and $B_t(x) = tx$ for $t \in \{2, 3, 4\}$. Then $\mathcal{Z} = \{\mathrm{id}\}$. With trusted resets, complete terminal labels for both arithmetic orders uniquely identify all arithmetic primitives. The conclusion also holds for row-wise projection under theorem 8.*

Proof of corollary 9. Let C commute with every $F_{s,t} = A_s B_t$. Since these are permutations, C also commutes with inverses and products. For fixed t,

$$F_{s,t} F_{s',t}^{-1} = A_s B_t B_t^{-1} A_{-s'} = A_{s-s'}.$$

Choosing $s = 2, s' = 1$ shows C commutes with A_1. Let $c = C(0)$. Repeated translation by one gives $C(x) = x + c$ for every $x \in \mathbb{Z}_5$. Commuting with $F_{1,2}(x) = 2x + 1$ implies

$$2x + 1 + c = 2(x + c) + 1 \pmod 5,$$

so $c = 0$ and C is the identity.

PatchWorld has five ADD tokens realizing increments $1, 2, 3, 4, 1$ and five MUL tokens realizing multipliers $2, 3, 4, 2, 3$. Complete support over these token indices includes all maps used above.

Apply theorem 7 to the token-indexed families: the same common C must govern every token, including duplicates. Thus all 50 arithmetic rows are identified. With a trusted reset prior, all 25 reset rows are fixed as well. In contrast, when only reset-first programs are labeled and no prior or intermediate constraints fix noninitial reset behavior, modifying any (x, SET_v) with $x \neq x_0$ leaves every training execution unchanged. Those 20 rows are genuinely unconstrained by such programs. □

The observed final-only-plus-reset row recovery of 0.7333 is therefore a *learning-procedure result*, not proof of insufficient terminal information.

4 What makes a learned executor certifiable

4.1 Semantic closure as an executable interface

Let a deterministic recurrent model have reachable hidden states Z_{reach}, update T_a, and semantic decoder $d : Z_{\mathrm{reach}} \to X$. The relevant structural property is

$$d(z) = d(z') \quad \Longrightarrow \quad d(T_a z) = d(T_a z') \quad \text{for all } z, z' \in Z_{\mathrm{reach}},\ a \in \mathcal{A}. \tag{5}$$

Theorem 10 (When semantic rows are well-defined)**.** *Condition (5) holds iff there is a unique transition map $\bar{\tau}_a$ on $d(Z_{\mathrm{reach}})$ satisfying $d \circ T_a = \bar{\tau}_a \circ d$. If $d(z_0) = x_0$, universal decoded correctness is equivalent to this closure condition together with $\bar{\tau}_a(x) = \tau_a(x)$ on all true-reachable rows (whose source states must lie in $d(Z_{\mathrm{reach}})$). Without a bound on hidden states or such structural restrictions, no finite set of history evaluations certifies all future prefixes.*

The last statement has a simple delayed-error construction: augment an otherwise correct machine with a counter that changes its decoded output only beyond the longest tested history. This remains possible even with exact trace invariance. Finite-state conformance testing likewise requires a declared implementation class or fault domain (Vaandrager & Melse, 2025). Intermediate supervision on finitely many histories is not a proof of eq. (5).

Trace-structured models. Trace-State Factorization (TSF) independently encodes each $\pi_k(h)$ with a GRU (Cho et al., 2014), shared across keys or key-specific. This enforces cross-key invariance but does not constrain hidden states decoding to the same x to have identical future behavior. Trace Operator Factorization (TOF) instead updates a semantic-state distribution by $p_{k,t} = p_{k,t-1} P_{a_t}$. Free learned rows use $P_a = \mathrm{softmax}(L_a)$ row-wise for scores $L_a \in \mathbb{R}^{R \times R}$; the stated exact-reset prior fixes $P_{\mathrm{SET}_v}[x, :] = \delta_v$ separately, where δ_v is the point mass at v. Let L_k be the local trace length, $x_{k,t}$ its true state after t events, and $\mathrm{CE}(p, x) = -\log p(x)$. The supervised objective is

$$\mathcal{L} = \sum_k \mathrm{CE}(p_{k,L_k}, x_{k,L_k}) + \lambda_{\mathrm{int}} \sum_k \sum_{t<L_k} \mathrm{CE}(p_{k,t}, x_{k,t}), \tag{6}$$

with $\lambda_{\mathrm{int}} = 0$ or 1. The recurrent control receives the same observed semantic targets, but TOF additionally assumes the enforced reset prior $\mathrm{SET}_v(x) = v$. The controls therefore match observed targets while distinguishing the reset prior as an additional structural constraint.

At inference, TOF retains the trusted resets, projects each learned row once with a fixed tie rule, and executes discrete states:

$$\widehat{\tau}_a(x) = \arg\max_y L_a[x, y], \qquad \widehat{x}_{k,t} = \widehat{\tau}_{a_t}(\widehat{x}_{k,t-1}). \tag{7}$$

Other keys are unchanged. Semantic closure and legal-schedule invariance now hold by construction; *correctness* still requires identifying or auditing the transitions. This is a finite-state restriction, not a guarantee that gradient optimization finds the right table.

4.2 Finite certificates and their scope

Theorem 11 (Reachable-row certificate)**.** *For deterministic tables $\tau, \widehat{\tau}$ on the same labeled state space and initial state, the following are equivalent: every finite prefix executes exactly; $\widehat{\tau}_a(x) = \tau_a(x)$ on every true-reachable row. For a fixed word, correctness on its true path rows suffices and is necessary*

for correctness of all its prefixes. For a known semantic reference and a candidate accessible only through successor queries, sound and complete deterministic auditors with no side information tying or fixing rows require all reachable rows to be queried in the worst case.

This is a white-box semantic interface: a point query checks a declared state–event successor, not an arbitrary recurrent history. Induction proves sufficiency; changing an unqueried row reached by a shortest access word proves necessity and the query bound. These standard conformance arguments do not make finite testing universally sufficient for unconstrained neural models.

Corollary 12 (Structure reduces audit size)**.** *Assume all R local states are reachable, A primitive tokens, K independent keys, and no trusted row values. Worst-case complete point-audit sizes are*

$$\underbrace{KAR^K}_{\text{unstructured joint table}}, \qquad \underbrace{KAR}_{\text{known keywise factorization}}, \qquad \underbrace{AR}_{\text{known factorization and sharing}}. \tag{8}$$

The bounds require the stated structural promises; they do not measure the cost of verifying those promises from arbitrary source code.

PatchWorld has 75 local rows for any K: 25 reset and 50 arithmetic rows. If all reset rows are trusted and correctly implemented, only the remaining 50 require independent row checks. A reported 75/75 audit is a complete table check, not a 75-query lower bound in the presence of that prior. Partial audits can instead abstain before an uncertified row and have zero semantic error on covered words (proposition 19).

4.3 Projection preserves the recovered transition system

Proposition 13 (An explicit soft-execution separation)**.** *For any $0 < \varepsilon < 1/2$, consider the true two-state identity primitive and start state 0. The stochastic kernel*

$$P = \begin{pmatrix} 1-\varepsilon & \varepsilon \\ \varepsilon/3 & 1-\varepsilon/3 \end{pmatrix}, \qquad (\delta_0 P^L)(0) = \tfrac{1}{4} + \tfrac{3}{4}(1-4\varepsilon/3)^L \tag{9}$$

has the correct argmax in every row, with true-successor mass at least $1-\varepsilon$. Nevertheless its decoded answer becomes wrong for $L > \log(1/3)/\log(1-4\varepsilon/3)$, whereas its projected table is exact for all L.

The projected table in this construction is exact at every length, while the soft-execution error appears at a horizon of order $1/\varepsilon$. The separation is a proved possibility, not a claim about all stochastic rollouts. More generally, true-path mass is at least $\prod_t (1-\varepsilon_t)$; this lower bound alone does not determine the decoded accuracy. Universal point-mass exactness and a logit-margin certificate for projection are given in proposition 22.

5 Controlled evidence for compositional execution

Setting and evaluation. We evaluate on synthetic PatchWorld programs with explicit semantic transitions. Each of six keys has state in $\mathbb{Z}_5$; typed resets, additions, and multiplications form 15 tokens and 75 semantic rows. For $x \in \mathbb{Z}_5$, the 15 typed primitive tokens act as follows:

$$\mathrm{SET}_v(x) = v, \tag{10}$$

$$\mathrm{ADD}_v(x) = x + (v \bmod 4) + 1 \pmod 5, \tag{11}$$

$$\mathrm{MUL}_v(x) = x((v \bmod 3) + 2) \pmod 5, \tag{12}$$

for $v \in \mathbb{Z}_5$. The arithmetic arguments are nonidentity, and addition and multiplication are generally noncommutative.

Standard local training chains contain reset followed by one addition and one multiplication in either order. The main split covers all 250 local token traces but no algebra test key-tuple. Its main test measures scheduling and global recombination of known local programs. Held-out composition removes ADD→MUL; length tests extend local chains from 3 to 16. The out-of-distribution reset test (SET-OOD) moves reset away from its training position (section B).

The symmetry and transition studies report 804 and 443 completed runs, respectively. The matched study contains 189 distinct completed runs, seven held-out seeds, and validation-only learning-rate

Table 1: **Vertical evaluations and TOF factor ablation.** Transition-study results. Audit is the fraction of 75 primitive rows with correct projected successor. Values are means over five seeds except Main/SET-OOD (ten).

(a) Vertical evaluation

Model	Main	SET-OOD	Held-out comp.	$L4$	$L8$	$L16$
Stable-sort GRU	.9947	.0542	.0000	.0000	.0000	.0000
TSF shared	1.0000	.0930	.0001	.0000	.0001	.0001
TSF unshared	1.0000	.0938	.0000	.0000	.0000	.0000
TOF projected	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000

(b) TOF factor ablation

TOF variant	Main	SET-OOD	Audit	$L4$	$L12$
Final state only	.2358	.1836	.2507	.0101	.0000
Final + typed reset	.6180	.6089	.7333	.6000	.5977
Intermediate, no reset prior	1.0000	.9857	.7813	.2753	.0001
Intermediate + prior + projection	1.0000	1.0000	1.0000	1.0000	1.0000

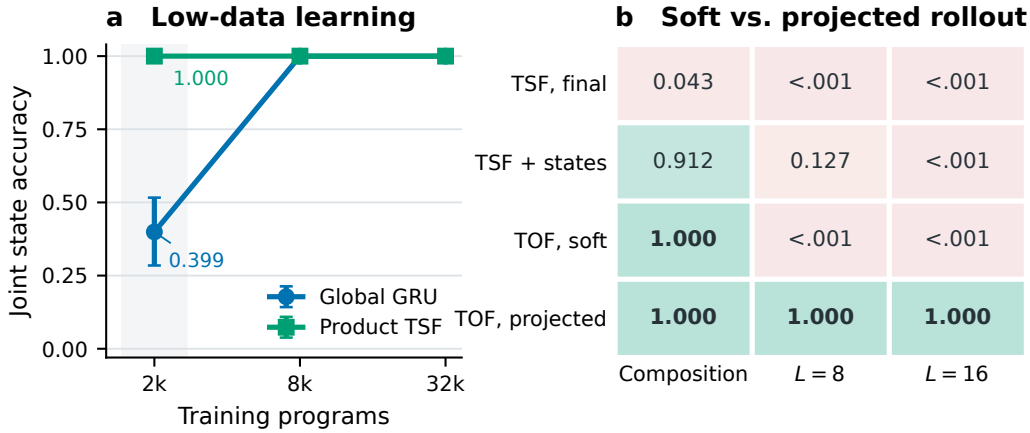

Figure 2: **Low-data learning and compositional execution.** (a) Product-state TSF learns more reliably at 2,048 programs; exact-canonical global recurrence nearly saturates with more data. Bars show 95% bootstrap intervals. (b) In the held-out-composition regime, intermediate labels improve short composition, while projected TOF maintains accuracy at longer lengths. TOF soft and projected use identical learned scores. Seven evaluation seeds; all values are joint accuracy, not probability mass.

selection. Its per-run records support independent aggregation. Complete numerical tables appear in sections C and D; implementation details and the record-to-table mapping are provided with the supplementary code.

Symmetry and primitive recovery. The six-key overwrite ceiling is $2^{-6} = 0.015625$; the Deep Sets result is 0.0157. Stable sorting removes schedule sensitivity without imposing full set symmetry. The TOF ablation reports row accuracies 0.2507 (final only), 0.7333 (plus reset prior), 0.7813 (intermediate states without prior), and 1.0000 (both, projected). These recovered-row fractions quantify learning success under the stated supervision and prior. Corollary 9 shows why terminal labels can identify arithmetic without intermediate supervision, while reset-first support leaves noninitial reset behavior unspecified without a prior. Together, the results distinguish primitive identifiability from recovery by the training procedure. Table 1 gives the complete transition-study ablation and vertical evaluations.

Table 2: **Projection preserves long execution.** Held-out ADD→MUL regime, seven seeds, 65,536 training programs. Joint goal accuracy; higher is better. TOF also fixes reset semantics. The isolated intervention is soft versus projected TOF, which share learned scores. Full standard-regime and auxiliary-control results remain in section D.

Model	Main	Composition	$L = 8$	$L = 16$
TSF, final labels	.5215	.0433	.0001	.0001
TSF, intermediate labels	.9763	.9118	.1274	.0008
TOF, soft rollout	1.0000	1.0000	.0001	.0001
TOF, projected same scores	1.0000	1.0000	1.0000	1.0000

Table 3: Mechanism-specific paired comparisons from the seven matched seeds. Confidence intervals are exhaustive paired-bootstrap percentiles; p is the exact two-sided sign-flip value.

Comparison (A − B)	Mean gap	95% CI	Exact p
TSF final − global GRU, 2,048 programs	.600830	[.483582,.714948]	.015625
TOF projected − TSF + states, held-out comp.	.088248	[.019871,.171265]	.250000
TOF projected − TSF + states, held-out $L16$	.999180	[.998823,.999503]	.015625
TOF projected − TOF soft, held-out $L16$	.999904	[.999834,.999974]	.015625

5.1 PRODUCT STATE IMPROVES LOW-DATA LEARNING

At 2,048 programs, the exact-canonical global GRU obtains 0.3992 joint goal accuracy (JGA; 95% confidence interval (CI) $[0.2845, 0.5164]$), versus 1.0 for final-only shared TSF. The paired gap is 0.6008 (CI $[0.4836, 0.7149]$, exact two-sided sign-flip $p = 0.0156$). Parameters are comparable (50,382 versus 43,253), but TSF stores six hidden states rather than one (576 versus 96 recurrent state elements). At 8,192 programs the global model reaches 0.9997.

With 98,304 programs and global hidden sizes 96–1024, canonicalized GRUs attain 0.999826–0.999895 at $K = 16$; product models remain 1.0. These controls locate the benefit of product organization in low-data learning and explicit semantic-state structure, rather than a representational impossibility for global recurrence. Resource comparisons account for both learned parameters and recurrent state.

5.2 INTERMEDIATE LABELS IMPROVE COMPOSITION; PROJECTION PRESERVES EXECUTION

Table 2 gives the strongest matched recurrent comparison. Exact intermediate targets raise held-out-composition JGA to 0.9118, a substantial improvement over final-only supervision. Its JGA is 0.1274 at length 8 and 0.0008 at length 16. Projected TOF is 1.0 throughout. Its composition advantage has mean 0.0882 and bootstrap CI $[0.0199, 0.1713]$, but the exact paired test has $p = 0.25$ because four seeds tie. This composition comparison is descriptive rather than statistically significant. At length 16 the paired gap is 0.99918 (CI $[0.99882, 0.99950]$, $p = 0.0156$).

The cleanest intervention changes no learned parameters. Both TOF modes are perfect at training length and held-out composition. At length 16, projection raises JGA from 9.6×10^{-5} under soft rollout to 1.0 for every seed (paired gap 0.99990, CI $[0.99983, 0.99997]$, $p = 0.0156$). This isolates the execution rule for these checkpoints. Proposition 13 supplies an analytic example of the same execution distinction. Tests in table 3 are unadjusted, mechanism-specific comparisons.

History-conditioned next-state and distributional diagnostics in section D.3 further separate prediction behavior from fixed transition rows. For certification, theorem 11 requires checking the actual deployed table: a complete correct 75-row check then establishes all-prefix execution. The supplementary row-audit utility compares each candidate table against the semantic reference.

6 RELATED WORK

Symmetry and semantic memory. Deep Sets, Set Transformer, and Janossy pooling target full permutation invariance (Zaheer et al., 2017; Lee et al., 2019; Murphy et al., 2019); partially exchange-

able networks, probabilistic symmetry theory, and learned canonicalization address other symmetry classes (Wiqvist et al., 2019; Bloem-Reddy & Teh, 2020; Kaba et al., 2023). The subset-invariant regularizer (SIRE) compares swapped recurrent updates (Cohen-Karlik et al., 2020). Trace monoids and asynchronous automata provide the classical concurrency foundation (Mazurkiewicz, 1987; Diekert & Rozenberg, 1995; Zielonka, 1987), while conflict-free replicated data types (CRDTs) engineer convergence under reordering (Shapiro et al., 2011). These established ideas provide the semantic interface for our analysis of identification and execution.

Discrete algorithm learning and verification. Regular-language and algorithmic-reasoning studies examine length generalization, intermediate hints, and causal alignment (Dan et al., 2022; Veličković et al., 2022; Ibarz et al., 2022; Bevilacqua et al., 2023; Rodionov & Prokhorenkova, 2023). We also build on the distinction between algorithmic Markov states and historical neural embeddings emphasized by Bohde et al. (2024); decoded-state closure makes the required quotient condition explicit. Discrete Neural Algorithmic Reasoning (DNAR) uses finite states, argmax inference, and exhaustive checks of discrete transitions to prove learned algorithm correctness (Rodionov & Prokhorenkova, 2025). Building on this discrete verification perspective, our contribution is the exact composition-order ambiguity and terminal-confidence condition, together with the distinction between decoded semantic closure and finite intermediate accuracy. The audit-size comparison makes explicit the benefit and cost assumptions of known keyed structure. Automata extraction uses a recurrent model as an oracle (Weiss et al., 2018); our correctness oracle is the stipulated task semantics. General conformance testing handles unknown implementation behavior under explicit fault domains (Vaandrager & Melse, 2025), whereas TOF exposes its local transition table directly.

Identifying compositional structure. Schug et al. (2024) relate compositional generalization to identification in modular teacher–student systems, obtaining module recovery up to linear transformations under support and architecture conditions. Our result concerns anchored finite-state function composition: its remaining ambiguity is a common centralizer, and strict terminal-majority conditions certify discrete projected transitions. Both analyses distinguish identifiability under a specified fit from attaining that fit through optimization.

Stateful applications. Dialogue-state tracking, order-independent prompt subsequences, and multi-turn degradation motivate stateful histories (Henderson et al., 2014; McIlroy-Young et al., 2024; Laban et al., 2026). They motivate our typed-event setting, whose guarantees begin after the events, dependencies, and state meaning have been specified.

7 Limitations and Conclusion

Our guarantees assume finite labeled semantic states, trusted access/reset behavior where stated, and genuinely independent keys. The composition-order theorem concerns permutation families with complete composition support, and its projection guarantee requires uniform terminal probabilities rather than classification accuracy alone. Known product structure, parameter sharing, semantic oracles, and routing must be established separately from the row audit. The experiments isolate these mechanisms in synthetic PatchWorld; transfer to learned routing, latent semantic states, and natural-language inputs remains a direction for further study.

The results give a constructive route from trace invariance to certified execution. Terminal composition labels can identify primitive maps through their common centralizer. Decoded semantic closure turns these maps into a finite auditable interface, and known product structure reduces verification to a local table. Discrete projection then preserves the identified or audited machine at every finite prefix. The experiments demonstrate the complementary benefits of product-state learning, intermediate supervision, and projected execution, with perfect projected joint accuracy through length 16 across seven matched seeds.

Reproducibility Statement. We provide anonymized source code, configurations, aggregate run records, and verification scripts for the results reported in this paper in the supplementary material.

AI USE STATEMENT

The author developed the research idea and proofs. AI tools provided coding, debugging, writing, and editing support, including polishing the exposition and presentation of the proofs. The author reviewed and verified the resulting material and retains responsibility for the research questions, methods, proofs, experiments, interpretation, and final content.

REFERENCES

Beatrice Bevilacqua, Kyriacos Nikiforou, Borja Ibarz, Ioana Bica, Michela Paganini, Charles Blundell, Jovana Mitrovic, and Petar Veličković. Neural algorithmic reasoning with causal regularisation. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp.
2272–2288, 2023.

Benjamin Bloem-Reddy and Yee Whye Teh. Probabilistic symmetries and invariant neural networks. *Journal of Machine Learning Research*, 21(90):1–61, 2020.

Montgomery Bohde, Meng Liu, Alexandra Saxton, and Shuiwang Ji. On the Markov property of neural algorithmic reasoning: Analyses and methods. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Kn7tWhuetn.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp.
1724–1734. Association for Computational Linguistics, 2014. doi: 10.3115/v1/D14-1179.

Edo Cohen-Karlik, Avichai Ben David, and Amir Globerson. Regularizing towards permutation invariance in recurrent models. In *Advances in Neural Information Processing Systems*, volume 33, pp.
18364–18374, 2020.

Soham Dan, Osbert Bastani, and Dan Roth. Understanding robust generalization in learning regular languages. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp.
4630–4643, 2022.

Volker Diekert and Grzegorz Rozenberg (eds.). *The Book of Traces*. World Scientific, 1995.

Matthew Henderson, Blaise Thomson, and Jason D. Williams. The second dialog state tracking challenge. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pp.
263–272, 2014.

Borja Ibarz, Vitaly Kurin, George Papamakarios, Kyriacos Nikiforou, Mehdi Bennani, Róbert Csordás, Andrew Joseph Dudzik, Matko Bošnjak, Alex Vitvitskyi, Yulia Rubanova, Andreea Deac, Beatrice Bevilacqua, Yaroslav Ganin, Charles Blundell, and Petar Veličković. A generalist neural algorithmic learner. In *Proceedings of the First Learning on Graphs Conference*, volume 198 of *Proceedings of Machine Learning Research*, pp.
2:1–2:23, 2022.

Sékou-Oumar Kaba, Arnab Kumar Mondal, Yan Zhang, Yoshua Bengio, and Siamak Ravanbakhsh. Equivariance with learned canonicalization functions. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp.
15546–15566, 2023.

Philippe Laban, Hiroaki Hayashi, Yingbo Zhou, and Jennifer Neville. LLMs get lost in multi-turn conversation. In *International Conference on Learning Representations*, 2026.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam R. Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3744–3753, 2019.

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

Antoni Mazurkiewicz. Trace theory. In Wilfried Brauer, Wolfgang Reisig, and Grzegorz Rozenberg (eds.), *Petri Nets: Applications and Relationships to Other Models of Concurrency*, volume 255 of *Lecture Notes in Computer Science*, pp. 278–324. Springer, 1987. doi: 10.1007/3-540-17906-2_30.

Reid McIlroy-Young, Katrina Brown, Conlan Olson, Linjun Zhang, and Cynthia Dwork. Order-independence without fine tuning. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Ryan L. Murphy, Balasubramaniam Srinivasan, Vinayak Rao, and Bruno Ribeiro. Janossy pooling: Learning deep permutation-invariant functions for variable-size inputs. In *International Conference on Learning Representations*, 2019.

Anil Nerode. Linear automaton transformations. *Proceedings of the American Mathematical Society*, 9(4):541–544, 1958.

Gleb Rodionov and Liudmila Prokhorenkova. Neural algorithmic reasoning without intermediate supervision. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Gleb Rodionov and Liudmila Prokhorenkova. Discrete neural algorithmic reasoning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 51851–51865, 2025.

Simon Schug, Seijin Kobayashi, Yassir Akram, Maciej Wołczyk, Alexandra Proca, Johannes von Oswald, Razvan Pascanu, João Sacramento, and Angelika Steger. Discovering modular solutions that generalize compositionally. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=H98CVcX1eh.

Marc Shapiro, Nuno Preguiça, Carlos Baquero, and Marek Zawirski. Conflict-free replicated data types. In *Stabilization, Safety, and Security of Distributed Systems*, pp. 386–400. Springer, 2011.

Frits Vaandrager and Ivo Melse. New fault domains for conformance testing of finite state machines. In *36th International Conference on Concurrency Theory (CONCUR 2025)*, volume 348 of *Leibniz International Proceedings in Informatics*, pp. 34:1–34:22. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2025. doi: 10.4230/LIPIcs.CONCUR.2025.34.

Petar Veličković, Adrià Puigdomènech Badia, David Budden, Razvan Pascanu, Andrea Banino, Misha Dashevskiy, Raia Hadsell, and Charles Blundell. The CLRS algorithmic reasoning benchmark. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 22084–22102, 2022.

Gail Weiss, Yoav Goldberg, and Eran Yahav. Extracting automata from recurrent neural networks using queries and counterexamples. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 5247–5256. PMLR, 2018.

Samuel Wiqvist, Pierre-Alexandre Mattei, Umberto Picchini, and Jes Frellsen. Partially exchangeable networks and architectures for learning summary statistics in approximate bayesian computation. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings*

of Machine Learning Research, pp.
6798–6807, 2019.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Póczos, Ruslan Salakhutdinov, and Alexander J. Smola. Deep sets. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Wiesław Zielonka. Notes on finite asynchronous automata. *RAIRO – Theoretical Informatics and Applications*, 21(2):99–135, 1987.

A COMPLETE PROOFS

A.1 TRACE SYMMETRY

Proof of proposition 2. Condition on the unordered pair of distinct overwrite values at every key. There are 2^K possible vectors of local orders. They all induce the same event multiset, while each order vector produces a different final ledger. Let M denote the common multiset and $Y_1, \dots, Y_{2^K}$ the distinct labels. A randomized permutation-invariant predictor chooses a distribution $p(\cdot \mid M)$ independent of the order vector. Its average exact-match probability is

$$2^{-K} \sum_{i=1}^{2^K} p(Y_i \mid M) \le 2^{-K},$$

because the Y_i are distinct and probabilities sum to at most one. Averaging over the value pairs preserves the bound. □

Proof of proposition 1. A legal adjacent swap exchanges events on different keys and therefore preserves every ordered projection π_k. Hence $h \equiv_I h'$ implies $\Pi(h) = \Pi(h')$.

Conversely, stably sort any history by key using adjacent inversion swaps. Every exchanged pair has different keys, so every swap is legal under eq. (1); stability preserves within-key order. For a local word $u = a_1 \cdots a_m$, let $\iota_k(u) = (k, a_1) \cdots (k, a_m)$ restore its key tags. Thus every history is trace-equivalent to the canonical representative $\iota_1(\pi_1(h)) \cdots \iota_K(\pi_K(h)) \in \Sigma^*$. Histories with the same projection tuple have the same representative and are trace-equivalent. For a trace-invariant f, define $\bar{f}(\Pi(h)) = f(h)$. The equivalence makes $\bar{f}$ well-defined and gives $f = \bar{f} \circ \Pi$. □

Proof of theorem 3. Assume the commutation identities. It suffices to show invariance under one generating swap. Let $h = uabv$ and $h' = ubav$ with $(a, b) \in I$, and let $z = F(u) \in \mathcal{R}$. Immediately after the exchanged pair, the states are $T_b T_a(z)$ and $T_a T_b(z)$, equal by assumption. Applying the common suffix v preserves equality. Chaining generating swaps proves trace invariance.

Conversely, suppose F is trace-invariant. For any reachable $z = F(u)$ and $(a, b) \in I$, the histories uab and uba are trace-equivalent. Therefore $T_b T_a(z) = F(uab) = F(uba) = T_a T_b(z)$. □

Proposition 14 (Approximate trace stability)**.** *Suppose every T_e is κ-Lipschitz and*

$$\delta_I = \sup_{z \in \mathcal{R}, (a,b) \in I} d(T_b T_a(z), T_a T_b(z)) < \infty,$$

with $\delta_I = 0$ when $I = \varnothing$. For a legal swap path P from h to h', let r_j be the number of events after swap j. Then

$$d(F(h), F(h')) \le \delta_I \sum_{j \in P} \kappa^{r_j}.$$

The infimum over legal swap paths also bounds the distance.

Proof. A single generating swap creates discrepancy at most δ_I immediately after the exchanged pair. The common suffix of length r_j amplifies it by at most κ^{r_j}. Insert the intermediate histories along P and apply the triangle inequality. Taking the infimum over paths can only tighten the bound. □

A.2 CONTINUATION QUOTIENTS AND CANONICAL PRODUCT MINIMALITY

For completeness, write δ^* for the extension of a transition map to words. A Moore-machine homomorphism preserves the initial state, output, and transition under every event.

Lemma 15 (Local continuation quotient)**.** *For $g : \mathcal{A}^* \to \mathcal{Y}$, the relation $\sim_g$ in eq. (2) is a right congruence. A reachable deterministic Moore machine computes g only if equality of its states refines $\sim_g$. If $\sim_g$ has finite index, the quotient machine with states $[u]_{\sim_g}$ is the unique minimal reachable exact executor up to isomorphism.*

Proof. The relation is an equivalence. It is right invariant because $u \sim_g v$ implies, for any $a \in \mathcal{A}$ and $w \in \mathcal{A}^*$, $g(uaw) = g(vaw)$, so $ua \sim_g va$.

Let $M = (Q, q_0, \delta, o)$ compute g. If $\delta^*(q_0, u) = \delta^*(q_0, v)$, then every continuation reaches the same state from this common state, hence $g(uw) = g(vw)$ for all w. Thus machine-state equality refines $\sim_g$, and M has at least as many reachable states as the congruence index.

When the index is finite, define $Q_g = \mathcal{A}^*/\sim_g$, initial state $[\epsilon]$, transition $\delta_g([u], a) = [ua]$, and output $o_g([u]) = g(u)$. Right invariance and the definition of $\sim_g$ make these maps well-defined. Induction gives $\delta_g^*([\epsilon], u) = [u]$, so the quotient computes g. The preceding lower bound proves minimality. Any two reachable minimal machines induce exactly the continuation classes and are isomorphic by mapping the state reached by u to $[u]$. □

Proof of theorem 4. Define global continuation equivalence by $h \sim_G h'$ iff $G(hc) = G(h'c)$ for every keyed continuation $c \in \Sigma^*$. We first show that Φ indexes these classes exactly.

Suppose $\Phi(h) = \Phi(h')$. Projection distributes over concatenation: $\pi_k(hc) = \pi_k(h)\pi_k(c)$. Local continuation equivalence gives $g_k(\pi_k(h)\pi_k(c)) = g_k(\pi_k(h')\pi_k(c))$ for every key, so $G(hc) = G(h'c)$ and $h \sim_G h'$.

Conversely, if $\Phi(h) \neq \Phi(h')$, then for some key j, $\pi_j(h) \not\sim_{g_j} \pi_j(h')$. There exists a local word w such that $g_j(\pi_j(h)w) \neq g_j(\pi_j(h')w)$. Address every symbol of w to key j to form a keyed continuation c. The jth coordinates of $G(hc)$ and $G(h'c)$ differ, so $h \not\sim_G h'$. Hence the global quotient is in bijection with the image of Φ.

Every tuple in $Q_1 \times \cdots \times Q_K$ is reachable. Choose a representative u_k for each desired local class and concatenate the keyed words (k, u_k) over $k = 1, \ldots, K$. Its projections are the selected representatives. Thus the quotient has exactly $\prod_k r_k$ states. The product machine updates only the addressed quotient coordinate and outputs the tuple of local outputs, so it computes G. By lemma 15 applied globally, it is minimal.

It remains to prove the canonical factor statement. Let $M = (Q_M, q_0, \delta_M, o_M)$ be any reachable deterministic Moore machine computing G. For a reachable state $q = \delta_M^*(q_0, h)$, define $\psi(q) = \Phi(h)$. If the same state is reached by h', then for every continuation c, determinism gives identical future machine states and outputs, so $G(hc) = G(h'c)$. The first part implies $\Phi(h) = \Phi(h')$; therefore ψ is well-defined. It is surjective because every product class is represented by a history. It preserves outputs because M computes G, and for $e = (k, a)$,

$$\psi(\delta_M(q, e)) = \Phi(he) = \delta_\Phi(\Phi(h), e) = \delta_\Phi(\psi(q), e).$$

All states of M are reachable, so this identity and $\psi(q_0) = \Phi(\epsilon)$ determine ψ uniquely. Any exact executor therefore maps onto the product quotient, and a minimal one must be isomorphic to it. □

Proof of corollary 5. Every value $x \in \mathbb{Z}_5$ is reachable by the one-event word $\textsc{Set}_x$. Prefixes ending at distinct values are distinguished by the empty continuation because the semantic value itself is output. Prefixes ending at the same value have identical behavior under all future primitives because every transition is a deterministic function of only the current value and next primitive. Thus there are exactly five local continuation classes. Apply theorem 4. □

Proposition 16 (Function-class hierarchy)**.** *Let a finite local-trace domain S have size M, each key output alphabet Y have size V, and $K, M, V \geq 2$. Let $\mathcal{F}_{\text{trace}}$ contain arbitrary maps from S^K to Y^K, $\mathcal{F}_{\text{prod}}$ contain keywise product maps, and $\mathcal{F}_{\text{shared}}$ use one shared local map. Then*

$$\mathcal{F}_{\text{shared}} \subsetneq \mathcal{F}_{\text{prod}} \subsetneq \mathcal{F}_{\text{trace}}, \quad |\mathcal{F}_{\text{shared}}| = V^M, \quad |\mathcal{F}_{\text{prod}}| = V^{KM}, \quad |\mathcal{F}_{\text{trace}}| = V^{KM^K}.$$

For a fixed initial state, a deterministic transition system with R local states and A primitives induces at most $R^{AR}V^R$ shared local functions.

Proof. The inclusions follow from the definitions. The first is strict by choosing different local maps at two keys; the second by allowing one output coordinate to depend on another key's trace. Counting assigns one of V outputs to each input in the shared and product classes. An arbitrary trace function assigns one of V^K output tuples to each of M^K input tuples. Finally, each of A deterministic maps $[R] \to [R]$ has R^R choices and the readout has V^R choices. □

Proposition 17 (Shared support coverage)**.** *Suppose all nK local traces in n global examples are independent draws from a distribution P on support size M with minimum nonzero mass $p_{\min}$. Assume a shared deterministic target and a lookup that predicts every observed trace correctly. A shared lookup pools nK observations, while key-specific lookups use n observations per key. Then*

$$\Pr[\textit{shared support incomplete}] \le Me^{-nKp_{\min}},$$
$$\Pr[\textit{some key support incomplete}] \le KMe^{-np_{\min}}.$$

For a fresh test tuple independent of training, with each coordinate marginal P, their expected joint errors are at most

$$K\sum_s P(s)(1-P(s))^{nK}, \qquad K\sum_s P(s)(1-P(s))^{n},$$

respectively.

Proof. A support point s is absent from m independent samples with probability $(1-P(s))^m \le e^{-mP(s)}$. Union bounds give the coverage statements. Conditional on training, a fresh K-tuple has joint error at most the sum of its coordinates' unseen-support probabilities. Take expectations and use $\Pr[s \text{ unseen}] = (1-P(s))^m$. □

A.3 Transition actions, identifiability, and audit certificates

Proposition 18 (Product transitions define a trace action)**.** *For local maps $\{\tau_a\}_{a\in\mathcal{A}}$, let a global event (k,a) update coordinate k of $[R]^K$ by τ_a and leave the others unchanged. The generator assignment extends uniquely to an action of $M(\Sigma, I)$ on $[R]^K$.*

Proof. Updates on distinct coordinates commute. Hence the free-monoid action respects every defining relation $uabv = ubav$ for independent events and factors uniquely through the trace quotient. □

Proof of theorem 6. Assume $R \ge 2$. Every row in $\mathcal{C}$ is fixed to its true successor. With no further constraints, each of the remaining $R|\mathcal{A}| - |\mathcal{C}|$ rows can take any of R values independently, yielding the stated count.

If $\mathcal{E}_\tau \subseteq \mathcal{C}$, every compatible table agrees with τ along every execution from the initial state. Induction on prefix length gives the same semantic state for every word, so universal execution is identified.

For necessity, let (x,a) be a reachable uncovered row. Choose a shortest word u reaching x. Its path cannot traverse (x,a), because the prefix immediately before such a traversal would already reach x and be shorter than u. Choose $x' \ne \tau_a(x)$ and define $\widetilde{\tau}$ to equal τ except that $\widetilde{\tau}_a(x) = x'$. This violates no observation or prior. The word u still reaches x under $\widetilde{\tau}$, but ua reaches x' rather than $\tau_a(x)$, so universal execution is not identified. □

Proof of theorem 11: correctness characterization. If $\widehat{\tau}$ agrees with τ on every reachable row, induct on prefix length. The two executions start at the same state. If they agree before primitive a, the common state is reachable and the relevant row is correct, so they agree afterward. Thus every prefix of every word is exact.

Conversely, suppose every prefix of every word is exact. For any reachable x, choose a word u with $\tau^*(x_0,u) = x$. Prefix exactness gives $\widehat{\tau}^*(x_0,u) = x$. For any $a \in \mathcal{A}$, exactness after ua yields $\widehat{\tau}_a(x) = \tau_a(x)$. Hence every reachable row is correct. The same induction restricted to one fixed word proves the pathwise statement. □

Proof of theorem 11: query lower bound. Assume $R \ge 2$. The auditor knows the true reference machine but accesses candidate successors only through point queries, with no side information fixing or tying rows. Run a sound and complete deterministic auditor on the correct candidate $\widehat{\tau} = \tau$. Suppose it accepts without querying a true-reachable row (x,a). A shortest true word u reaching x cannot traverse this row: otherwise its prefix before that traversal would already reach x and be shorter. Change only the unqueried candidate successor to a different state. Its answers on the full

query transcript are unchanged, so the auditor still accepts; yet it remains correct through u and fails after ua. This contradicts soundness. Querying all true-reachable rows attains the bound. An equivalent formulation gives the auditor an explicit candidate and only point access to an otherwise unconstrained reference table. Neither model permits uncharged inference from additional structural or compositional information. □

Proposition 19 (Selective prefix certificate)**.** *Let $\mathcal{C} \subseteq \mathcal{E}_\tau$ be rows verified to satisfy $\widehat{\tau}_a(x) = \tau_a(x)$. Define a selective executor that starts at x_0, applies $\widehat{\tau}_a(x)$ only when the current row (x, a) lies in $\mathcal{C}$, and otherwise abstains before emitting the next state. Every state emitted before abstention is exact. For any distribution D over words, its coverage is*

$$\Pr_{w \sim D}\big[\textit{every true path row of } w \textit{ lies in } \mathcal{C}\big],$$

with no semantic errors on covered words. Thus its conditional semantic error is zero whenever coverage is positive. For fixed length L, the abstention probability is at most $\sum_{t=1}^{L} \Pr[(X_{t-1}, A_t) \notin \mathcal{C}]$.

Proof. Induct until the first uncertified row. Initially the candidate and true states agree. If the current row is certified, the candidate transition equals the true transition, so the states continue to agree. Hence every emitted prefix is exact. The executor covers a word exactly when no true path row leaves $\mathcal{C}$; because the states agree until that event, the candidate-side test detects the same first uncertified row. The union bound gives the final display. □

A.4 Identifiability from ordered compositions

Throughout this subsection, X is a finite labeled set, S, T are nonempty, and A_s, B_t are permutations. Products of functions act right to left. All composition observations include every x, s, t; absent combinations cannot simply be filled in by assumption.

The complete proof of theorem 7 appears in section 3.

The ambiguity in this proof is not a simultaneous conjugation of every operation and output. It inserts a permutation between two operation families while keeping the reset-access states and terminal labels fixed. Consequently, one-order terminal success need not imply correct intermediate outputs or reverse-order behavior. The reverse-order observations remove this ambiguity exactly to the extent measured by the common centralizer.

Lemma 20 (High-confidence stochastic composition has consistent hard factors)**.** *Let P, Q be row-stochastic $R \times R$ matrices and f a permutation of X. If*

$$(PQ)[x, f(x)] \geq 1 - \varepsilon \quad \textit{for all } x, \qquad 0 \leq \varepsilon < 1/3,$$

then the row-wise argmax maps p, q of P, Q are uniquely defined permutations and $q \circ p = f$. More precisely, there is a bijection $c : X \to X$ such that

$$Q[c(x), f(x)] \geq 1 - \varepsilon, \qquad P[x, c(x)] \geq \frac{1 - 2\varepsilon}{1 - \varepsilon} > \frac{1}{2}.$$

Proof. For every x, the target coordinate of the product is a convex combination:

$$\sum_y P[x, y] Q[y, f(x)] \geq 1 - \varepsilon.$$

Hence some $c(x)$ satisfies $Q[c(x), f(x)] \geq 1 - \varepsilon$. Distinct x, x' have distinct target columns. Since $2(1 - \varepsilon) > 1$, the same row of Q cannot serve two such targets. Thus c is injective, and hence bijective. Each row of Q has a unique entry at least $1 - \varepsilon > 1/2$, so its argmax is $q(c(x)) = f(x)$.

For $y \neq c(x)$, write $y = c(x')$ with $x' \neq x$. That row assigns at least $1 - \varepsilon$ to $f(x')$, so $Q[y, f(x)] \leq \varepsilon$. Therefore

$$1 - \varepsilon \leq (PQ)[x, f(x)] \leq P[x, c(x)] + \varepsilon(1 - P[x, c(x)]).$$

Rearranging gives the stated bound. It is greater than $1/2$ exactly when $\varepsilon < 1/3$, so $p(x) = c(x)$ uniquely and $q(p(x)) = f(x)$. □

Lemma 21 (Majority-product rigidity)**.** *Suppose row-stochastic matrices P, Q have permutation-valued majority maps p, q, meaning $P[x, p(x)] > 1/2$ and $Q[y, q(y)] > 1/2$ for all states. If PQ also has a permutation-valued majority map f, then $f = q \circ p$.*

Proof. Relabel the intermediate coordinates by p and the output coordinates by $q \circ p$. Equivalently, let Π_p, Π_q be the row-action permutation matrices and set

$$\widetilde{P} = P\Pi_p^{-1}, \qquad \widetilde{Q} = \Pi_p Q \Pi_q^{-1} \Pi_p^{-1}.$$

Both have diagonal entries greater than $1/2$, while their product is $PQ(\Pi_p\Pi_q)^{-1}$. Its majority map is the permutation $r = (q \circ p)^{-1} \circ f$. We show r must be identity.

Write $a_i = \widetilde{P}[i, i] > 1/2$ and $b_i = \widetilde{Q}[i, i] > 1/2$. For any $j \neq i$, the diagonal majority in each row of $\widetilde{Q}$ gives $\widetilde{Q}[i, j] \leq 1 - b_i$ and $\widetilde{Q}[k, j] \leq b_j$ for every $k \neq i$: when $k \neq j$, its off-diagonal entry is less than $1/2 < b_j$. Therefore

$$(\widetilde{P}\widetilde{Q})[i, j] \leq a_i(1 - b_i) + (1 - a_i)b_j < \frac{1}{2} + \frac{b_j - b_i}{2},$$

because $b_i + b_j > 1$ and $a_i > 1/2$. If $j = r(i) \neq i$, the left side exceeds $1/2$, so $b_{r(i)} > b_i$. A nontrivial cycle of the permutation r would require a strict increase around a closed cycle, which is impossible. Thus $r = \mathrm{id}$ and $f = q \circ p$. □

Proof of theorem 8. Let P_s implement candidate U_s and Q_t implement candidate V_t. With row-vector distributions, the two execution orders satisfy

$$(P_sQ_t)[x, B_tA_s(x)] \geq 1 - \varepsilon > 1/2, \qquad (Q_tP_s)[x, A_sB_t(x)] \geq 1 - \varepsilon > 1/2$$

for every x, s, t. For any fixed pair, the first product is a convex combination of rows of Q_t. For each distinct target $B_tA_s(x)$ there must be a row of Q_t assigning it probability at least $1 - \varepsilon$. No row can supply two targets with probability greater than $1/2$. There are R targets and R rows, so all rows of Q_t have distinct majority successors. The reverse-order product gives the same conclusion for every row of P_s. Thus both kernels have unique permutation-valued argmax maps, each with selected probability at least $1 - \varepsilon$.

Write those projected maps as $\widehat{U}_s, \widehat{V}_t$. Applying lemma 21 to each product gives

$$\widehat{V}_t\widehat{U}_s = B_tA_s, \qquad \widehat{U}_s\widehat{V}_t = A_sB_t.$$

The projections are unique and therefore independent of which partner matrix is used in the proof. They satisfy theorem 7, so $\widehat{U}_s = CA_s$ and $\widehat{V}_t = B_tC^{-1}$ for one $C \in \mathcal{Z}$. A trivial centralizer forces $C = \mathrm{id}$. Trusted reset operations are correct as well, and induction proves exact projected execution of every finite word. Keywise product execution extends the result to any number of keys. □

What the bound does and does not require. The theorem allows arbitrary stochastic primitive kernels, not just perturbations whose correct rows are already known. One-order confidence below error $1/3$ already suffices for consistent hard factors by lemma 20; the second order both relaxes this to strict terminal majority and restricts the remaining gauge. It uses terminal *probabilities* at every supported anchor and pair. At $\varepsilon = 0$ it includes exact point-mass terminal matching, which forces permutation factors without presupposing deterministic candidates. Finite softmax logits cannot themselves yield exact zero probabilities, so this zero-error case describes the closure of that parameterization; the strict positive-error guarantee applies directly. Neither an average loss nor perfect argmax accuracy implies the required uniform bound. The theorem provides a sufficient finite terminal test under its uniform confidence condition, independently of how the learner is optimized.

Thresholds and strict inequalities. The strict majority requirement in lemma 21 cannot be replaced by non-strict majority: uniform two-state kernels have tied argmaxes and assign probability $1/2$ to both targets in both orders. This demonstrates the need for strictness in the general factor-decoding argument, not optimality for every particular identifiable family.

For the one-order lemma, the constant $1/3$ cannot be included in the lemma as a guarantee of a unique first-factor argmax. With target identity, take

$$P = \begin{pmatrix} 1/2 & 1/2 \\ 0 & 1 \end{pmatrix}, \qquad Q = \begin{pmatrix} 1 & 0 \\ 1/3 & 2/3 \end{pmatrix}.$$

Both diagonal entries of PQ are $2/3$, but the first row of P ties. Slightly increasing the permitted error makes its incorrect successor strictly preferred. This sharpness concerns the generic two-factor lemma, not necessity of the same threshold when both-order algebraic constraints are jointly exploited.

The complete proof of corollary 9 appears in section 3.

A.5 Decoded-state closure and the limits of finite history tests

Proof of theorem 10. If eq. (5) holds, define $\bar{\tau}_a(x) = d(T_a z)$ for any reachable z satisfying $d(z) = x$. The condition makes this independent of the representative; its value lies in $d(Z_{\text{reach}})$ because $T_a z$ is reachable. Every x in that domain has such a representative, so the map is unique and $d \circ T_a = \bar{\tau}_a \circ d$. Conversely, this identity immediately implies eq. (5).

Assume $d(z_0) = x_0$, closure, and agreement of the induced map with the true map on every true-reachable state–event row (in particular, those states are in its domain). Induction on word length then gives $d(T_w z_0) = \tau^*(x_0, w)$ for every w. Conversely, universal decoded correctness implies $d(Z_{\text{reach}})$ equals the true reachable state set. If $z = T_u z_0$ and $z' = T_v z_0$ have the same decoded state x, correctness gives $\tau^*(x_0, u) = \tau^*(x_0, v) = x$. Correctness after appending any a gives

$$d(T_a z) = \tau_a(x) = d(T_a z'),$$

which proves closure and row agreement.

For the finite-test impossibility, take any finite collection of evaluated words and let N be their maximum length. Let $\rho : X \to X$ be a permutation without fixed points, which exists for $R \geq 2$. Construct a hidden state $(x, c) \in X \times \{0, \ldots, N+1\}$, initial state $(x_0, 0)$, and transition

$$T_a(x, c) = (\tau_a(x), \min\{c+1, N+1\}).$$

Define the decoder by $d(x, c) = x$ for $c \leq N$ and $d(x, N+1) = \rho(x)$. Every tested prefix agrees with the true machine, but every word of length $N+1$ has an incorrect decoded state. Thus any procedure that accepts a correct implementation solely from finitely many such observations also accepts an incorrect implementation with the same transcript, unless its admissible implementation class excludes this construction.

The counter does not depend on event order. In the keyed setting, use the true product state and the total event count; independent true updates commute and both increment the count equally. The resulting hidden representation remains trace-invariant. Therefore exact scheduling invariance alone does not prevent the delayed error. The statement concerns unrestricted hidden-state size, not a known bounded finite-state fault domain. □

A.6 Audit complexity under explicit structural promises

Proof of corollary 12. All joint semantic states are reachable because each local state is reachable and keywise updates can be concatenated independently. An unstructured candidate maps each of R^K joint states and KA keyed inputs to one joint successor, so there are KAR^K independently testable input rows. Checking all of them is sufficient by theorem 11. It is necessary in the point-query model: on a correct candidate, change an unqueried row to a different successor. A shortest true access word to that row's source does not use the row. The modified candidate agrees with all query answers but fails on that access word followed by the row's event.

If keywise factorization of the *candidate implementation* is promised, a row is indexed only by (k, x, a); the other coordinates are known to remain unchanged and not to affect it. There are KRA such rows. Changing any unqueried one gives the same indistinguishability argument within this factorized class. If sharing is also promised, the key index is redundant and there are RA independent rows. Their sufficiency and necessity follow identically.

These are costs of checking independently unknown successor entries, not lower bounds against symbolic source inspection or against a learner given extra compositional constraints. If a subset $\mathcal{C}_0$

of rows is known to be implemented correctly, only the remaining reachable independent rows must be queried. Similarly, enforced parameter ties between semantically duplicate tokens can reduce the query space; the displayed bounds deliberately assume none beyond the stated keywise structure and sharing. □

A.7 Soft execution and robust projection

Proof of proposition 13. The true update is identity. Both diagonal entries of P are strictly greater than $1/2$, so the unique row-wise argmax map is identity. Let $m_L = (\delta_0 P^L)(0)$. Then

$$m_{L+1} = (1-\varepsilon)m_L + (\varepsilon/3)(1-m_L) = \varepsilon/3 + (1-4\varepsilon/3)m_L,$$

with $m_0 = 1$. Solving the affine recurrence gives eq. (9). Since $0 < 1 - 4\varepsilon/3 < 1$, the condition $m_L < 1/2$ is equivalent to

$$(1-4\varepsilon/3)^L < 1/3,$$

which is precisely the strict inequality on L in the proposition. At equality the two terminal states tie; beyond it the decoded answer is uniquely wrong. As $\varepsilon \downarrow 0$, the threshold is asymptotic to $3\log 3/(4\varepsilon)$. Projected identity execution is exact at every length. Only state 0 is required to be true-reachable for this example; both matrix rows nevertheless have correct argmaxes. □

Proposition 22 (Point-mass exactness, path mass, and projection margins)**.** *For row-stochastic kernels P_a, initial distribution δ_{x_0}, and true deterministic transitions τ_a, universal point-mass exactness holds iff $P_a[x,:] = \delta_{\tau_a(x)}$ on every true-reachable row. For a fixed true path whose step-t successor has probability at least $1-\varepsilon_t$, with $0 \le \varepsilon_t \le 1$, the final correct-state mass is at least $\prod_t(1-\varepsilon_t)$, and total variation from the target point mass is at most $1 - \prod_t(1-\varepsilon_t)$.*

If the projected table is correct on all reachable rows and the minimum correct-versus-competitor logit gap on those rows is $\gamma > 0$, every elementwise score perturbation of magnitude strictly less than $\gamma/2$ preserves universal projected correctness.

Proof of proposition 22. Suppose every reachable kernel row is a point mass on its true successor. Induction immediately gives point-exactness after every prefix. Conversely, assume universal point-exactness. For any reachable state x, choose a word u reaching it. Exactness after u gives input distribution δ_x to the next kernel. Exactness after ua for every primitive a then requires $P_a(x,\cdot) = \delta_{\tau_a(x)}$.

For the quantitative bound, the probability of taking the true transition at every step is at least $\prod_t(1-\varepsilon_t)$. Every such path ends at the true state, so the final mass there is at least this product. For a target point mass δ_{x^*}, $\mathrm{TV}(p, \delta_{x^*}) = 1 - p(x^*)$, proving the bound.

Finally, suppose the correct logit exceeds every competitor by at least γ. Under an elementwise perturbation of magnitude less than $\gamma/2$, the correct-minus-competitor gap decreases by less than γ and remains positive. Every projected row is unchanged; theorem 11 preserves the certificate. □

B PatchWorld, models, and protocols

B.1 PatchWorld semantics and support

The primitive update maps are defined in section 5; the local state and token argument both lie in $\mathbb{Z}_5$.

In overwrite, every local chain contains two distinct reset tokens. There are $5 \cdot 4 = 20$ ordered local types. In standard algebra, every chain contains an initial Set, one Add, and one Mul, with the arithmetic tokens in either order. There are $2 \cdot 5^3 = 250$ complete local token traces. All are represented among 4,096 training programs, while no six-key test tuple repeats. A fixed six-key algebra tuple has $18!/(3!)^6 = 137{,}225{,}088{,}000$ legal schedules.

Table 4: Support geometry of the main split. Local coverage is coverage of complete local token traces; global overlap is repeated test key-tuples.

Task	Local types	Local coverage	Global overlap	Schedules/tuple
Overwrite	20	1.000	2.44×10^{-5}	7,484,400
Algebra	250	1.000	0	137,225,088,000

Legal schedules for equal-length chains are sampled uniformly by repeatedly choosing the next key with probability proportional to its remaining events. Fixed evaluation transformations include canonical key blocks, round-robin interleaving, reverse key blocks, a valid random schedule, and arithmetic-order changes. All-32 additionally samples 32 fresh legal linearizations. Same-key counterfactual evaluation retains examples whose oracle answer changes. Dep-ID reverses the supported arithmetic order. SET-OOD moves reset away from its training position. The held-out-composition regime excludes every adjacent ADD→MUL bigram from training while retaining each primitive and all relevant semantic rows.

B.2 Architectures and supervision

Flat and augmented GRUs. The flat model consumes the observed linearization. The wider control trains on valid random interleavings but does not enforce invariance.

Deep Sets and Set Transformer. These models are invariant to all event permutations, including dependent same-key swaps, and directly test proposition 2.

Stable-sort global GRU. The input is stably sorted by key before one global recurrent encoder. By proposition 1, this is an exact quotient representation, although every key must still be represented through one global hidden state.

TSF. Each $\pi_k(h)$ is encoded independently. Shared TSF reuses one GRU; unshared TSF has key-specific parameters. Final-only TSF receives only final-state cross-entropy. The matched-information control adds cross-entropy after every local event using the same oracle semantic states supplied to TOF. The auxiliary typed-reset condition additionally replaces the addressed local GRU state after SET_v by one of five learned value-specific reset vectors; at hidden width H, this adds $5H$ trainable parameters. Its outcomes are reported alongside ordinary intermediate supervision in tables 10 to 12.

TOF. For every primitive a, TOF has a 5×5 score matrix. The soft rollout uses row-wise $P_a = \mathrm{softmax}(L_a)$ for free learned rows and is trained with final and intermediate semantic-state cross-entropy. The stated reset prior instead requires $P_{\mathrm{SET}_v}[x, :] = \delta_v$ and $\widehat{\tau}_{\mathrm{SET}_v}(x) = v$, including 20 noninitial reset rows absent because reset appears first in standard training. Exact point-mass resets cannot be represented by finite softmax logits; the implementation imposes them directly for both soft and projected execution. Projected inference applies eq. (7) to the learned rows and retains the prescribed reset maps. Soft and projected rows in the matched suite use the same trained scores. The full table has 375 score entries; some reset entries are constrained by the declared prior.

B.3 Optimization, selection, and reproducibility

The primary state-only study uses AdamW (Loshchilov & Hutter, 2019), weight decay 10^{-4}, gradient clipping at 5, batch size 256, and at most 30 epochs. The final matched suite uses validation-only selection from learning rates $\{3 \times 10^{-4}, 10^{-3}, 3 \times 10^{-3}\}$, 6,000 optimizer steps, and validation every 250 steps. Calibration and evaluation seeds are disjoint, and evaluation datasets are instantiated after checkpoint selection.

The matched study uses a single NVIDIA A100 and records seven evaluation seeds 1101–1107, trainable parameters, activation-state elements, elapsed time, and peak accelerator memory. Its per-evaluation records contain 16,384 examples for matched-information and data-efficiency conditions, and 24,576 for scaling. Condition intervals are bootstrap intervals over seeds. For paired comparisons in the main text, we exhaustively enumerate all 7^7 paired bootstrap resamples and all 2^7 sign flips

under the sign-exchangeability null; with seven nonzero same-sign differences, the smallest exact two-sided p-value is $2/2^7 = 0.015625$.

C PRIMARY AND TRANSITION-STUDY RESULTS

C.1 PRIMARY STATE-ONLY STUDY

Table 5: Primary state-only results, means over ten held-out seeds. Worst is the minimum across legal schedules; All-32 requires correctness under every sampled legal linearization. † denotes intermediate-state supervision and the typed reset prior.

(a) Overwrite Model	Canon.	Worst	All-32	Dep-ID	SET-OOD
Flat GRU	.5533	.0002	.0000	.5529	–
Wide GRU + augmentation	.9999	.9997	.9980	.9997	–
Deep Sets	.0157	.0157	.0157	.0156	–
Set Transformer	.0150	.0150	.0150	.0153	–
Stable-sort GRU	.9998	.9998	.9998	.9998	–
TSF shared	1.0000	1.0000	1.0000	1.0000	–
TSF unshared	1.0000	1.0000	1.0000	1.0000	–
TOF projected†	1.0000	1.0000	1.0000	1.0000	–
(b) Algebra Model	Canon.	Worst	All-32	Dep-ID	SET-OOD
Flat GRU	.5990	.0000	.0000	.5998	.0402
Wide GRU + augmentation	.0590	.0519	.0006	.0596	.0110
Deep Sets	.0002	.0002	.0002	.0002	.0001
Set Transformer	.0029	.0029	.0029	.0028	.0007
Stable-sort GRU	.9947	.9947	.9947	.9947	.0542
TSF shared	1.0000	1.0000	1.0000	1.0000	.0930
TSF unshared	1.0000	1.0000	1.0000	1.0000	.0938
TOF projected†	1.0000	1.0000	1.0000	1.0000	1.0000

All exact stable-sort, TSF, and TOF completion runs have zero measured prediction changes across legal schedules. In the broader declared-invariance audit, all 357 runs preserve predicted labels; five Set Transformer runs have maximum logit drift 1.3×10^{-5}, narrowly above the predeclared 10^{-5} numerical tolerance.

C.2 VERTICAL GENERALIZATION AND OPERATOR FACTORS

Table 1 in the main text reports the vertical evaluations and the complete TOF factor ablation.

In the transition study, TOF soft has all row argmaxes correct and is perfect at training length three, with 0.0197 accuracy at length four and essentially zero by length 12. Projected execution retains 1.0 through length 16, directly demonstrating the benefit of composing the recovered discrete transitions.

C.3 DATA EFFICIENCY AND FIXED-BUDGET KEY SCALING

Table 6: State-only algebra data efficiency: worst-valid JGA, mean over five seeds.

Model	512	1,024	2,048	4,096	8,192
Stable-sort GRU	.0003	.0005	.0634	.9954	.9993
Wide GRU + augmentation	.0000	.0000	.0000	.0660	.7842
TSF shared	.0001	.0017	.1398	1.0000	1.0000
TSF unshared	.0003	.0090	.5196	1.0000	1.0000

Table 7: Fixed-budget key scaling at 8,192 programs: worst-valid JGA, mean over three seeds. The larger-sample study in section D complements these fixed-budget measurements.

Model	$K = 2$	$K = 4$	$K = 8$	$K = 12$	$K = 16$
Stable-sort GRU	1.0000	.9997	.9995	.4541	.0007
Wide GRU + augmentation	1.0000	.9992	.1203	.0000	.0000
TSF shared	1.0000	1.0000	1.0000	1.0000	1.0000
TSF unshared	1.0000	1.0000	1.0000	1.0000	1.0000
TOF projected	1.0000	1.0000	1.0000	1.0000	1.0000

A deterministic lookup diagnostic isolates support pooling without neural optimization. Across 200 train/test seed pairs, shared complete-trace lookup attains 0.7856 JGA with 128 global programs, 0.9870 with 256, and 1.0 with 512; key-specific lookup attains 0.0170, 0.1187, and 0.5000. This is consistent with proposition 17. The fact that unshared TSF optimizes faster than shared TSF at intermediate sample sizes shows that support complexity and neural optimization are distinct.

C.4 Other controls

Removing an arbitrary synthetic action auxiliary raises stable-sort algebra JGA from 0.8792 to 0.9956 and the wider augmented GRU from 0.0184 to 0.0456 on three independent seeds; TSF and projected TOF remain 1.0. The state-only objective is therefore primary, and joint-objective results serve as a sensitivity analysis. Corrupting one observed key pair reduces algebra JGA from 1.0 to 0.1032 for TSF and 0.0704 for projected TOF, demonstrating that exactness is conditional on routing.

D Matched experiments

D.1 Data efficiency and higher-sample scaling

Table 8: Data efficiency at $K = 6$, hidden size 96, state-only objective (seven seeds). Parentheses give 95% bootstrap intervals.

Training programs	Stable-sort global GRU mean (95% CI)	TSF final only
2,048	.399170 (.284546–.516418)	1.000000
8,192	.999747 (.999608–.999895)	1.000000
32,768	.999756 (.999695–.999817)	1.000000

At 2,048 programs, the seven paired TSF-minus-global JGA differences are 0.5533, 0.7939, 0.7047, 0.7948, 0.5117, 0.3394, and 0.5081. Their mean is 0.600830 (95% exhaustive paired-bootstrap interval $[0.483582, 0.714948]$; exact two-sided sign-flip $p = 0.015625$).

Table 9: Convergence-oriented scaling at 98,304 programs (seven seeds). Parameters and recurrent state elements are reported for $K = 16$.

Model	H	$K = 12$	$K = 16$	Params ($K = 16$)	State elems. ($K = 16$)
Stable-sort global	96	.999849	.999826	55,392	96
Stable-sort global	192	.999866	.999855	165,024	192
Stable-sort global	384	.999843	.999895	550,176	384
Stable-sort global	768	.999965	.999843	1,984,032	768
Stable-sort global	1024	–	.999890	3,431,456	1,024
TSF shared	96	1.000000	1.000000	43,253	1,536
TSF unshared	96	1.000000	1.000000	692,048	1,536

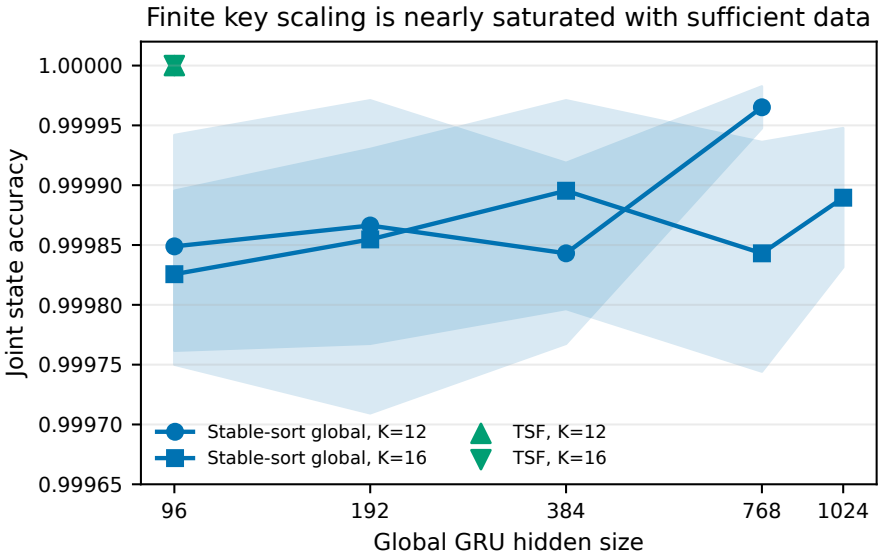

Figure 3: Higher-sample key scaling. With 98,304 programs, exact-canonical global GRUs are within roughly 10^{-4} JGA of perfect at $K = 16$ for every selected width. Product models remain perfect while allocating one local state per key.

The activation-state column counts recurrent state elements rather than all training activations. At $K = 16$, shared TSF uses fewer parameters but 16 times as many recurrent state elements as the $H = 96$ global model. Wider global conditions approach the product activation-state budget while using more parameters. These resource counts distinguish parameter capacity from recurrent-state storage.

D.2 Matched intermediate-information tests

Table 10: Standard regime after training on local length three (seven seeds). The auxiliary reset-information condition is shown alongside ordinary intermediate supervision.

Model	Main	SET-OOD	$L4$	$L8$	$L16$
TSF final only	1.0000	.0002	.0000	.0000	.0001
TSF + intermediate states	1.0000	1.0000	.9975	.5086	.0134
Aux. TSF + reset information	1.0000	1.0000	.7498	.0035	.0001
TOF soft rollout	1.0000	1.0000	.0046	.0001	.0001
TOF projected, same scores	1.0000	1.0000	1.0000	1.0000	1.0000

Table 11: Held-out ADD→MUL regime (seven seeds).

Model	Main	SET-OOD	Composition	$L8$	$L16$
TSF final only	.5215	.0003	.0433	.0001	.0001
TSF + intermediate states	.9763	1.0000	.9118	.1274	.0008
Aux. TSF + reset information	.8345	1.0000	.4723	.0004	.0001
TOF soft rollout	1.0000	1.0000	1.0000	.0001	.0001
TOF projected, same scores	1.0000	1.0000	1.0000	1.0000	1.0000

Plain intermediate-supervised TSF is the strongest recurrent comparison in this study. The auxiliary reset-information control and all its evaluation conditions are retained in the table and specified in the supplementary code.

Table 3 in the main text gives all four mechanism-specific paired comparisons.

The composition comparison has four seeds tied at 1.0 and an exact paired $p = 0.25$. The length-16 comparisons give larger, more consistent differences across seeds. All listed tests are unadjusted, mechanism-specific comparisons.

D.3 History-conditioned transition consistency

The diagnostic groups histories by oracle pre-event state x and arithmetic primitive a. Next-state accuracy is the fraction of correct predicted successors. Agreement is the modal predicted-successor frequency within each group. Mean Jensen–Shannon (JS) divergence compares each prediction distribution with its group centroid; worst-group accuracy summarizes the least accurate group. The supplementary documentation specifies the corresponding data fields and aggregation rules.

A TOF kernel row $P_a[x, :]$ is history-independent by construction. The diagnostics instead summarize history-conditioned predictions, which can depend on the incoming soft-state distribution. They describe prediction consistency rather than certifying decoded-state closure; the latter requires the structural condition in eq. (5) and a check of the deployed transition table.

Table 12: History-conditioned transition consistency, means over seven seeds. Accuracy and Jensen–Shannon divergence summarize predicted successors associated with histories, rather than the intrinsic fixed TOF row.

Regime	Model	Next acc.	Agreement	Mean JS	Worst acc.
Standard	TOF soft	.7111	.7111	.0037	.5234
	TSF final	.3460	.4021	.5821	.2121
	TSF + intermediate	.8103	.8103	.3104	.7277
	Aux. TSF + reset info.	.7949	.7949	.2926	.6540
Held-out comp.	TOF soft	.7082	.7082	.0034	.5234
	TSF final	.3017	.3848	.6167	.1551
	TSF + intermediate	.7858	.7858	.3176	.6529
	Aux. TSF + reset info.	.7440	.7440	.3083	.5223