

Portable Learned Task Automata: Exact Spline Compilation and a Thirty-Pool Capability Failure

Daniel Schmitter

Abstract

Compact learned task descriptions can be exchanged without transmitting a donor policy, but interchange does not establish collective capability. We evaluate coordinate-dependent temporal automata represented by stochastic cardinal transition fields. Exact cellwise Bernstein conversion preserves their functions and accelerates a support-state planner. A frozen thirty-pool confirmation nevertheless yields only 161 of 480 composed missions for cardinal support, compared with 242 for a matched-query MLP and 464 for a supplied-model reference. Classical sequential reuse matches the corresponding learned product-planner counts. Fourteen inspection acquisitions collapse because the discovery alphabet lacks informative coverage. Portability and immutable-module checks pass; capability and beyond-classical composition claims fail. The contribution is a scoped compiler and a complete acquisition-to-decision failure analysis, not a robotic swarm or recursive self-improvement system.

1 Introduction

The motivating vision is a collection of small agents that acquire and exchange useful knowledge without repeatedly retraining a central network. This experiment tests a narrower prerequisite: separately learned temporal task descriptions supporting navigation under a supplied motion model. Acquisition may use distillation; that does not make the supplied task model a learned motor policy.

Learners receive binary sequence-acceptance answers from cheap synthetic programs. Coordinates, a motion graph, and an action-synthesizing planner are supplied. We report the full acquisition-to-decision test, not merely message size or standalone classifier accuracy.

2 Related work

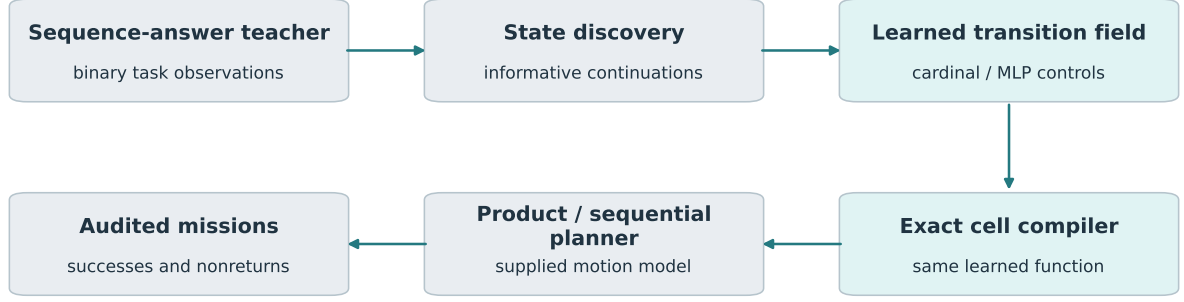
Reward machines expose temporal task structure [1], and learning such machines from experience has substantial prior art [2]. Product-state planning and sequential composition are classical controls. Cardinal inner products and polynomial cell extraction are representation tools. The question is whether this particular learned interchange remains useful after acquisition errors and stronger alternative representations are included.

3 Architecture and information flow

The pipeline exchanges learned task semantics rather than motor policies. A teacher answers whether sequences satisfy a temporal requirement. Discovery builds candidate states, transition examples fit coordinate-dependent matrices, and a planner combines these modules with a supplied motion graph. The compiler sits after acquisition: it changes evaluation cost but cannot add temporal distinctions absent from the learned state space.

This ordering makes the inspection-state collapse diagnostically important. If two prefixes are never distinguished by the chosen continuation alphabet, no amount of faithful serialization recovers their different future requirements. Query count is not information coverage. Even after accounting for collapsed discoveries, the cardinal system retains a capability deficit against the matched-query MLP, so one diagnosed bottleneck is not a complete explanation of the result.

Acquisition, compilation, composition



Read left to right above, then right to left below. Arrows show data or computation flow.

Figure 1: Acquisition, compilation, composition. Boxes distinguish supplied information, fitted components, and the quantity evaluated. Arrows show computation or data dependence, not a newly trained deep network.

4 Coordinate-dependent temporal state

Prefixes are distinguished by responses to a finite continuation set; evaluator state labels and template identifiers are withheld. Given inferred states, transition examples fit a two-dimensional cardinal field:

$$A(x, y) = \sum_{i,j} B_i(x) B_j(y) C_{ij}, \quad (1)$$

$$(C_{ij})_{rs} \geq 0, \quad \sum_s (C_{ij})_{rs} = 1. \quad (2)$$

Nonnegative partition of unity makes every transition matrix row-stochastic in exact arithmetic on its declared domain. A soft acceptance score multiplies these matrices along a word. Maximum-probability state decisions are a further approximation, not an equivalent stochastic model.

The final planner retains possible temporal states using a fixed support cutoff 0.02 and eight subdivisions per motion segment. Finite-alphabet minimization reduces the set-state machines. The cutoff is not calibrated uncertainty; eight samples do not prove continuous avoidance. Multiple memories combine through a classical product dynamic program, with all-order sequential reuse as a control.

5 Exact evaluation compilation

A cardinal cubic basis restricts to a cubic polynomial on each cell. A fixed extraction matrix converts its active coefficient block into Bernstein form. For each two-dimensional transition entry,

$$A_{rs}(x, y) = b_3(u)^T E_{rs} b_3(v), \quad (3)$$

$$E_{rs} = T C_{rs}^{\text{local}} T^T. \quad (4)$$

Proposition 1. Exact basis conversion preserves the transition function throughout the cell. Substitute the one-dimensional basis identity on each coordinate to obtain the formula. Floating-point evaluation and threshold decisions require separate checks. The identity preserves the learned function, not the correctness of inferred temporal states.

The compiler specializes support evaluation and retains only required program state. Continuous mass and derivative products use four-point Gaussian cell quadrature, exact for these polynomial products before rounding. Finite-domain Grams are banded rather than circulant. None of these operations removes the product-state growth that can occur with additional task requirements.

6 Experimental methods

Thirty independent donor pools provide visit, avoidance, and ordered-inspection models. Eight scenes per requirement count give 720 primary missions, including 480 with three or four requirements. A six-pool heading-lattice panel adds 36 missions. Seeds, grids, thresholds, and horizons remain frozen during confirmation.

Matched-query bounded decision trees, a 64-tree ensemble, and a 2-64-64-state-squared ReLU MLP use the same inferred states and transition examples. Controls include maximum-probability decisions, support sets, sequential reuse, additive cost maps, no transfer, individual donors, and supplied-state models. A common cutoff does not equal matched uncertainty calibration. Supplied-model planning is an implementation reference, not an optimal or safe ceiling.

Success requires a returned goal-reaching path, collision avoidance, and temporal acceptance under a 64-substep audit. Nonreturns are failures. Uncertainty uses 10,000 fixed bootstrap resamples of donor pools, not independent missions. A separate interval audit subsequently checks continuous center avoidance of every returned path without changing the primary score.

7 Results

Table 1: Three-/four-requirement successes out of 480; full-plan median uses all 720 primary missions.

Method	Successes	Median ms
Cardinal support	161	349.7
Cardinal maximum	147	878.2
MLP support	242	498.3
MLP maximum	162	536.7
Cardinal sequential	161	470.3
MLP sequential	242	615.7
Cardinal cost map	0	219.9
Supplied-model reference	464	428.6

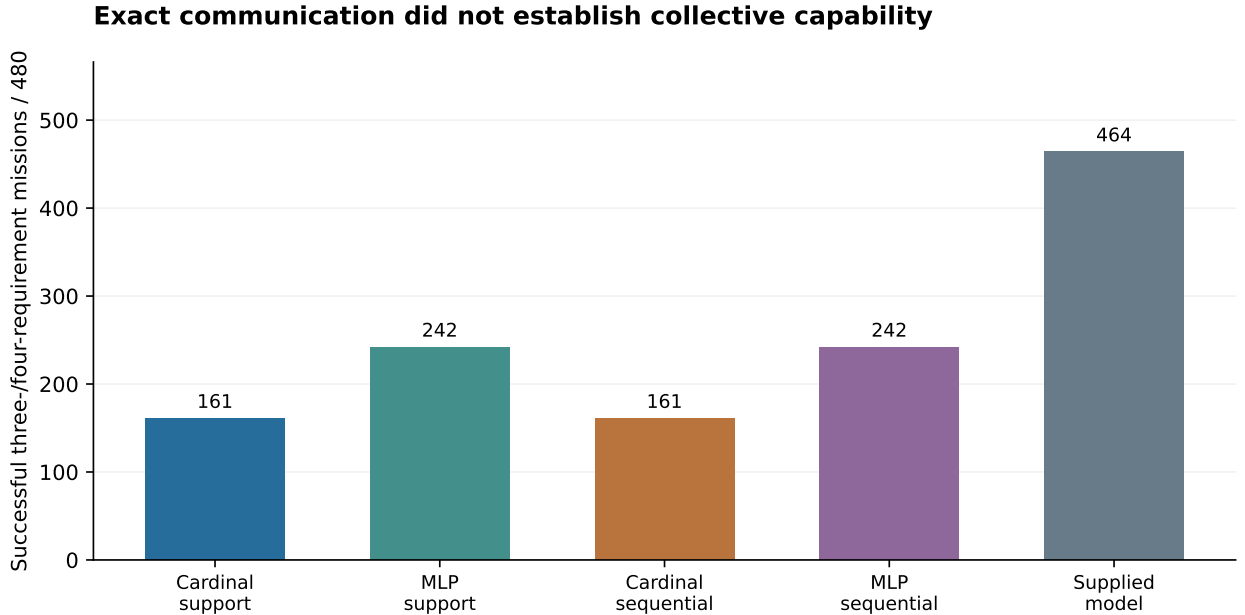


Figure 2: Complete three-/four-requirement success counts for the principal learned and supplied-model controls. Sequential reuse matches the corresponding learned product-planner counts.

Cardinal support achieves 33.5%, with descriptive donor-cluster interval 21.5-46.7%; MLP support achieves 50.4%, interval 34.0-67.3%. Their paired difference is minus 16.875 percentage points, interval minus 25.2 to

minus 9.4. Sequential reuse matches each learned success count. Capability and beyond-classical-composition claims fail. Secondary counts are 22/36 cardinal, 25/36 MLP, and 34/36 reference.

Fourteen inspection discoveries collapse to one nonaccepting state. A saved-query audit predicts all state counts from whether both inspection regions occur in the 64-anchor discovery alphabet. After collapse, single-point fitting queries with the empty continuation cannot reveal the two-step order. This accounts for 224 composed nonreturns shared with MLP, but not the entire remaining cardinal deficit. The diagnosis does not rescue the experiment.

All 401 returned cardinal primary paths pass the original score; its 319 nonreturns remain failures. A later interval audit checks 9,158 returned method-mission paths, including duplicates, finding 7,926 avoidance-clear paths, 1,232 violation witnesses, and no unresolved checks. It assumes exact stored geometry and the interval runtime and checks center avoidance only, not physical robot safety.

The compiler improvement survives: on reused development scenes, specialization preserves checked tables, paths, and memories while reducing full-plan time $2.10^{-2}.46$ times and program arrays from 999,424 to 142,325 bytes. Confirmation median exported cardinal messages total 17,096 bytes, excluding mission meta-data and planning memory. Acquisition uses 1,813,994 nonvalidation teacher queries plus 92,160 validation labels. All thirty disconnection, exact replay, and old-module immutability checks pass.

8 Discussion

A small payload encoding an incorrect model is not successful compression. Many queries need not cover informative sequences, and exact compilation cannot create states that were never acquired. These facts separate sound representation engineering from the unestablished collective capability.

Immutable separate modules provide structural retention, not learned zero forgetting or recursive learning-rule redesign. Synthetic task acceptance omits unknown physical dynamics. The pipeline is closed without a third semantic revision or threshold sweep. Future physical skill transfer needs an acquisition interface and task benefit that survive classical modular and shared-data controls.

9 Application boundary and research implication

Portable modules and exact compilation remain useful software properties. Demonstrated collective intelligence would additionally require acquisition that survives composition and a task benefit beyond ordinary sequential reuse. The thirty-pool confirmation explicitly fails that stronger claim and should remain closed.

10 Conclusion

Exact cardinal-to-Bernstein compilation yields a useful internal performance improvement. The independently confirmed task-description pipeline fails its capability objective. Reporting both outcomes identifies a concrete compiler result without promoting it into a demonstrated swarm-learning breakthrough.

References

- [1] R. Toro Icarte, T. Klassen, R. Valenzano, and S. McIlraith. Using Reward Machines for High-Level Task Specification and Decomposition in Reinforcement Learning. ICML, 2018.
<https://proceedings.mlr.press/v80/icarte18a.html>
- [2] R. Toro Icarte, E. Waldie, T. Q. Klassen, R. Valenzano, M. P. Castro, and S. A. McIlraith. Learning Reward Machines: A Study in Partially Observable Reinforcement Learning. Author preprint, 2021.
<https://arxiv.org/abs/2112.09477>