STATE DYNAMICS OF AGENT HARNESSES: WHAT STATES MATTER FOR PRESERVATION?

Anonymous authors
Paper under double-blind review

ABSTRACT

Agent harnesses shape long-horizon agent behavior by controlling base model and tool calls, feedback integration, and persistent workspace state. As a result, the same base model can exhibit different interaction dynamics and task outcomes across Harnesses. Yet existing Harness designs are still shaped largely by empirical rules, leaving the propagation of local changes over a finite interaction window, the conditions that a state representation must preserve for later operations, and the formation of a readable task object without a common dynamical account. In this work, we develop a conditional state dynamics framework that formalizes finite response, conditional attractor structure, operation-aware state fidelity, and task readout. Analyses across models, harnesses, and tasks provide empirical support for the relationships among spectral structure, interaction persistence, successor prediction, and code readout. Building on the task-object criterion, we propose **BasinGuard**, an online plugin that runs at Harness protocol events, continues the host protocol while the object is still forming, and invokes the specified readout once the object is determined. Across the evaluated model–Harness conditions, BasinGuard raises the normalized task score by 17.15% on average on code workspaces and improves official pass rates on GAIA2. An anonymous demo is available at anonymous.4open.science/r/BasinGuardPlugin.

1 INTRODUCTION

Agent harnesses have become central to long-horizon agents, broadening the focus from model scaling alone to the design of the surrounding system (Gu, 2026; Guo et al., 2026; Li et al., 2026b; Dong et al., 2026). In software repair, scientific programming, and command-line tasks, harnesses coordinate repeated cycles of tool use and environmental feedback as the task unfolds. Even with the same base model, differences in call scheduling, feedback integration, and retained workspace state can lead to different interaction trajectories and task outcomes (Jimenez et al., 2024; Tian et al., 2024; Merrill et al., 2026; Li et al., 2026c; Yang et al., 2024; Wang et al., 2025a).

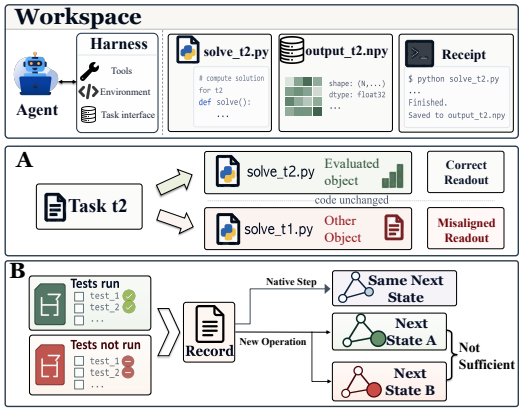

Figure 1: Harnesses can place the same program in different conditional states. Execution facts and the evaluated code determine the task object.

Most of the existing Harness designs remain largely empirical, making the downstream effects of local design choices difficult to predict. Recent systems introduce harness programming languages, long-horizon runtimes, and learnable control policies (Li et al., 2026d; Zheng et al., 2026; Pan et al., 2026; Fu et al., 2026; Jiang et al., 2026). A common dynamical account is needed for how a local change propagates through a finite interaction, which operation-dependent distinctions a state representation must preserve, and how a program becomes a task object that an evaluator can read. Cordis provides a formal basis for the spatiotemporal composition of dynamic components through reversible effects and reactive dependencies (Shi et al., 2026); Meta-Harness and AutoHarness treat historical code, environmen-

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.

tal feedback, and execution traces as material for continually rewriting a Harness (Lee et al., 2026; Lou et al., 2026). These advances make harnesses programmable and revisable, increasing the need to understand how revisions affect subsequent execution and task evaluation. A unified dynamical account is therefore needed to explain how local changes propagate over finite interaction windows, which operation-dependent distinctions a state representation must preserve, and when code and workspace facts jointly determine an evaluable task object.

We model interaction under a harness as a local stochastic process driven jointly by the model, tools, and environment. Specifically, Finite-response spectra and conditional distributions determine how a local change propagates and how access is weighted over time; operation-aware fidelity determines which distinctions later operations must retain; and the task readout map connects these conditions to an evaluable object. Interaction records from multiple models, Harnesses, and tasks exhibit the corresponding spectral, predictive, and readout relations, while adaptive selection tasks show how the same conditional state enters later choices. Building on the task-object criterion, BasinGuard turns task-object formation into an online readout boundary for code-workspace tasks with different interaction organizations. The paper makes three contributions that follow from this analysis:

- We derive a common relation among finite response, conditional attractors, and operation-aware state fidelity, connecting local state changes to task readout.
- We find the same conditional organization in interaction records across models, Harnesses, and tasks, linking spectral residence, successor prediction, code readout, and adaptive selection.
- We propose BasinGuard as an online plugin that turns task-object formation into a protocol-level readout boundary. On public code-workspace tasks, it raises the equal-task score by 17.15% for Qwen3.8-27B across three Harnesses and raises the official GAIA2 pass rate in every evaluated model–Harness condition.

2 Conditional State Dynamics

An agent Harness advances a long interaction through model calls, tool execution, workspace changes, and environmental responses. We model this interaction as a conditional dynamical system and combine formal derivations with empirical evidence. The analysis moves from the spectral attractor structure of a local transition kernel through operation-aware state fidelity and code-workspace readout to conditional control in adaptive decisions.

2.1 Spectral Attractor Dynamics of Local Interaction

Within a local interaction phase whose transition rule is approximately stable, the evolution can be represented by a locally absorbing Markov chain. Let τ denote a task, and let $c = (h, m, \mathcal{P})$ collect the Harness, model, and task-distribution conditions. The admissible operation set is $\mathcal{U}$. At event step t, the public history H_t contains the task objective, code state, tool records, and environmental feedback, and induces the conditional state $Z_t = \Phi_c(H_t)$. Initially, $\bar{Z}_0 \in C$. If, for every operation $u \in \mathcal{U}$, the joint distribution of the next state and its exit category conditioned on Z_t is independent of the remainder of H_t, then Z_t is a control-sufficient representation for $\mathcal{U}$: it retains the environmental facts that subsequent operations actually read (Singh et al., 2012; Chen, 2026; Subaharan, 2026; Wu et al., 2026).

Let the active state set be $C = \{1, \dots, n\}$ and the exit-category set be $\mathcal{B} = \{1, \dots, k\}$. The extended state $\bar{Z}_t$ takes values in C before the first exit and remains in the corresponding exit category afterward. With the event convention that a transition at step t maps time t to time $t + 1$, define $\sigma = \inf\{t \geq 1 : \bar{Z}_t \in \mathcal{B}\}$. For operation $u \in \mathcal{U}$, let the within-region transition kernel and the first-exit kernel be $\mathbf{Q}_u \in [0,1]^{n \times n}$ and $\mathbf{R}_u \in [0,1]^{n \times k}$, respectively. They satisfy the row-mass identity $\mathbf{Q}_u \mathbf{1}_n + \mathbf{R}_u \mathbf{1}_k = \mathbf{1}_n$, where $\mathbf{1}_d$ is the d-dimensional all-ones column vector. Write the reference rule as $\mathbf{Q} = \mathbf{Q}_0$ and $\mathbf{R} = \mathbf{R}_0$, and let the initial distribution be the probability row vector $\boldsymbol{\mu} \in [0,1]^n$ with $\boldsymbol{\mu}\mathbf{1}_n = 1$. For an event window of length $T \in \mathbb{N}$, and for times with survival mass $S_t > 0$, define

$$S_t = \boldsymbol{\mu}\mathbf{Q}^t\mathbf{1}_n, \qquad \boldsymbol{\nu}_t = \frac{\boldsymbol{\mu}\mathbf{Q}^t}{S_t}, \qquad \mathbf{K}_T = \sum_{t=0}^{T-1} \mathbf{Q}^t\mathbf{R}. \tag{1}$$

Here $S_t = \mathbb{P}(\sigma > t)$ is the mass that remains in C at time t, $\boldsymbol{\nu}_t$ is the corresponding conditional state distribution, and $\mathbf{K}_T$ is the subdistribution of first exits by category within the window. Consequently, $\mathbf{K}_T\mathbf{1}_k = \mathbf{1}_n - \mathbf{Q}^T\mathbf{1}_n$, and the mass that has not exited by the end of the window is $S_T = 1 - \boldsymbol{\mu}\mathbf{K}_T\mathbf{1}_k$. To study how a rule change propagates to the exit distribution, let the modified rule be $(\mathbf{Q}', \mathbf{R}')$ and write the local kernel differences as $\Delta\mathbf{Q} = \mathbf{Q}' - \mathbf{Q}$ and $\Delta\mathbf{R} = \mathbf{R}' - \mathbf{R}$. Define $\mathbf{K}'_T = \sum_{t=0}^{T-1}(\mathbf{Q}')^t\mathbf{R}'$ analogously, with $\mathbf{K}_0 = \mathbf{K}'_0 = \mathbf{0}$.

Theorem 1 (Finite response to a local modification)**.** *For any valid transition pair and $T \geq 1$, the response of the cumulative exit subdistribution is*

$$\boldsymbol{\mu}(\mathbf{K}'_T - \mathbf{K}_T) = \sum_{t=0}^{T-1} \boldsymbol{\mu}(\mathbf{Q}')^t\big[\Delta\mathbf{R} + \Delta\mathbf{Q}\mathbf{K}_{T-1-t}\big]. \tag{2}$$

Equation (2) separates the response of a rule change into two channels. $\Delta\mathbf{R}$ changes the mass sent directly to an exit category, while $\Delta\mathbf{Q}\mathbf{K}_{T-1-t}$ changes the within-region state and lets the difference propagate to the exit boundary over the remaining $T-1-t$ steps. When the modification is localized to a single event, the expansion reduces exactly to the mass reaching that event, weighted by the subsequent propagation. The construction is illustrated in Figure 2. Let the simplex of probability row vectors on C be $\Delta_C = \{\boldsymbol{\nu} \in [0,1]^n : \boldsymbol{\nu}\mathbf{1}_n = 1\}$. On the survival region $\boldsymbol{\nu}\mathbf{Q}\mathbf{1}_n > 0$, define the conditional update map $\mathcal{F}_\mathbf{Q}(\boldsymbol{\nu}) = \frac{\boldsymbol{\nu}\mathbf{Q}}{\boldsymbol{\nu}\mathbf{Q}\mathbf{1}_n}$. If $\mathbf{Q}$ is a primitive transient kernel, meaning that it is nonnegative, has row sums at most one, has a strictly positive integer power, and has positive exit mass ($\mathbf{Q}\mathbf{1}_n \neq \mathbf{1}_n$), then its spectral radius satisfies $0 < \rho < 1$. Quasi-stationary distribution theory gives a unique positive probability row vector $\boldsymbol{\alpha}$ such that $\boldsymbol{\alpha}\mathbf{Q} = \rho\boldsymbol{\alpha}$ and $\boldsymbol{\alpha}\mathbf{1}_n = 1$, and $\mathcal{F}^t_\mathbf{Q}(\boldsymbol{\mu}) \to \boldsymbol{\alpha}$ for every initial distribution $\boldsymbol{\mu}$ (Darroch & Seneta, 1965; Champagnat & Villemonais, 2016). The distribution $\boldsymbol{\alpha}$ gives the long-run access weights conditional on survival, and the mean residence scale is $\tau_{\text{res}} = (1-\rho)^{-1}$. Let $(\rho', \boldsymbol{\alpha}', \mathbf{v}')$ be the leading spectral triple of $\mathbf{Q}'$, with $\boldsymbol{\alpha}'\mathbf{Q}' = \rho'\boldsymbol{\alpha}'$ and $\mathbf{Q}'\mathbf{v}' = \rho'\mathbf{v}'$, normalized by $\boldsymbol{\alpha}'\mathbf{1}_n = 1$ and $\boldsymbol{\alpha}'\mathbf{v}' = 1$. Substituting this triple into Equation (2) decomposes the finite response into the leading slow mode and a higher-order spectral remainder $\mathbf{e}_T$:

$$\boldsymbol{\mu}(\mathbf{K}'_T - \mathbf{K}_T) = (\boldsymbol{\mu}\mathbf{v}')\sum_{t=0}^{T-1}(\rho')^t\boldsymbol{\alpha}'\big[\Delta\mathbf{R} + \Delta\mathbf{Q}\mathbf{K}_{T-1-t}\big] + \mathbf{e}_T, \tag{3}$$

where the norm bound for $\mathbf{e}_T \in \mathbb{R}^{1\times k}$ is given in Appendix B.4. Equation (3) identifies the role of the leading eigenvalue in temporal decay and the role of $\boldsymbol{\alpha}'$ in weighting how the perturbation spreads through the state space.

To evaluate this spectral structure empirically, we analyze interaction logs from MiniSWE, OpenHands, and SWE-Agent in software engineering, SciCode, and Terminus in terminal-based task execution. The logs cover open-weight models from several families and scales, including Gemma4-31B, Qwen3.5-122B, Qwen3.6-27B, and Qwen3.8-27B (Gemma Team, 2026; Qwen Team, 2026a;b;c; Yang et al., 2024; Wang et al., 2025a; Tian et al., 2024; Li et al., 2026c). The labels in Figure 3 use Gem4, Q3.5, Q3.6, and Q3.8 for the models and MS, OH, SWE, Sci, and Trm for the five Harnesses listed above. Each model call or tool return defines an event step t and its next event Y_t. We label protocol states as ready R, waiting W, interacting I, and returning T; the live region is $C = \{\text{R}, \text{W}\}$ and the exit categories are $\mathcal{B} = \{\text{I}, \text{T}\}$. Observations ending before an exit are right-censored, and these labels provide the protocol record Z_t used below. All kernels are estimated within a fixed model–Harness–operation condition, so the spectral and survival quantities describe a local interaction phase rather than a pooled global transition rule.

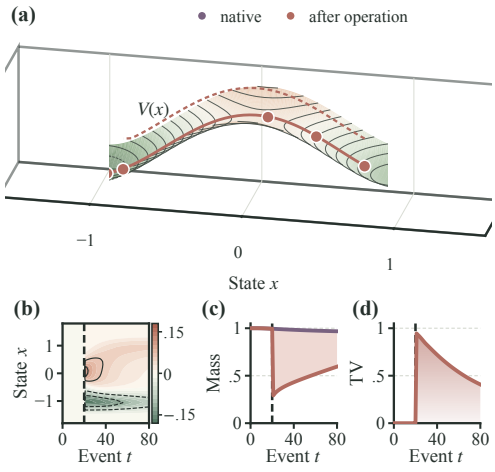

Figure 2: Finite response to an operation change. The modification at event t propagates through the remaining window via the cumulative exit kernel $\mathbf{K}_{T-1-t}$.

From the event sequences, we estimate the transition kernel by maximum likelihood and compute its empirical spectral quantity $\widehat{\rho}$ and residence scale $\widehat{\tau_{\text{res}}} = (1-\widehat{\rho})^{-1}$. For the observed process, the log-survival slope predicted by the substochastic kernel is $-\log\widehat{\rho}$. We fit the decay slope of interaction length with Kaplan–Meier survival analysis. Figures 3(a) and (b) show a layered distribution of empirical residence scales across Harness scheduling structures, with models of different sizes in the same environment occupying nearby spectral regions. This indicates a stable correspondence between local residence time and Harness scheduling topology. Figure 3(c) gives a Pearson correlation of $r_{\text{corr}} = .88$ between the kernel decay rate $-\log\widehat{\rho}$ and the Kaplan–Meier tail-decay rate on the log scale. Figure 3(d) shows agreement between the empirical survival curve and $\widehat{\rho}^t$; the Pearson correlation of residence scales across task partitions reaches $r_{\text{corr}} = .95$. Here r_{corr} denotes the Pearson correlation coefficient. Together, these results connect spectral quantities of the conditional attractor model to observable residence and tail decay in real interactions, supporting the empirical interpretation of Equation (3).

2.2 State Fidelity under Operation Constraints

Long-horizon interaction requires an agent to compress its history into a manageable state representation. The central criterion for a representation is whether later operations can still distinguish the state differences that they read. Let the surjection $\varphi : C \to \widetilde{C} = \{1, \dots, r\}$ partition fine states into coarse states, and let $\mathbf{L} \in \{0,1\}^{n\times r}$ be the grouping matrix with $L_{iz} = \mathbf{1}\{\varphi(i) = z\}$, where $\mathbf{1}\{\cdot\}$ is the indicator function. The within-region and exit kernels of the coarse process are $\widetilde{\mathbf{Q}}_u \in [0,1]^{r\times r}$ and $\widetilde{\mathbf{R}}_u \in [0,1]^{r\times k}$. The fine and coarse processes share the same operation support, and the decision policy depends only on the coarse-state history.

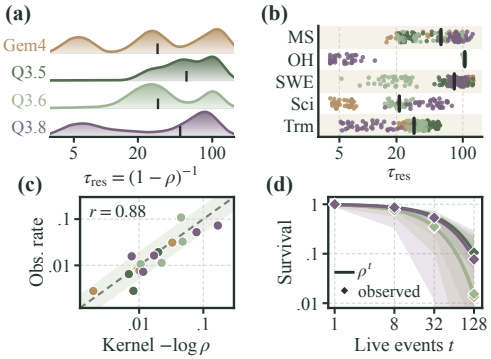

Figure 3: Conditional-attractor evidence. (a) Residence scale by Harness; (b) spectral radius by model; (c) kernel vs survival decay ($r = .88$); (d) survival curves ($r = .95$).

Theorem 2 (State fidelity under subsequent operations)**.** *Suppose that, for every operation $u \in \mathcal{U}$, the algebraic intertwining relations*

$$\mathbf{Q}_u\mathbf{L} = \mathbf{L}\widetilde{\mathbf{Q}}_u, \qquad \mathbf{R}_u = \mathbf{L}\widetilde{\mathbf{R}}_u \tag{4}$$

hold. Starting from $\boldsymbol{\mu}$ and $\widetilde{\boldsymbol{\mu}} = \boldsymbol{\mu}\mathbf{L}$, the fine and coarse processes then have exactly the same joint law over coarse states and operations under any policy based on the coarse history. Under a fixed homogeneous policy, if both transient kernels are primitive, they have the same spectral residence scale and their conditional attractor distributions satisfy $\widetilde{\boldsymbol{\alpha}} = \boldsymbol{\alpha}\mathbf{L}$.

Equation (4) is the strong lumpability condition for the operation set $\mathcal{U}$ (Givan et al., 2003; Li et al., 2006; Geiger & Temmel, 2014). It makes the transition operators commute exactly with the state-grouping operator; the proof is given in Appendix B.5. In the finite control construction below, V_T^{fine} and V_T^{φ} denote the probabilities of reaching a specified absorbing category within a window of length T under the optimal fine-state policy and the policy restricted to the coarse representation φ, respectively, and $\Delta V := V_T^{\text{fine}} - V_T^{\varphi}$. Figure 4 compares two representations with the same storage capacity. Here $\lambda \in [0,1]$ is the coupling between the retained state coordinate and the later operation. Retaining a variable used by that operation keeps ΔV at zero; retaining an unrelated variable lets the error grow with T and λ.

Let the next protocol event $Y \in \{1, \dots, d\}$ be the observable successor used to assess state fidelity. The simplex of probability row vectors is $\Delta_d = \{\mathbf{p} \in [0,1]^d : \mathbf{p}\mathbf{1}_d = 1\}$. Let Z^{full} be the full record and $Z^{\text{coarse}} = \psi(Z^{\text{full}})$ the coarse record, with conditional distributions

$$\mathbf{p}_{\text{full}}(z) = \mathbb{P}(Y = \cdot \mid Z^{\text{full}} = z), \qquad \mathbf{p}_{\text{coarse}}(z') = \mathbb{P}(Y = \cdot \mid Z^{\text{coarse}} = z').$$

For a predictor $\mathbf{q}$, define the Brier loss by $\mathcal{L}_Z(\mathbf{q}) = \mathbb{E}\|\mathbf{e}_Y - \mathbf{q}(Z)\|_2^2$, where $\mathbf{e}_Y \in \mathbb{R}^d$ is the one-hot vector for Y, $\|\cdot\|_2$ is the Euclidean norm, and $\mathbb{E}$ denotes expectation. The orthogonal decomposition of conditional expectations gives the exact identity

$$G_* := \mathcal{L}_{Z^{\text{coarse}}}(\mathbf{p}_{\text{coarse}}) - \mathcal{L}_{Z^{\text{full}}}(\mathbf{p}_{\text{full}}) = \mathbb{E}\|\mathbf{p}_{\text{full}}(Z^{\text{full}}) - \mathbf{p}_{\text{coarse}}(Z^{\text{coarse}})\|_2^2 =: I_*. \tag{5}$$

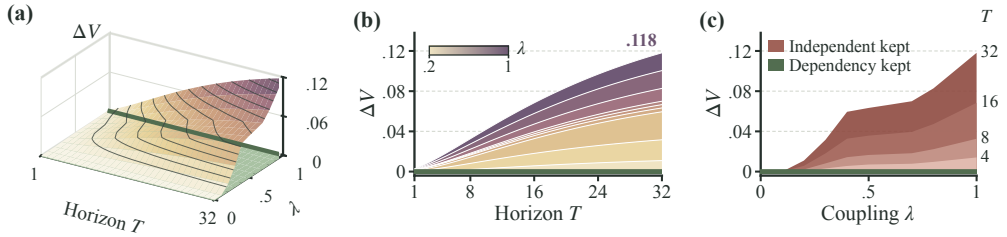

Figure 4: Equal-capacity representations. (a) ΔV over window T and coupling λ; green retains the dependency variable. (b,c) ΔV along T and λ.

We call I_* the squared predictive cost induced by merging states, and G_* the optimal Brier improvement obtained from the full record. The proof is given in Appendix B.8. If the pre-operation state follows the conditional attractor distribution $\boldsymbol{\alpha}$, the operation-weighted squared predictive cost for operation u is

$$I_{\boldsymbol{\alpha},u}(\varphi) = \sum_{i \in C} \alpha_i \left\| \mathbf{p}_u(i) - \overline{\mathbf{p}}_u(\varphi(i)) \right\|_2^2, \tag{6}$$

where $\mathbf{p}_u(i) = [\mathbf{Q}_u\mathbf{L} \quad \mathbf{R}_u]_{i,:}$ is the joint successor distribution after operation u from fine state i, represented as a row vector of length $r + k$. The brackets denote column concatenation and the subscript $i,:$ selects row i. For every coarse group z with $\sum_{i:\varphi(i)=z} \alpha_i > 0$, its $\boldsymbol{\alpha}$-weighted successor mean is

$$\overline{\mathbf{p}}_u(z) = \frac{\sum_{i:\varphi(i)=z} \alpha_i \mathbf{p}_u(i)}{\sum_{i:\varphi(i)=z} \alpha_i}.$$

On the support of $\boldsymbol{\alpha}$, this cost is zero if and only if fine states within every coarse group have identical successors. If the condition holds for every operation, the exact recovery statement in Theorem 2 follows from Equation (4). For global empirical statistics, let the protocol state be the random variable $\mathsf{S} \in \mathcal{S}$ with marginal weight $\omega_s = \mathbb{P}(\mathsf{S} = s)$. Define the conditional squared difference

$$a_s = \mathbb{E}\left[\left\| \mathbf{p}_{\text{full}}(Z^{\text{full}}) - \mathbf{p}_{\text{coarse}}(Z^{\text{coarse}}) \right\|_2^2 \middle| \mathsf{S} = s \right].$$

Then the total cost has the weighted orthogonal decomposition $I_* = \sum_s \omega_s a_s$. In the empirical analysis, we replace the population conditional distributions with task-held-out predictors $\widehat{\mathbf{p}}_{\text{coarse}}$ and $\widehat{\mathbf{p}}_{\text{full}}$ and compute $\widehat{I}$ and $\widehat{G}$ accordingly. These are empirical proxies for Equation (5); additional estimation details are given in Appendix D.2.

To test Theorem 2, we compare the base record $Z_t^{\text{coarse}} = (\kappa_t, a_t)$, which contains the call phase κ_t and within-call position a_t, with the full record $Z_t^{\text{full}} = (\kappa_t, a_t, s_t)$, which also retains protocol state s_t. The call phase restricts the active state to either $\{\mathrm{R}, \mathrm{T}\}$ or $\{\mathrm{W}, \mathrm{I}\}$. Among four-bit encodings, `0011` and `0101` preserve the distinction within the supported set and satisfy the theorem's fidelity condition; `0110` groups states whose distinction is already determined by the phase. Figures 5(b)–(d) show that Brier improvement $\widehat{G}$ follows the squared distribution difference $\widehat{I}$. The successor distributions give the base-record reference for `0110`, whereas `0011` exhibits the predictive gain of Equation (5).

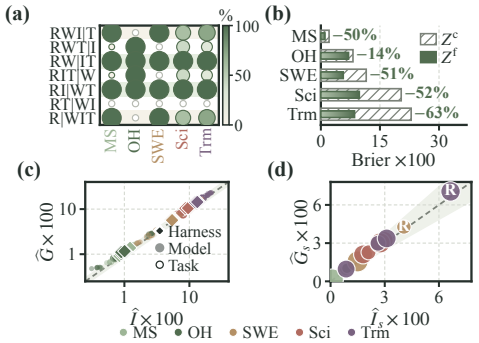

Figure 5: State fidelity from successor prediction. (a) Predictive cost $\widehat{I}$; (b) Brier improvement $\widehat{G}$; (c) $\widehat{G}$–$\widehat{I}$ alignment; (d) state-dependent gain.

2.3 Binding State Readout to a Code Workspace

To represent operation-dependent grouping, introduce the extended grouping operator $\bar{\mathbf{L}} = \mathrm{diag}(\mathbf{L}, \mathbf{I}_k)$, where $\mathbf{I}_k$ is the $k \times k$ identity matrix, and treat exit categories as absorbing states. The extended transition kernels are

$$\mathbf{P}_u = \begin{pmatrix} \mathbf{Q}_u & \mathbf{R}_u \\ 0 & \mathbf{I}_k \end{pmatrix}, \qquad \widetilde{\mathbf{P}}_u = \begin{pmatrix} \widetilde{\mathbf{Q}}_u & \widetilde{\mathbf{R}}_u \\ 0 & \mathbf{I}_k \end{pmatrix}.$$

The representation-defect operator induced by operation u is

$$\mathbf{E}_u = \mathbf{P}_u\bar{\mathbf{L}} - \bar{\mathbf{L}}\widetilde{\mathbf{P}}_u,$$

where $\mathbf{E}_u \in \mathbb{R}^{(n+k)\times(r+k)}$. The operator records the one-step departure between the fine and grouped transition operators. Its exposure under the state mass reached at each step and its subsequent propagation determine the evaluation-error bound, derived in Appendix B.6. At the code-workspace interface, the descriptor, function identifier, and test rule share a coordinate system; the corresponding recoding construction is given in Appendix C.5.

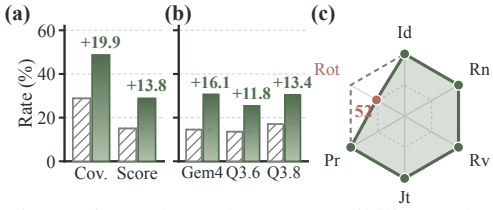

Figure 6: Code-readout compatibility under coordinate permutation. Hatched bars: rotated code–task associations; solid bars: preserved associations. (a) Coverage and task score; (b) model-wise breakdown; (c) readout accuracy under six recodings.

In code tasks such as SciCode (Tian et al., 2024), the task descriptor, function identifier, and test rule jointly define the readout map. Readable-code coverage is the fraction of code that the task protocol can parse uniquely and successfully. We report the task score on its normalized evaluation scale, so each percentage below is an absolute change on that scale. We use a controlled coordinate-permutation experiment to examine this physical binding. A synchronized permutation of the descriptor, function identifier, and test rule, corresponding to the synchronized recoding in Section 2.2, leaves the evaluation input and score unchanged. The paired setting changes only the function-name coordinate while keeping the test rule fixed. Preserving the binding increases readable-code coverage by 19.84% and task score by 13.79% (Figure 6). These two percentages use the normalized scale of their respective observables. The result shows that state fidelity must pass through the text surface and remain compatible with the physical workspace, which motivates the explicit readout boundary $\mathcal{D}_\tau$.

2.4 Conditional State Control in Adaptive Decisions

Adaptive Harnesses synthesize execution policies from feedback (Lee et al., 2026; Lou et al., 2026).

Theorem 2 requires an adaptive rule to retain the conditional details read by later operations, and Equation (5) binds an update to the state that generated its transition. Code snapshots, task descriptions, and public game states provide these conditions for version selection, classification routing, and action choice. We examine the relation in three Qwen3.8-27B protocols.

In the code self-evolution task based on FiNER (Loukas et al., 2022), each test receipt is attached to the successor state of the code snapshot x that generated it, and the pair $(x, \text{receipt})$ is used for candidate selection. Candidate-version selection accuracy increases by 7.00%, and subproblem accuracy by an equal-weight average of 5.45%. Conditioning task routing on the task description and historical feedback increases macro-average accuracy by 1.41% overall and by 1.22% on OOD9. In TextArena (Guertler et al., 2025), retaining the public game state read by the next action preserves the condition used for that action; single-player reward increases by 2.1%, two-player win rate by 2.97%, and action legality by 10.93%. Table 1 summarizes the central observations for each mechanism. Conditional state dynamics supplies an operational criterion for state drift and action collapse in long-horizon execution, and the next section turns that criterion into an online readout boundary for code workspaces.

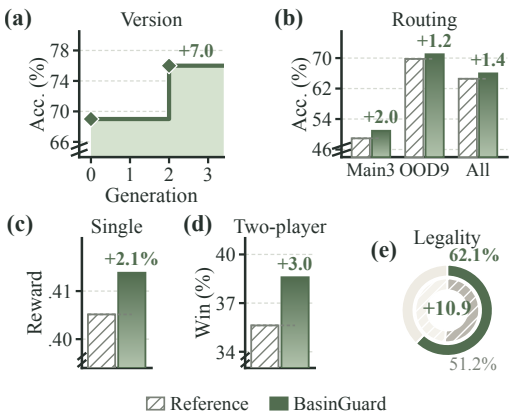

Figure 7: Adaptive decisions, Qwen3.8-27B. (a) Version selection; (b) routing; (c) single-player reward; (d) two-player win rate; (e) legality.

Table 1: Mechanism signatures. The Brier entry is the median across the five Harnesses; r denotes Pearson correlation.

Relation	Readout	Value
Finite response	Kernel–tail agreement	$r = .88$
Attractor	Partition residence scale	$r = .95$
State fidelity	Successor Brier reduction (median)	50.6%
Code–workspace	Readable-code coverage	$+19.84\%$
Code–workspace	Normalized task score	$+13.79\%$
Adaptive selection	FiNER version selection	$+7.00\%$
Adaptive control	Action legality	$+\mathbf{10.93}\%$

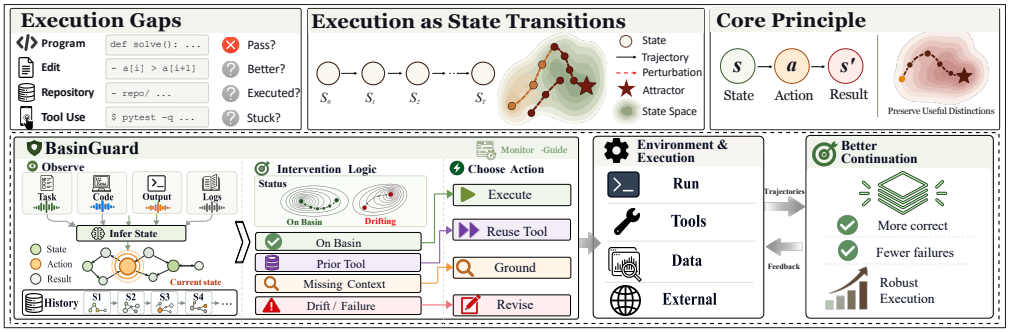

Figure 8: BasinGuard's online gating: task readout at readable protocol states, otherwise continued conditional evolution until the task object forms.

3 BasinGuard: Online State Control

Long-horizon code interaction reaches task evaluation through two coupled state changes: the model changes the program representation, while execution changes the workspace facts that make the program readable (Ning et al., 2026). Their temporal relation determines when a task object has formed, and the local attractor structure determines how long the interaction remains in a conditional state. Building on the state-fidelity and physical-binding analysis, we propose BasinGuard, an online control plugin that treats task-object formation as a readable-state boundary and chooses between continued evolution and task readout.

3.1 Algebraic Gating at the Task-Readout Boundary

Fix system conditions $c = (h, m, \mathcal{P})$ and let $\tau \sim \mathcal{P}$ be a task. At interaction point t, H_t induces the control-sufficient state $Z_t = \Phi_c(H_t) \in \mathcal{Z}_c$. The task protocol extracts the readout state

$$x_t = \psi_\tau(Z_t) \in \mathcal{X}_\tau,$$

where ψ_τ retains the task requirements, current code, workspace image, and execution receipts that affect object parsing. We decompose a public state into a public output $o_\tau(x) \in \mathcal{O}_\tau$ and physical workspace facts $w_\tau(x) \in \mathcal{W}_\tau$. Let

$$\Gamma_\tau^{\mathrm{out}} : \mathcal{O}_\tau \to \mathcal{Y}_\tau \cup \{\bot\}, \qquad \Gamma_\tau^{\mathrm{str}} : \mathcal{W}_\tau \times \mathcal{O}_\tau \to \mathcal{Y}_\tau \cup \{\bot\}$$

be the direct-output and structural parsing maps, respectively. Here $\mathcal{Y}_\tau$ is the space of valid task objects, and $\bot$ indicates that the current state does not determine a unique evaluable object. The task readout map uses a fixed precedence:

$$\Gamma_\tau(x) = \begin{cases} \Gamma_\tau^{\mathrm{out}}(o_\tau(x)), & \Gamma_\tau^{\mathrm{out}}(o_\tau(x)) \in \mathcal{Y}_\tau, \\ \Gamma_\tau^{\mathrm{str}}(w_\tau(x), o_\tau(x)), & \Gamma_\tau^{\mathrm{out}}(o_\tau(x)) = \bot \text{ and } \Gamma_\tau^{\mathrm{str}}(w_\tau(x), o_\tau(x)) \in \mathcal{Y}_\tau, \\ \bot, & \text{otherwise.} \end{cases} \tag{7}$$

The readable-state set is $\mathcal{D}_\tau := \{x \in \mathcal{X}_\tau : \Gamma_\tau(x) \neq \bot\}$. Let $\bar{g}_\tau : \mathcal{Y}_\tau \cup \{\bot\} \to [0,1]$ extend the protocol score with $\bar{g}_\tau(\bot) = 0$, and write the normalized state score as $J_\tau(x) = \bar{g}_\tau(\Gamma_\tau(x))$.

The online controller uses a frozen continuation predictor together with public protocol facts. At boundary t, let a_t be the model-request index, n_t the admitted input size in tokens, and δn_t its change relative to the preceding input size, set to zero at the first observation. Let g_t count completed tool calls, e_t count tool errors in the current history, d_t count calls awaiting a result, r_t denote the no-progress streak, and $\mathfrak{o}_t \in \{0,1\}$ indicate an open task obligation. Let a_{last} be the request index of the most recent structured transition, with the cooldown condition vacuous before the first transition. The predictor receives

$$\mathbf{z}_t = \left(a_t, \log(1+n_t), \delta n_t, g_t, e_t, d_t, r_t, \frac{g_t}{\max(1,a_t)}, \frac{e_t}{\max(1,g_t)}\right)$$

and returns $q_\theta(\mathbf{z}_t) \in [0,1]$, where θ is frozen before evaluation. Let $q_\star$ be its fixed threshold, $a_\star$ the earliest eligible request index, $c_\star$ the cooldown in model requests, and $j_\star$ the episode transition limit, with j_t counting transitions already applied. Let $b_t \in \{0,1\}$ indicate the host's context-capacity boundary and $\beta_c \in \{0,1\}$ specify whether its fixed capacity policy permits a cooldown override. The capacity boundary takes priority, and the eligibility indicator is

$$\epsilon_t = \mathfrak{o}_t\, \mathbf{1}\{a_t \geq a_\star\}\, \mathbf{1}\{a_t - a_{\mathrm{last}} \geq c_\star \vee \beta_c b_t = 1\}\, \mathbf{1}\{j_t < j_\star \vee b_t = 1\}\, \mathbf{1}\{q_\theta(\mathbf{z}_t) \leq q_\star \vee b_t = 1\}.$$

The finite continuation set $\mathcal{U}_c$ contains the identity, a bounded history release, a pending-obligation re-anchor, and a pending-tool recovery. The latter operations are selected from public predicates that detect insufficient context, persistent recurrence, lack of state advance, duplicate history groups, or unresolved tool obligations. Thus the predictor supplies the conditional viability signal, while the protocol predicates determine which structured continuation is admissible. The continuation decision is tied to the current public state and the formation of the task object, so the interaction evolves according to its observed protocol state.

The predictor is a logistic function of the standardized coordinates in $\mathbf{z}_t$. Its coefficients and normalization constants are fixed before evaluation, together with the threshold, scheduling parameters, and retained-history length. Tool calls and their associated results form atomic groups, allowing the controller to retain a coherent interaction history. Repeated completed groups and the outcomes of recent calls determine the recurrence predicates, while the public task-obligation state determines whether continuation remains relevant.

History release retains the task prefix and recent atomic groups. When completed groups recur while a task obligation remains open, re-anchoring removes consecutive duplicates from the retained history. Tool recovery supplies a typed result for a pending call, making its status available to the next decision. After each operation, the controller recomputes the public state. A sequence ends when the scheduling rule admits no further operation, the state remains unchanged or repeats, or the host compaction event is reached. The resulting history then conditions the next model request.

In Pi, the plugin represents the public tool catalog through a typed interface and associates each call with its execution receipt. A fixed allocation derived from the context window and output reserve determines how much of each tool result enters the next request. The request interface adjusts the output allowance to the remaining capacity, and the host executes subsequent calls and applies its task evaluator. Appendix E.1 gives the integration details. BasinGuard places a binary gate at the boundary of the readable set:

$$\pi_{\mathrm{BG}}(x) = \begin{cases} \mathsf{r}, & x \in \mathcal{D}_\tau, \\ \mathsf{e}, & x \notin \mathcal{D}_\tau. \end{cases} \tag{8}$$

The branch r applies the task readout $\bar{g}_\tau \circ \Gamma_\tau$, whereas e continues the conditional evolution. BasinGuard determines this branch at the readout boundary and retains the same parser and score map. Let $\mathcal{A}_c : \mathcal{Z}_c \times (\mathcal{X}_\tau \setminus \mathcal{D}_\tau) \to \mathcal{U}_c$ select the state-dependent continuation. It returns the identity when $\epsilon_t = 0$ and composes the admissible structured operations associated with the active public predicates when $\epsilon_t = 1$. The controlled transition is

$$u_t^{\mathrm{BG}} = \mathcal{A}_c(Z_t^{\mathrm{BG}}, x_t^{\mathrm{BG}}),\ H_{t+1}^{\mathrm{BG}} \sim \mathsf{P}_c(\cdot \mid H_t^{\mathrm{BG}}, u_t^{\mathrm{BG}}),\ x_{t+1}^{\mathrm{BG}} = \psi_\tau \circ \Phi_c(H_{t+1}^{\mathrm{BG}}). \tag{9}$$

Here P_c is the history transition kernel induced by the host Harness. The identity action preserves the native evolution, while a structured action changes the public protocol state according to its predicate. The controller may apply several admissible operations between model or tool events under the scheduling and capacity rules above. BasinGuard uses the frozen predictor $q_\theta(\mathbf{z}_t)$ to assess continuation from the observed history, with the public predicates selecting the operation and Γ_τ determining when task readout is available. Let $\mathcal{I}_{\tau,\ell}$ be the observable interaction points after model or tool requests, and let $\zeta_{\tau,\ell}$ be the protocol termination boundary.

$$\sigma_{\tau,\ell}^{\mathrm{BG}} := \min\big(\{\zeta_{\tau,\ell}\} \cup \{t \in \mathcal{I}_{\tau,\ell} : x_t^{\mathrm{BG}} \in \mathcal{D}_\tau\}\big),\ J_{\tau,\ell}^{\mathrm{BG}}(m,h) := \bar{g}_\tau\big(\Gamma_\tau(x_{\sigma_{\tau,\ell}^{\mathrm{BG}}}^{\mathrm{BG}})\big). \tag{10}$$

3.2 Task-Object Readout Across Interaction Interfaces

GitTaskBench couples program construction, command-line execution facts, and assertion-based evaluation (Ni et al., 2026). We evaluate BasinGuard on code-workspace tasks under three interaction architectures, Pi, Aider, and DeepSeek Harness, using Qwen3.8-27B and Muse-Glimmer-30B (Qwen Team, 2026c; Meta Superintelligence Lab, 2026).

Each comparison starts from a shared public checkpoint and uses the same readout map and task evaluator on both paths. We average score differences within tasks and then across model–Harness conditions, giving every task equal weight; we denote this task-equal difference by Δ_{task}. Further details of the score definition and readout-boundary decomposition appear in Appendix E.1.

Table 2: BasinGuard gains over the host Harness. GitTaskBench reports equal-task score changes; GAIA2 entries give the official pass rate before and after BasinGuard (%).

GitTaskBench Δ_{task} (%)					*GAIA2* official pass rate (%)				
Model	Pi	Aider	DeepSeek	Mean	Model	Pi	Aider	DeepSeek	Mean
Qwen3.8	+19.57	+8.70	+23.19	**+17.15**	Qwen3.8	4.17 → 6.46	1.67 → 2.29	4.79 → 6.46	**3.54 → 5.07**
Muse	+2.17	+9.42	+2.17	+4.59	Muse	25.83 → 28.13	13.96 → 14.58	23.13 → 24.79	**20.97 → 22.50**
All	+10.87	+9.06	+12.68	**+10.87**	All	15.00 → 17.30	7.82 → 8.44	13.96 → 15.63	**12.26 → 13.79**

Table 2 and Figure 9 summarize the configuration-level results. On GitTaskBench, the equal-weight difference is 10.87% on the normalized task-score scale; the corresponding model-wise differences are 17.15% for Qwen3.8-27B and 4.59% for Muse-Glimmer-30B. The gain is positive under all three interaction architectures. Table 2 also reports the official GAIA2 pass rate, where heterogeneous tool facts jointly form the evaluated object. BasinGuard increases the official pass rate in every model–Harness condition, extending the readable-state boundary from code workspaces to task objects formed across several tool interactions.

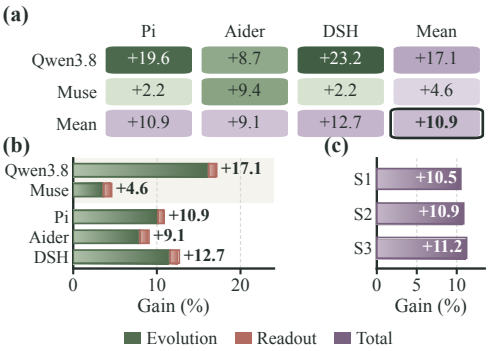

Figure 9: GitTaskBench gains (%). (a) Per-condition means; (b) evolution and readout parts; (c) paired runs S1–S3.

3.3 From Program Representation to Physical Execution Facts

Program text specifies a computation, while task evaluation reads the execution facts that make its result available in the workspace. In the Eparse task, the program state x_{prog} precedes the output required by the task, so Equation (7) gives $\Gamma_\tau(x_{\text{prog}}) = \bot$. At x_{prog}, Eq. (8) selects branch `e`. Figure 10 shows the continuation that forms `output.txt` and reaches x_{fact} with

$$x_{\text{prog}} \xrightarrow{\mathsf{e}} x_{\text{fact}}, \qquad \Gamma_\tau(x_{\text{fact}}) \in \mathcal{Y}_\tau.$$

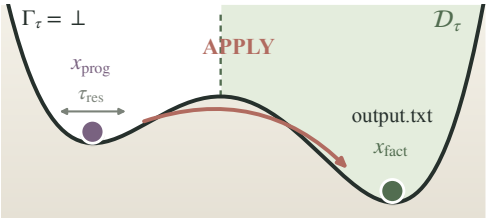

Figure 10: Eparse task-object formation. From x_{prog} ($\Gamma_\tau = \bot$), APPLY writes `output.txt` and reaches $x_{\text{fact}} \in \mathcal{D}_\tau$.

At x_{fact}, branch `r` applies the task readout. The sequence connects the program representation to the workspace fact that defines the task object. Further code-workspace instances of this relation appear in Appendix E.2.

4 Related Work

Agent Harnesses couple models to tools, workspaces, and environmental feedback, and increasingly expose this coupling as a programmable runtime or learned control policy (Yang et al., 2024; Wang et al., 2025a; Li et al., 2026d; Pan et al., 2026; Fu et al., 2026; Jiang et al., 2026). Continually rewritten Harnesses further use interaction history to revise the surrounding system (Lee et al., 2026; Lou et al., 2026; Shi et al., 2026). We connect these systems to quasi-stationary dynamics, state abstraction, and predictive-state representations (Darroch & Seneta, 1965; Givan et al., 2003; Singh et al., 2012); Appendix A situates the work more fully.

5 Conclusion

We developed a conditional state-dynamics framework for agent Harnesses that formalizes finite response, conditional attractor structure, and operation-dependent state fidelity. Results across models and benchmarks show that the value of a state representation is determined by the conditional distinctions that later operations can read. Based on this relation, BasinGuard provides online control of the task readout boundary and yields gains across code-workspace Harnesses.

AI USE STATEMENT

Except for the evaluated model interactions and benchmarks that naturally involve LLMs, generative AI tools were used solely for manuscript polishing, assisting in drafting experimental scripts, and cross-checking the correctness of mathematical derivations developed by the authors. The authors reviewed and verified all code implementations, theoretical derivations, and text, and take full responsibility for the manuscript and its artifacts.

ETHICS STATEMENT

This study uses publicly available benchmarks, open-source code repositories, interaction logs, and open-weight language models. It does not involve human participants, private personal data, or sensitive demographic attributes. All code executions and agent interactions were conducted in isolated container environments.

REPRODUCIBILITY STATEMENT

Sections 2 and 3 specify the formal definitions, operational principles, and algorithmic procedures for the conditional state dynamics framework and the BasinGuard online controller. Complete mathematical proofs for Theorem 1 and Theorem 2 are presented in Appendix A. Experimental configurations, benchmark task partitions, discrete protocol state conventions, and evaluation metric details are fully documented across the main text and Appendices C–E. An anonymous repository in the supplementary material provides the complete source code and detailed documentation for all research artifacts.

REFERENCES

David Abel, David Hershkowitz, and Michael Littman. Near optimal behavior via approximate state abstraction. In Maria Florina Balcan and Kilian Q. Weinberger (eds.), *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 2915–2923, New York, New York, USA, 20–22 Jun 2016. PMLR. URL https://proceedings.mlr.press/v48/abel16.html.

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron Courville, and Marc Bellemare. Deep reinforcement learning at the edge of the statistical precipice. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 29304–29320. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/f514cec81cb148559cf475e7426eed5e-Paper.pdf.

Lakshya A Agrawal, Shangyin Tan, Dilara Soylu, Noah Ziems, Rishi Khare, Krista Opsahl-Ong, Arnav Singhvi, Herumb Shandilya, Michael J Ryan, Meng Jiang, Christopher Potts, Koushik Sen, Alex Dimakis, Ion Stoica, Dan Klein, Matei Zaharia, and Omar Khattab. GEPA: Reflective prompt evolution can outperform reinforcement learning. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=RQm2KQTM5r.

Francesco Barbieri, Jose Camacho-Collados, Luis Espinosa Anke, and Leonardo Neves. TweetEval: Unified benchmark and comparative evaluation for tweet classification. In Trevor Cohn, Yulan He, and Yang Liu (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 1644–1650, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.148. URL https://aclanthology.org/2020.findings-emnlp.148/.

Anton Bovier, Michael Eckhoff, Véronique Gayrard, and Markus Klein. Metastability and low lying spectra¶in reversible markov chains. *Communications in Mathematical Physics*, 228 (2):219–255, 2002. doi: 10.1007/s002200200609. URL https://doi.org/10.1007/s002200200609.

GLENN W. BRIER. Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 78(1):1 – 3, 1950. doi: 10.1175/1520-0493(1950)078⟨0001:VOFEIT⟩2.0.CO;2. URL https://journals.ametsoc.org/view/journals/mwre/78/1/1520-0493_1950_078_0001_vofeit_2_0_co_2.xml.

Zhiqiang Cai, Ling Lin, and Xiang Zhou. Learn quasi-stationary distributions of finite state markov chain. *Entropy*, 24(1), 2022. ISSN 1099-4300. doi: 10.3390/e24010133. URL https://www.mdpi.com/1099-4300/24/1/133.

Iñigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. Efficient intent detection with dual sentence encoders. In Tsung-Hsien Wen, Asli Celikyilmaz, Zhou Yu, Alexandros Papangelis, Mihail Eric, Anuj Kumar, Iñigo Casanueva, and Rushin Shah (eds.), *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pp. 38–45, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.nlp4convai-1.5. URL https://aclanthology.org/2020.nlp4convai-1.5/.

Nicolas Champagnat and Denis Villemonais. Exponential convergence to quasi-stationary distribution and $$q$$-process. *Probability Theory and Related Fields*, 164(1):243–283, 2016. doi: 10.1007/s00440-014-0611-7. URL https://doi.org/10.1007/s00440-014-0611-7.

Xiwei Chen. State-aware runtime for long-horizon llm agents: A conceptual framework and research agenda, 2026.

Arman Cohan, Waleed Ammar, Madeleine van Zuylen, and Field Cady. Structural scaffolds for citation intent classification in scientific publications. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 3586–3596, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1361. URL https://aclanthology.org/N19-1361/.

Pierre Collet, Servet Martínez, and Jaime San Martín. Quasi-stationary distributions. *Markov chains, diffusions and dynamical systems*, 2013.

J. N. Darroch and E. Seneta. On quasi-stationary distributions in absorbing discrete-time finite markov chains. *Journal of Applied Probability*, 2(1):88–100, 1965. doi: 10.2307/3211876.

Dorottya Demszky, Dana Movshovitz-Attias, Jeongwoo Ko, Alan Cowen, Gaurav Nemade, and Sujith Ravi. GoEmotions: A dataset of fine-grained emotions. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4040–4054, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.372. URL https://aclanthology.org/2020.acl-main.372/.

Peter Deuflhard and Marcus Weber. Robust perron cluster analysis in conformation dynamics. *Linear Algebra and its Applications*, 398:161–184, 2005. ISSN 0024-3795. doi: https://doi.org/10.1016/j.laa.2004.10.026. URL https://www.sciencedirect.com/science/article/pii/S0024379504004689. Special Issue on Matrices and Mathematical Biology.

Guanting Dong, Xiaoshuai Song, Yuyang Hu, Jiajie Jin, Chenghao Zhang, Yifei Chen, Xiaoxi Li, Huaying Yuan, Xinyu Yang, Tongyu Wen, Jiejun Tan, Hongjin Qian, Shijue Huang, Junting Lu, Zhenyu Li, Wanjun Zhong, Yutao Zhu, Tat-Seng Chua, Zhicheng Dou, and Ji-Rong Wen. Towards long-horizon agents: A survey. *Preprints*, July 2026. doi: 10.20944/preprints202607.1328.v1. URL https://doi.org/10.20944/preprints202607.1328.v1.

Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Roth. The reusable holdout: Preserving validity in adaptive data analysis. *Science*, 349(6248):636–638, 2015. doi: 10.1126/science.aaa9375. URL https://www.science.org/doi/abs/10.1126/science.aaa9375.

Zhiwei Fei, Xiaoyu Shen, Dawei Zhu, Fengzhe Zhou, Zhuo Han, Alan Huang, Songyang Zhang, Kai Chen, Zhixin Yin, Zongwen Shen, Jidong Ge, and Vincent Ng. LawBench: Benchmarking

legal knowledge of large language models. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 7933–7962, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.452. URL https://aclanthology.org/2024.emnlp-main.452/.

Jiarun Fu, Lizhong Ding, Sida Chen, Honglei Xin, Chunhui Zhang, Pengqi Li, Qiuning Wei, Ye Yuan, and Guoren Wang. CHILL-Harness: Counterfactual harness learning for efficient reasoning in long-horizon agents, 2026. URL https://arxiv.org/abs/2607.25825.

Bernhard C. Geiger and Christoph Temmel. Lumpings of markov chains, entropy rate preservation, and higher-order lumpability. *Journal of Applied Probability*, 51(4):1114–1132, December 2014. ISSN 1475-6072. doi: 10.1239/jap/1421763331. URL http://dx.doi.org/10.1239/jap/1421763331.

Gemma Team. Gemma 4 technical report, 2026. URL https://arxiv.org/abs/2607.02770.

Robert Givan, Thomas Dean, and Matthew Greig. Equivalence notions and model minimization in markov decision processes. *Artificial Intelligence*, 147(1):163–223, 2003. ISSN 0004-3702. doi: https://doi.org/10.1016/S0004-3702(02)00376-4. URL https://www.sciencedirect.com/science/article/pii/S0004370202003764. Planning with Uncertainty and Incomplete Information.

Shangding Gu. From model scaling to system scaling: Scaling the harness in agentic ai, 2026. URL https://arxiv.org/abs/2605.26112.

Leon Guertler, Bobby Cheng, Simon Yu, Bo Liu, Leshem Choshen, and Cheston Tan. Textarena, 2025. URL https://arxiv.org/abs/2504.11442.

Jianyuan Guo, Zhiwei Hao, Chengcheng Wang, Cheng Fan, Tingzhang Luo, Hongguang Li, Ying Gao, Hefei Mei, Jiankun Peng, Rongjian Xu, Minjing Dong, Han Wu, Mengyu Zheng, Kai Han, Shiqi Wang, Chang Xu, and Yunhe Wang. From question answering to task completion: A survey on agent system and harness design, 2026. URL https://arxiv.org/abs/2606.20683.

Xuehang Guo, Xingyao Wang, Yangyi Chen, Sha Li, Chi Han, Manling Li, and Heng Ji. SyncMind: Measuring agent out-of-sync recovery in collaborative software engineering. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 20993–21066. PMLR, 2025. URL https://proceedings.mlr.press/v267/guo25l.html.

J J Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79(8):2554–2558, 1982. doi: 10.1073/pnas.79.8.2554. URL https://www.pnas.org/doi/abs/10.1073/pnas.79.8.2554.

Shengran Hu, Cong Lu, and Jeff Clune. Automated design of agentic systems. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 21344–21377, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/36b7acf6f6010652b3f2a433774a66fe-Paper-Conference.pdf.

Xinke Jiang, Zhixin Zhang, Zhibang Yang, Jiaran Gao, Rihong Qiu, Shijin Chen, Xu Chu, Junfeng Zhao, and Yasha Wang. Harness-RL: Black-box reinforcement learning with action-args decoupling for central-agent multi-agent harnesses, 2026. URL https://arxiv.org/abs/2608.29641.

Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik R Narasimhan. SWE-bench: Can language models resolve real-world github issues? In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=VTF8yNQM66.

Phillip Keung, Yichao Lu, György Szarvas, and Noah A. Smith. The multilingual Amazon reviews corpus. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 4563–4568, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.369. URL https://aclanthology.org/2020.emnlp-main.369/.

Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan A, Saiful Haq, Ashutosh Sharma, Thomas T. Joshi, Hanna Moazam, Heather Miller, Matei Zaharia, and Christopher Potts. DSPy: Compiling declarative language model calls into state-of-the-art pipelines. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=sY5N0zY5Od.

Tushar Khot, Ashish Sabharwal, and Peter Clark. Scitail: A textual entailment dataset from science question answering. In Sheila A. McIlraith and Kilian Q. Weinberger (eds.), *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pp. 5189–5197. AAAI Press, 2018. doi: 10.1609/AAAI.V32I1.12022. URL https://doi.org/10.1609/aaai.v32i1.12022.

Yoonho Lee, Roshen Nair, Qizheng Zhang, Kangwook Lee, Omar Khattab, and Chelsea Finn. Meta-harness: End-to-end optimization of model harnesses, 2026. URL https://arxiv.org/abs/2603.28052.

Han Li, Yifan Yao, Letian Zhu, Rili Feng, Hongyi Ye, Jiaming Wang, Yancheng He, Pengyu Zou, Lehan Zhang, Xinping Lei, Haoyang Huang, Ken Deng, Ming Sun, Zhaoxiang Zhang, He Ye, and Jiaheng Liu. Codetracer: Towards traceable agent states, 2026a. URL https://arxiv.org/abs/2604.11641.

Junjie Li, Xi Xiao, Yunbei Zhang, Chen Liu, Lin Zhao, Xiaoying Liao, Yingrui Ji, Janet Wang, Yingqiang Ge, Weijie Xu, Xi Fang, Xiang Xu, Tianchen Zhao, Youngeun Kim, Jihun Hamm, Tianyang Wang, and Chandan Reddy. Agent harness engineering: A survey, 2026b. URL https://openreview.net/pdf?id=eONq7FdiHa.

Lihong Li, Thomas J. Walsh, and Michael L. Littman. Towards a unified theory of state abstraction for mdps. In *International Symposium on Artificial Intelligence and Mathematics, AI&Math 2006, Fort Lauderdale, Florida, USA, January 4-6, 2006*, 2006. URL http://anytime.cs.umass.edu/aimath06/proceedings/P21.pdf.

Xiangyi Li, Yimin Liu, Wenbo Chen, Bingran You, Zonglin Di, Yifeng He, Shenghan Zheng, Kyoung Whan Choe, Jiankai Sun, Shuyi Wang, Chujun Tao, Binxu Li, Xuandong Zhao, Hejia Geng, Xiaojun Wu, Junwei Zhou, Xiaokun Chen, Hanwen Xing, Yubo Li, Qunhong Zeng, Di Wang, Yuanli Wang, Roey Ben Chaim, Penghao Jiang, Haotian Shen, Luyang Kong, Xinyi Liu, Runhui Wang, Xuanqing Liu, Haoran Lyu, Yizhuo He, Yuchen Tian, Shutong Wu, Bowei Wang, Yixuan Gao, Bo Chen, Litong Liu, Sikai Cheng, Jiajun Bao, Shuaicheng Tong, Shuwen Xu, Terry Yue Zhuo, Tinghan Ye, Qi Qi, Miao Li, Longtai Liao, Zelin Tan, Chang Shi, Xilin Tang, Srinath Tankasala, Boqin Yuan, Yaoyao Qian, Jianhong Tu, Chenguang Wang, Yizhou Sun, Wei Wang, Aaron Taylor, Ziyue Yang, Changkun Guan, Zhikang Dong, Xinyu Zhang, Steven Dillmann, Han chung Lee, and Dawn Song. Skillsbench: Benchmarking how well agent skills work across diverse tasks, 2026c. URL https://arxiv.org/abs/2602.12670.

Zhening Li, Joshua Liu, Mateja Vukelic, Nicole Shen, Supriya Lall, Amitayush Thakur, Alex Zhang, Omar Khattab, Jonathan Light, and Armando Solar-Lezama. Harness as a language: A minimalist agent framework with maximal expressivity, 2026d. URL https://arxiv.org/abs/2609.26891.

Jiahang Lin, Shichun Liu, Chengjun Pan, Lizhi Lin, Shihan Dou, Zhiheng Xi, Xuanjing Huang, Hang Yan, Zhenhua Han, Tao Gui, and Yu-Gang Jiang. Agentic harness engineering: Observability-driven automatic evolution of coding-agent harnesses, 2026. URL https://arxiv.org/abs/2604.25850.

Chengzhi Liu, Yichen Guo, Yepeng Liu, Yuzhe Yang, Qianqi Yan, Xuandong Zhao, Wenyue Hua, Sheng Liu, Sharon Li, Yuheng Bu, and Xin Eric Wang. Auditing agent harness safety, 2026. URL https://arxiv.org/abs/2605.14271.

Zhihan Liu, Hao Hu, Shenao Zhang, Hongyi Guo, Shuqi Ke, Boyi Liu, and Zhaoran Wang. Reason for future, act for now: A principled architecture for autonomous LLM agents. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 31186–31261. PMLR, 2024. URL https://proceedings.mlr.press/v235/liu24ab.html.

Xinghua Lou, Miguel Lázaro-Gredilla, Antoine Dedieu, Carter Wendelken, Wolfgang Lehrach, and Kevin P. Murphy. Autoharness: improving llm agents by automatically synthesizing a code harness, 2026. URL https://arxiv.org/abs/2603.03329.

Lefteris Loukas, Manos Fergadiotis, Ilias Chalkidis, Eirini Spyropoulou, Prodromos Malakasiotis, Ion Androutsopoulos, and Georgios Paliouras. FiNER: Financial numeric entity recognition for XBRL tagging. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 4419–4431, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.303. URL https://aclanthology.org/2022.acl-long.303/.

Ziyu Ma, Hailang Huang, Shun Zou, Yong Wang, Shidong Yang, Yiming Hu, Fei Wei, and XiangXiang Chu. Longhorizon-harness: Advancing long-horizon agents for real-world tasks, 2026. URL https://arxiv.org/abs/2608.01964.

Pekka Malo, Ankur Sinha, Pekka Korhonen, Jyrki Wallenius, and Pyry Takala. Good debt or bad debt: Detecting semantic orientations in economic texts. *J. Assoc. Inf. Sci. Technol.*, 65 (4):782–796, April 2014. ISSN 2330-1635. doi: 10.1002/asi.23062. URL https://doi.org/10.1002/asi.23062.

Mike A Merrill, Alexander Glenn Shaw, Nicholas Carlini, Boxuan Li, Harsh Raj, Ivan Bercovich, Lin Shi, Jeong Yeon Shin, Thomas Walshe, E. Kelly Buchanan, Junhong Shen, Guanghao Ye, Haowei Lin, Jason Poulos, Maoyu Wang, Marianna Nezhurina, Di Lu, Orfeas Menis Mastromichalakis, Zhiwei Xu, Zizhao Chen, Yue Liu, Robert Zhang, Leon Liangyu Chen, Anurag Kashyap, Jan-Lucas Uslu, Jeffrey Li, Jianbo Wu, Minghao Yan, Song Bian, Vedang Sharma, Ke Sun, Steven Dillmann, Akshay Anand, Andrew Lanpouthakoun, Bardia Koopah, Changran Hu, Etash Kumar Guha, Gabriel H. S. Dreiman, Jiacheng Zhu, Karl Krauth, Li Zhong, Niklas Muennighoff, Robert Kwesi Amanfu, Shangyin Tan, Shreyas Pimpalgaonkar, Tushar Aggarwal, Xiangning Lin, Xin Lan, Xuandong Zhao, Yiqing Liang, Yuanli Wang, Zilong Wang, Changzhi Zhou, David Heineman, Hange Liu, Harsh Trivedi, John Yang, Junhong Lin, Manish Shetty, Michael Yang, Nabil Omi, Negin Raoof, Shanda Li, Terry Yue Zhuo, Wuwei Lin, Yiwei Dai, Yuxin Wang, Wenhao Chai, Shang Zhou, Dariush Wahdany, Ziyu She, Jiaming Hu, Zhikang Dong, Yuxuan Zhu, Sasha Cui, Ahson Saiyed, Arinbjörn Kolbeinsson, Christopher Michael Rytting, Ryan Marten, Yixin Wang, Jenia Jitsev, Alex Dimakis, Andy Konwinski, and Ludwig Schmidt. Terminal-bench: Benchmarking agents on hard, realistic tasks in command line interfaces. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=a7Qa4CcHak.

Meta Superintelligence Lab. Muse glimmer model card, August 2026. URL https://huggingface.co/meta-models/Muse-Glimmer-30B.

John Milnor. On the concept of attractor. *Communications in Mathematical Physics*, 99(2):177–195, 1985. doi: 10.1007/BF01212280. URL https://doi.org/10.1007/BF01212280.

A. Yu. Mitrophanov. Sensitivity and convergence of uniformly ergodic markov chains. *Journal of Applied Probability*, 42(4):1003–1014, 2005. doi: 10.1239/jap/1134587812.

Allan H. Murphy. A new vector partition of the probability score. *Journal of Applied Meteorology and Climatology*, 12(4):595 – 600, 1973. doi: 10.1175/1520-0450(1973)012⟨0595:ANVPOT⟩2.0.CO;2. URL https://journals.ametsoc.org/view/journals/apme/12/4/1520-0450_1973_012_0595_anvpot_2_0_co_2.xml.

Ziyi Ni, Huacan Wang, Shuo Zhang, Shuo Lu, Ziyang He, Wang You, Zhenheng Tang, Sen Hu, Bo Li, Chen Hu, Binxing Jiao, Daxin Jiang, Yuntao Du, and Pin Lyu. Gittaskbench: A benchmark for code agents solving real-world tasks through code repository leveraging. In Sven Koenig, Chad Jenkins, and Matthew E. Taylor (eds.), *Fortieth AAAI Conference on Artificial Intelligence, Thirty-Eighth Conference on Innovative Applications of Artificial Intelligence, Sixteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2026, Singapore, January 20-27, 2026*, pp. 32564–32572. AAAI Press, 2026. doi: 10.1609/AAAI.V40I38.40533. URL https://doi.org/10.1609/aaai.v40i38.40533.

Erik Nijkamp, Anurag Koul, Egor Pakhomov, and Bo Pang. An architecture for long-horizon agents: Levels, ticks and cascaded intelligence, 2026. URL https://arxiv.org/abs/2609.19519.

Xuying Ning, Katherine Tieu, Dongqi Fu, Tianxin Wei, Zihao Li, Yuanchen Bei, Jiaru Zou, Mengting Ai, Zhining Liu, Ting-Wei Li, Lingjie Chen, Yanjun Zhao, Ke Yang, Bingxuan Li, Cheng Qian, Gaotang Li, Xiao Lin, Zhichen Zeng, Ruizhong Qiu, Sirui Chen, Yifan Sun, Xiyuan Yang, Ruida Wang, Rui Pan, Chenyuan Yang, Dylan Zhang, Liri Fang, Zikun Cui, Yang Cao, Pan Chen, Dorothy Sun, Ren Chen, Mahesh Srinivasan, Nipun Mathur, Yinglong Xia, Hong Li, Hong Yan, Pan Lu, Lingming Zhang, Tong Zhang, Hanghang Tong, and Jingrui He. Code as agent harness, 2026. URL https://arxiv.org/abs/2605.18747.

Sara Oliver-Bonafoux, Javier Aguilar, Tobias Galla, and Raúl Toral. Numerical methods for quasi-stationary distributions. *Phys. Rev. E*, 113:034316, Mar 2026. doi: 10.1103/8qrl-jnzc. URL https://link.aps.org/doi/10.1103/8qrl-jnzc.

Siru Ouyang, Jun Yan, I-Hung Hsu, Yanfei Chen, Ke Jiang, Zifeng Wang, Rujun Han, Long Le, Samira Daruki, Xiangru Tang, Vishy Tirumalashetty, George Lee, Mahsan Rofouei, Hangfei Lin, Jiawei Han, Chen-Yu Lee, and Tomas Pfister. Reasoningbank: Scaling agent self-evolving with reasoning memory. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=jL7fwchScm.

Linyue Pan, Lexiao Zou, Shuo Guo, Jingchen Ni, and Hai-Tao Zheng. Natural-language agent harnesses, 2026. URL https://arxiv.org/abs/2603.25723.

PK Pollett and Erik A van Doorn. Quasi-stationary distributions for reducible absorbing markov chains in discrete time, 2008.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026a. URL https://qwen.ai/blog?id=qwen3.5.

Qwen Team. Qwen3.6-27B: Flagship-level coding in a 27B dense model, April 2026b. URL https://qwen.ai/blog?id=qwen3.6-27b.

Qwen Team. Qwen3.8-Max: A new bar for coding and cowork, August 2026c. URL https://qwen.ai/blog?id=qwen3.8.

Hubert Ramsauer, Bernhard Schäfl, Johannes Lehner, Philipp Seidl, Michael Widrich, Lukas Gruber, Markus Holzleitner, Thomas Adler, David Kreil, Michael K Kopp, Günter Klambauer, Johannes Brandstetter, and Sepp Hochreiter. Hopfield networks is all you need. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=tL89RnzIiCd.

David Ruelle. A review of linear response theory for general differentiable dynamical systems. *Nonlinearity*, 22(4):855–870, March 2009. ISSN 1361-6544. doi: 10.1088/0951-7715/22/4/009. URL http://dx.doi.org/10.1088/0951-7715/22/4/009.

Nadine Schneider, Nikolaus Stiefl, and Gregory A. Landrum. What's what: The (nearly) definitive guide to reaction role assignment. *J. Chem. Inf. Model.*, 56(12):2336–2346, 2016. doi: 10.1021/ACS.JCIM.6B00564. URL https://doi.org/10.1021/acs.jcim.6b00564.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In Francis Bach and David Blei (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp.

1889–1897, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/schulman15.html.

Yifan Shi, Wei Zhang, and Tianyi Cui. A programming paradigm for spatiotemporal composability, 2026. URL https://arxiv.org/abs/2608.25512.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/1b44b878bb782e6954cd888628510e90-Abstract-Conference.html.

Stefan Siegert. Simplifying and generalising murphy's brier score decomposition. *Quarterly Journal of the Royal Meteorological Society*, 143(703):1178–1183, 2017.

Satinder Singh, Michael James, and Matthew Rudary. Predictive state representations: A new theory for modeling dynamical systems, 2012. URL https://arxiv.org/abs/1207.4167.

Sukesh Subaharan. Controlling long-horizon behavior in language model agents with explicit state dynamics, 2026. URL https://arxiv.org/abs/2601.16087.

Minyang Tian, Luyu Gao, Shizhuo Dylan Zhang, Xinan Chen, Cunwei Fan, Xuefei Guo, Roland Haas, Pan Ji, Kittithat Krongchon, Yao Li, Shengyan Liu, Di Luo, Yutao Ma, Hao Tong, Kha Trinh, Chenyu Tian, Zihan Wang, Bohao Wu, Yanyu Xiong, Shengzhu Yin, Minhui Zhu, Kilian Lieret, Yanxin Lu, Genglin Liu, Yufeng Du, Tianhua Tao, Ofir Press, Jamie Callan, Eliu Huerta, and Hao Peng. Scicode: A research coding benchmark curated by scientists. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 30624–30650. Curran Associates, Inc., 2024. doi: 10.52202/079017-0963. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/36850592258c8c41cecdaa3dea5ff7de-Paper-Datasets_and_Benchmarks_Track.pdf.

Erik A. van Doorn and Philip K. Pollett. Quasi-stationary distributions for discrete-state models. *European Journal of Operational Research*, 230(1):1–14, 2013. ISSN 0377-2217. doi: https://doi.org/10.1016/j.ejor.2013.01.032. URL https://www.sciencedirect.com/science/article/pii/S0377221713000799.

Xingyao Wang, Yangyi Chen, Lifan Yuan, Yizhe Zhang, Yunzhu Li, Hao Peng, and Heng Ji. Executable code actions elicit better LLM agents. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 50208–50232. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/wang24h.html.

Xingyao Wang, Boxuan Li, Yufan Song, Frank F. Xu, Xiangru Tang, Mingchen Zhuge, Jiayi Pan, Yueqi Song, Bowen Li, Jaskirat Singh, Hoang H. Tran, Fuqiang Li, Ren Ma, Mingzhang Zheng, Bill Qian, Daniel Shao, Niklas Muennighoff, Yizhe Zhang, Binyuan Hui, Junyang Lin, Robert Brennan, Hao Peng, Heng Ji, and Graham Neubig. Openhands: An open platform for AI software developers as generalist agents. In *The Thirteenth International Conference on Learning Representations*, 2025a. URL https://openreview.net/forum?id=OJd3ayDDoF.

Zora Zhiruo Wang, Jiayuan Mao, Daniel Fried, and Graham Neubig. Agent workflow memory. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 63897–63911. PMLR, 2025b. URL https://proceedings.mlr.press/v267/wang25bx.html.

Wenyi Wu, Sibo Zhu, Kun Zhou, Aayush Salvi, Zixuan Song, and Biwei Huang. Structagent: Harness long-horizon digital agents with unified causal structure, 2026. URL https://arxiv.org/abs/2607.11388.

Chunqiu Steven Xia, Yinlin Deng, Soren Dunn, and Lingming Zhang. Agentless: Demystifying llm-based software engineering agents, 2024. URL https://arxiv.org/abs/2407.01489.

John Yang, Carlos E. Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. Swe-agent: agent-computer interfaces enable automated software engineering. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA, 2024. Curran Associates Inc. ISBN 9798331314385.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=WE_vluYUL-X.

Mert Yuksekgonul, Federico Bianchi, Joseph Boen, Sheng Liu, Pan Lu, Zhi Huang, Carlos Guestrin, and James Zou. Optimizing generative ai by backpropagating language model feedback. *Nature*, 639(8055):609–616, 2025. doi: 10.1038/s41586-025-08661-4. URL https://doi.org/10.1038/s41586-025-08661-4.

Cheng Zhang, Minnan Luo, Zesheng Yang, MING LI, Yong-Jin Liu, and Qinghua Zheng. Controlled dynamics attractor transformer. In *Forty-third International Conference on Machine Learning*, 2026a. URL https://openreview.net/forum?id=WLRuaimgMw.

Jiayi Zhang, Jinyu Xiang, Zhaoyang Yu, Fengwei Teng, Xiong-Hui Chen, Jiaqi Chen, Mingchen Zhuge, Xin Cheng, Sirui Hong, Jinlin Wang, Bingnan Zheng, Bang Liu, Yuyu Luo, and Chenglin Wu. AFlow: Automating agentic workflow generation. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=z5uVAKwmjf.

Qizheng Zhang, Changran Hu, Shubhangi Upasani, Boyuan Ma, Fenglu Hong, Vamsidhar Kamanuru, Jay Rainton, Chen Wu, Mengmeng Ji, Hanchen Li, Urmish Thakker, James Y Zou, and Kunle Olukotun. Agentic context engineering: Evolving contexts for self-improving language models. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 86069–86100, 2026b. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/8a94ff6f922d995d7d3f4ebf4143e442-Paper-Conference.pdf.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper_files/paper/2015/file/250cf8b51c773f3f8dc8b4be867a9a02-Paper.pdf.

Xiangxin Zhao, Han Li, Shuaiting Li, Tianyi Zhao, Earl T. Barr, Federica Sarro, and He Ye. Failure as a process: An anatomy of cli coding agent trajectories, 2026. URL https://arxiv.org/abs/2607.09510.

Jingsheng Zheng, Xinyuan Fang, Jintian Zhang, Zhengke Gui, Huajun Chen, and Ningyu Zhang. Onedayagent: Towards a long-horizon harness for autonomous agents, 2026. URL https://arxiv.org/abs/2608.05013.

A Extended Related Work

A.1 Long-Horizon Interaction and Continually Rewritten Harnesses

Long-horizon agents organize models, tools, workspaces, and environmental feedback into a continuing decision process. Recent surveys and system-level accounts describe the Harness as a central source of long-horizon capability, complementing model scaling with runtime structure and task coordination (Gu, 2026; Guo et al., 2026; Li et al., 2026b; Dong et al., 2026; Nijkamp et al., 2026). ReAct alternates reasoning and action within a common loop, CodeAct expresses actions as executable code, SWE-Agent organizes repository navigation, editing, and testing around a software environment, and OpenHands extends this style of interaction into a continuously running development platform (Yao et al., 2023; Wang et al., 2024; Yang et al., 2024; Wang et al., 2025a). Agentless separates localization, editing, and evaluation in software repair (Xia et al., 2024), while

SyncMind studies desynchronization and recovery as collaborative development changes a codebase (Guo et al., 2025). Together, these works make clear that the relation among visible state, available actions, and environmental feedback is part of the task itself.

Corresponding benchmarks have moved evaluation beyond a single output to repository state, command-line facts, executed scientific programs, and skill selection in an exposed environment (Jimenez et al., 2024; Merrill et al., 2026; Tian et al., 2024; Ni et al., 2026; Guertler et al., 2025; Li et al., 2026c). LongHorizon-Harness studies state persistence and completion conditions in extended tasks, CodeTracer follows heterogeneous execution states, and Failure as a Process treats failure as something that develops over an interaction (Ma et al., 2026; Li et al., 2026a; Zhao et al., 2026). In such tasks, whether a distinction in the history remains consequential is determined by the conditions that later decisions and task evaluation can read.

A Harness can also be written as an explicit agent language, expressed through natural-language components, or organized as a persistent long-horizon runtime (Li et al., 2026d; Pan et al., 2026; Zheng et al., 2026). Reflexion writes language feedback from failed attempts into the memory used by later attempts, while Agent Workflow Memory extracts reusable workflows from prior interactions (Shinn et al., 2023; Wang et al., 2025b). Once the Harness itself becomes an object of search, automated design has expanded to agent structures, workflows, and feedback memory. DSPy compiles declarative model calls into optimizable programs; ADAS and AFlow search over agent structures and workflows; and TextGrad and GEPA update systems or prompts from language feedback (Khattab et al., 2024; Hu et al., 2025; Zhang et al., 2025; Yuksekgonul et al., 2025; Agrawal et al., 2026). ACE and ReasoningBank organize context and experiential memory that evolve with interaction (Zhang et al., 2026b; Ouyang et al., 2026). Meta-Harness, AutoHarness, and Agentic Harness Engineering incorporate historical code, environmental feedback, and execution traces into continual Harness rewriting (Lee et al., 2026; Lou et al., 2026; Lin et al., 2026); StructAgent introduces a unified causal structure for long-horizon digital agents, while CHILL-Harness and Harness-RL learn Harness-level policies from interaction (Wu et al., 2026; Fu et al., 2026; Jiang et al., 2026). Cordis formalizes the effects and dependencies of components on shared context through spatiotemporal composability (Shi et al., 2026). Code as Agent Harness and work on Harness safety further emphasize that program structure, tool interfaces, and runtime mediation determine which actions and facts become available to the agent (Ning et al., 2026; Liu et al., 2026). We study when such continual rewriting preserves the conditional facts required by later decisions and task evaluation.

A.2 Conditional Dynamics, State Abstraction, and Predictive Information

Quasi-stationary distributions describe long-run visitation before absorption, conditional on a process remaining in a local phase. From finite absorbing chains to general Markov processes, prior work establishes existence and convergence conditions for quasi-stationary distributions and develops corresponding discrete-state, reducible-chain, learning, and numerical methods (Darroch & Seneta, 1965; Champagnat & Villemonais, 2016; Collet et al., 2013; van Doorn & Pollett, 2013; Pollett & van Doorn, 2008; Cai et al., 2022; Oliver-Bonafoux et al., 2026). Probabilistic definitions of attractors, low-frequency spectral structure in metastable systems, and Perron clustering describe long-lived visitation sets and their slow modes from geometric and spectral perspectives (Milnor, 1985; Bovier et al., 2002; Deuflhard & Weber, 2005). Hopfield networks, modern Hopfield memories, and the Controlled Dynamics Attractor Transformer bring attractor updates into neural memory and representation learning (Hopfield, 1982; Ramsauer et al., 2021; Zhang et al., 2026a). We connect conditional limits, stable regions, and dominant spectral modes to local Harness processes generated jointly by a model, its tools, and the environment.

The response of a distribution to a changed rule is related to Ruelle's linear-response theory and Mitrophanov's stability analysis for ergodic Markov chains (Ruelle, 2009; Mitrophanov, 2005). RAFA organizes long-horizon language-model planning as a Bayesian adaptive Markov decision process and analyzes regret, while trust-region methods constrain changes between neighboring policies (Liu et al., 2024; Schulman et al., 2015). Work on MDP equivalence and state abstraction studies preservation of behavior or value; Markov-chain lumping studies preservation of transition structure and information rate; and predictive-state representations organize state by future behavior (Givan et al., 2003; Li et al., 2006; Abel et al., 2016; Geiger & Temmel, 2014; Singh et al., 2012). State-aware runtimes and explicit state-dynamics approaches bring related ideas into long-horizon language-model agents (Chen, 2026; Subaharan, 2026). Brier scores and the decompositions of

Murphy and Siegert express probabilistic prediction error through interpretable conditional terms (BRIER, 1950; Murphy, 1973; Siegert, 2017). Building on these foundations, we use operational successors to characterize state fidelity, a conditional attractor distribution to determine visitation weights over a finite response window, and a task readout map to show how these relations enter task outcomes.

B Derivations and Proofs for Conditional-State Dynamics

This appendix fixes the local conditional process used in the main text and derives the finite response, conditional attractor structure, fidelity of state aggregation, and squared predictive cost. A continuous interaction is represented as a finite transient process with absorbing exit classes, retaining the state relations required by subsequent operations and task readout.

B.1 Formalization of the Local Conditional Process

Fix system conditions $c = (h, m, \mathcal{P})$ and let the control-sufficient state be $Z_t = \Phi_c(H_t)$. Let $C = \{1, \dots, n\}$ be the local active-state space and let $\mathcal{B} = \{1, \dots, k\}$ be the set of exit classes. The extended process $\bar{Z}_t$ remains in C until

$$\sigma = \inf\{t \geq 1 : \bar{Z}_t \in \mathcal{B}\},$$

after which it stays in the exit class it reached.

Let U_t denote the admissible operation selected at time t. Operations are drawn from a feasible set $\mathcal{U}$ shared by the fine and coarse processes, and $u = 0$ denotes the reference rule. Define the within-region and first-exit kernels row-wise by

$$[\mathbf{Q}_u]_{ij} = \mathbb{P}(\bar{Z}_{t+1} = j \mid \bar{Z}_t = i, U_t = u),$$
$$[\mathbf{R}_u]_{ib} = \mathbb{P}(\bar{Z}_{t+1} = b \mid \bar{Z}_t = i, U_t = u), \qquad i, j \in C,\ b \in \mathcal{B}.$$

Here $\mathbf{Q}_u \in [0,1]^{n \times n}$ is substochastic and $\mathbf{R}_u \in [0,1]^{n \times k}$ records the exit class, with

$$\mathbf{Q}_u \mathbf{1}_n + \mathbf{R}_u \mathbf{1}_k = \mathbf{1}_n.$$

Let $\boldsymbol{\mu}$ be the initial probability row vector, with $\boldsymbol{\mu}\mathbf{1}_n = 1$. Write $\mathbf{Q} = \mathbf{Q}_0$ and $\mathbf{R} = \mathbf{R}_0$, and set $\mathbf{K}_0 = \mathbf{0}_{n \times k}$. On the joint space of active states and exit classes, define

$$\mathbf{P}_u = \begin{pmatrix} \mathbf{Q}_u & \mathbf{R}_u \\ 0 & \mathbf{I}_k \end{pmatrix}, \qquad \widetilde{\mathbf{P}}_u = \begin{pmatrix} \widetilde{\mathbf{Q}}_u & \widetilde{\mathbf{R}}_u \\ 0 & \mathbf{I}_k \end{pmatrix}, \qquad \mathbf{F} = \begin{pmatrix} \mathbf{1}_n & 0_{n \times k} \\ 0_{k \times 1} & \mathbf{I}_k \end{pmatrix}. \tag{11}$$

Here $\mathbf{I}_k$ is the $k \times k$ identity and the zero blocks have the indicated dimensions. The extended initial distribution is $\bar{\boldsymbol{\mu}} = (\boldsymbol{\mu}, 0)$. The first column of $\mathbf{F}$ records mass that remains in the active region, and the remaining k columns record the exit classes. The quantities S_t, $\boldsymbol{\nu}_t$, and $\mathbf{K}_T$ in the main text are defined by Eq. (1). When the local rule varies with event position, matrix powers below are replaced by time-ordered products, with the same identities holding event by event.

B.2 Proof of the Finite-Response Expansion

Let $\Delta\mathbf{Q} = \mathbf{Q}' - \mathbf{Q}$ and $\Delta\mathbf{R} = \mathbf{R}' - \mathbf{R}$, and define

$$\mathbf{D}_j = \Delta\mathbf{R} + \Delta\mathbf{Q}\mathbf{K}_j.$$

The first-exit recursion,

$$\mathbf{K}_T = \mathbf{R} + \mathbf{Q}\mathbf{K}_{T-1}, \qquad \mathbf{K}'_T = \mathbf{R}' + \mathbf{Q}'\mathbf{K}'_{T-1},$$

gives

$$\begin{aligned} \mathbf{K}'_T - \mathbf{K}_T &= \Delta\mathbf{R} + \mathbf{Q}'(\mathbf{K}'_{T-1} - \mathbf{K}_{T-1}) + \Delta\mathbf{Q}\mathbf{K}_{T-1} \\ &= \mathbf{D}_{T-1} + \mathbf{Q}'(\mathbf{K}'_{T-1} - \mathbf{K}_{T-1}). \end{aligned}$$

Iterating from $\mathbf{K}_0 = \mathbf{K}'_0 = 0$ yields

$$\begin{aligned} \mathbf{K}'_T - \mathbf{K}_T &= \sum_{t=0}^{T-1} (\mathbf{Q}')^t \mathbf{D}_{T-1-t} \\ &= \sum_{t=0}^{T-1} (\mathbf{Q}')^t \big[\Delta\mathbf{R} + \Delta\mathbf{Q}\mathbf{K}_{T-1-t}\big]. \end{aligned}$$

Left multiplication by $\boldsymbol{\mu}$ gives the finite-response expansion in Theorem 1, Eq. (2). A localized modification replaces the powers by time-inhomogeneous products before and after the event, with the corresponding arrival and propagation weights.

B.3 Quasi-Stationary Attractor and Spectral Decomposition

Lemma 1 (Local quasi-stationary attractor)**.** *Let $\mathbf{Q}$ be a primitive finite-dimensional substochastic matrix with $0 < \rho(\mathbf{Q}) < 1$, and write $\rho = \rho(\mathbf{Q})$. There is a unique positive probability row vector $\boldsymbol{\alpha}$ and a positive column vector $\mathbf{v}$ satisfying*

$$\boldsymbol{\alpha}\mathbf{Q} = \rho\boldsymbol{\alpha}, \qquad \mathbf{Q}\mathbf{v} = \rho\mathbf{v}, \qquad \boldsymbol{\alpha}\mathbf{1}_n = \boldsymbol{\alpha}\mathbf{v} = 1.$$

Let $\boldsymbol{\Pi} = \mathbf{v}\boldsymbol{\alpha}$. For $n > 1$, let r_* be the largest modulus among the eigenvalues other than ρ, and set $r_* = 0$ when $n = 1$. For any $s \in (r_*, \rho)$, there is $C_s > 0$ such that

$$\mathbf{Q}^t = \rho^t\boldsymbol{\Pi} + \mathbf{J}_t, \qquad \|\mathbf{J}_t\|_\infty \le C_s s^t, \qquad \boldsymbol{\nu}_t \longrightarrow \boldsymbol{\alpha},$$

whenever the survival mass is positive. Starting from $\boldsymbol{\alpha}$, the mean exit time is

$$\tau_{\mathrm{res}} = (1-\rho)^{-1}.$$

The Perron–Frobenius theorem gives the simple dominant eigenvalue and its positive eigenvectors. Normalize them by $\boldsymbol{\alpha}\mathbf{1}_n = 1$ and $\boldsymbol{\alpha}\mathbf{v} = 1$. Since r_* is the largest modulus of the remaining spectrum, every other eigenvalue has modulus at most $r_* < \rho$. The leading projection satisfies

$$\boldsymbol{\Pi}^2 = \boldsymbol{\Pi}, \qquad \mathbf{Q}\boldsymbol{\Pi} = \boldsymbol{\Pi}\mathbf{Q} = \rho\boldsymbol{\Pi}.$$

With $\mathbf{J}_t = \mathbf{Q}^t - \rho^t\boldsymbol{\Pi}$, the Jordan decomposition gives $\|\mathbf{J}_t\|_\infty \le C_s s^t$ for every $s > r_*$.

For any initial probability row vector $\boldsymbol{\mu}$,

$$\boldsymbol{\mu}\mathbf{Q}^t = \rho^t(\boldsymbol{\mu}\mathbf{v})\boldsymbol{\alpha} + \boldsymbol{\mu}\mathbf{J}_t.$$

Multiplying by $\mathbf{1}_n$ gives

$$S_t = \boldsymbol{\mu}\mathbf{Q}^t\mathbf{1}_n = \rho^t(\boldsymbol{\mu}\mathbf{v}) + \boldsymbol{\mu}\mathbf{J}_t\mathbf{1}_n.$$

Because $\boldsymbol{\mu}\mathbf{v} > 0$ and $s < \rho$, the remainder is negligible relative to the leading term, and hence

$$\boldsymbol{\nu}_t = \frac{\boldsymbol{\mu}\mathbf{Q}^t}{S_t} \longrightarrow \boldsymbol{\alpha}.$$

Starting from $\boldsymbol{\alpha}$, $\mathbb{P}(\sigma > t) = \rho^t$, so summing the tail probabilities gives

$$\mathbb{E}_{\boldsymbol{\alpha}}[\sigma] = \sum_{t\ge 0}\rho^t = (1-\rho)^{-1}.$$

B.4 Slow-Mode Expansion and the Higher-Order Remainder

Assume that the modified kernel $\mathbf{Q}'$ is also primitive and transient. Let $(\rho', \boldsymbol{\alpha}', \mathbf{v}')$ be its leading spectral triple, with

$$\boldsymbol{\alpha}'\mathbf{Q}' = \rho'\boldsymbol{\alpha}', \qquad \mathbf{Q}'\mathbf{v}' = \rho'\mathbf{v}', \qquad \boldsymbol{\alpha}'\mathbf{1}_n = \boldsymbol{\alpha}'\mathbf{v}' = 1.$$

If r'_* is the largest modulus of its remaining spectrum, choose $r'_* < s < \rho'$. Then

$$(\mathbf{Q}')^t = (\rho')^t\mathbf{v}'\boldsymbol{\alpha}' + \mathbf{J}'_t, \qquad \|\mathbf{J}'_t\|_1 \le C_s s^t.$$

Substitution into the finite-response expansion gives

$$\boldsymbol{\mu}(\mathbf{K}'_T - \mathbf{K}_T) = (\boldsymbol{\mu}\mathbf{v}')\sum_{t=0}^{T-1}(\rho')^t\boldsymbol{\alpha}'\mathbf{D}_{T-1-t} + \mathbf{e}_T,$$

$$\mathbf{e}_T = \sum_{t=0}^{T-1}\boldsymbol{\mu}\mathbf{J}'_t\mathbf{D}_{T-1-t}.$$

Using compatible induced 1-norms and $\|\boldsymbol{\mu}\|_1 = 1$,

$$\|\mathbf{e}_T\|_1 \le C_s\sum_{t=0}^{T-1}s^t\|\mathbf{D}_{T-1-t}\|_1.$$

The remainder bound in Eq. (3) separates stable conditional visitation in the leading spectral term from the initial-state and faster-mode effects that shape finite-window behavior.

B.5 Proof of Strong Lumpability under Operational Constraints

Let $\varphi : C \to \widetilde{C}$ be a partition map with indicator matrix $\mathbf{L} \in \{0,1\}^{n\times r}$, where $L_{iz} = \mathbf{1}\{\varphi(i) = z\}$. Define

$$\bar{\mathbf{L}} = \mathrm{diag}(\mathbf{L}, \mathbf{I}_k), \qquad \widetilde{\mathbf{F}} = \begin{pmatrix} \mathbf{1}_r & 0_{r\times k} \\ 0_{k\times 1} & \mathbf{I}_k \end{pmatrix}.$$

Thus $\mathbf{F} = \bar{\mathbf{L}}\widetilde{\mathbf{F}}$. Suppose that, for every $u \in \mathcal{U}$,

$$\mathbf{Q}_u\mathbf{L} = \mathbf{L}\widetilde{\mathbf{Q}}_u, \qquad \mathbf{R}_u = \mathbf{L}\widetilde{\mathbf{R}}_u.$$

Block multiplication gives the intertwining relation

$$\mathbf{P}_u\bar{\mathbf{L}} = \bar{\mathbf{L}}\widetilde{\mathbf{P}}_u. \tag{12}$$

Repeated application yields, for every fixed operation sequence,

$$\mathbf{P}_{u_0}\cdots\mathbf{P}_{u_{T-1}}\bar{\mathbf{L}} = \bar{\mathbf{L}}\widetilde{\mathbf{P}}_{u_0}\cdots\widetilde{\mathbf{P}}_{u_{T-1}}.$$

Consequently, conditional on a coarse-state history and the next operation, every fine state in the same coarse group induces the same next coarse-state distribution. A policy that depends only on the coarse history therefore produces the same joint law of coarse states and operations in the two processes, together with the shared finite-window readout.

For a fixed homogeneous coarse-state policy, the mixed kernels still satisfy $\mathbf{QL} = \mathbf{L}\widetilde{\mathbf{Q}}$. If the fine and coarse kernels are primitive transient kernels and $\boldsymbol{\alpha}$ is the fine conditional attractor, then

$$(\boldsymbol{\alpha}\mathbf{L})\widetilde{\mathbf{Q}} = \boldsymbol{\alpha}\mathbf{Q}\mathbf{L} = \rho(\boldsymbol{\alpha}\mathbf{L}).$$

Uniqueness of the leading eigenvector gives $\widetilde{\rho} = \rho$ and $\widetilde{\boldsymbol{\alpha}} = \boldsymbol{\alpha}\mathbf{L}$, hence $\widetilde{\tau_{\mathrm{res}}} = \tau_{\mathrm{res}}$ and identical finite-window exit observations. This proves Theorem 2.

B.6 Telescoping Expansion of the Representation Defect

When strong lumpability fails, define the representation defect of operation u by

$$\mathbf{E}_u = \mathbf{P}_u\bar{\mathbf{L}} - \bar{\mathbf{L}}\widetilde{\mathbf{P}}_u.$$

Use the notation

$$\prod_{j<t}\mathbf{P}_{u_j} = \mathbf{P}_{u_0}\cdots\mathbf{P}_{u_{t-1}}, \qquad \prod_{j>t}\widetilde{\mathbf{P}}_{u_j} = \widetilde{\mathbf{P}}_{u_{t+1}}\cdots\widetilde{\mathbf{P}}_{u_{T-1}},$$

with empty products equal to the identity of the corresponding dimension. Term-by-term cancellation gives

$$\begin{aligned}&\mathbf{P}_{u_0}\cdots\mathbf{P}_{u_{T-1}}\bar{\mathbf{L}} - \bar{\mathbf{L}}\widetilde{\mathbf{P}}_{u_0}\cdots\widetilde{\mathbf{P}}_{u_{T-1}} \\ &\quad= \sum_{t=0}^{T-1}\left(\prod_{j<t}\mathbf{P}_{u_j}\right)\mathbf{E}_{u_t}\left(\prod_{j>t}\widetilde{\mathbf{P}}_{u_j}\right).\end{aligned} \tag{13}$$

For an extended state $x \in C \cup \mathcal{B}$, let $\mathbf{E}_{u_t}(x,:)$ denote the corresponding row and define

$$\varepsilon_t(x) = \tfrac{1}{2}\|\mathbf{E}_{u_t}(x,:)\|_1, \qquad \boldsymbol{\varepsilon}_t = (\varepsilon_t(x))_{x\in C\cup\mathcal{B}}.$$

Let $\boldsymbol{\xi}_T$ and $\widetilde{\boldsymbol{\xi}}_T$ be the fine and coarse distributions under the common observation. Contractivity of stochastic kernels and the observation map in total variation gives

$$\mathrm{TV}(\boldsymbol{\xi}_T, \widetilde{\boldsymbol{\xi}}_T) \le \min\left\{1, \sum_{t<T}\bar{\boldsymbol{\mu}}\left(\prod_{j<t}\mathbf{P}_{u_j}\right)\boldsymbol{\varepsilon}_t\right\}. \tag{14}$$

The right-hand side is the arrival-mass-weighted sum of local defects. When the intertwining relation holds exactly, every $\mathbf{E}_u$ vanishes, so the coarse process preserves the shared operation paths and finite-window readout.

B.7 Measure Transform and Stability of the Conditional Map

On the positive-survival set

$$\Delta_C^+ = \{\boldsymbol{\nu} \in \Delta_C : \boldsymbol{\nu}\mathbf{Q}\mathbf{1}_n > 0\},$$

the conditional update is

$$\mathcal{F}_{\mathbf{Q}}(\boldsymbol{\nu}) = \frac{\boldsymbol{\nu}\mathbf{Q}}{\boldsymbol{\nu}\mathbf{Q}\mathbf{1}_n}.$$

Let $\mathbf{D}_{\mathbf{v}} = \mathrm{diag}(\mathbf{v})$ and define the Doob transform

$$\mathbf{Q}^{\mathrm{D}} = \rho^{-1}\mathbf{D}_{\mathbf{v}}^{-1}\mathbf{Q}\mathbf{D}_{\mathbf{v}}, \qquad \mathcal{H}_{\mathbf{v}}(\boldsymbol{\nu}) = \frac{\boldsymbol{\nu}\mathbf{D}_{\mathbf{v}}}{\boldsymbol{\nu}\mathbf{v}}.$$

Since $\mathbf{Q}\mathbf{v} = \rho\mathbf{v}$, $\mathbf{Q}^{\mathrm{D}}\mathbf{1}_n = \mathbf{1}_n$. Direct calculation gives

$$\mathcal{H}_{\mathbf{v}}(\mathcal{F}_{\mathbf{Q}}(\boldsymbol{\nu})) = \mathcal{H}_{\mathbf{v}}(\boldsymbol{\nu})\mathbf{Q}^{\mathrm{D}}.$$

Primitivity makes $\mathbf{Q}^{\mathrm{D}}$ an ergodic stochastic kernel whose iterates converge to a unique stationary distribution. Applying the inverse of $\mathcal{H}_{\mathbf{v}}$ on its image gives

$$\mathcal{F}_{\mathbf{Q}}^t(\boldsymbol{\nu}) \longrightarrow \boldsymbol{\alpha}.$$

Thus $\boldsymbol{\alpha}$ is a globally asymptotically stable fixed point of the conditional update on its domain.

B.8 Orthogonal Conditional-Expectation Decomposition of Squared Predictive Cost

Let $Y \in \{1, \dots, d\}$ denote the next event and let $\mathbf{e}_Y \in \mathbb{R}^d$ be its one-hot vector. Let Z^{f} be the full record and $Z^{\mathrm{c}} = \psi(Z^{\mathrm{f}})$ the coarse record, with

$$\mathbf{p}_{\mathrm{f}}(z) = \mathbb{P}(Y = \cdot \mid Z^{\mathrm{f}} = z), \qquad \mathbf{p}_{\mathrm{c}}(z') = \mathbb{P}(Y = \cdot \mid Z^{\mathrm{c}} = z').$$

For a predictor $\mathbf{q}$, define the Brier loss by

$$\mathcal{L}_Z(\mathbf{q}) = \mathbb{E}\|\mathbf{e}_Y - \mathbf{q}(Z)\|_2^2.$$

Because Z^{c} is a function of Z^{f}, $\mathbf{p}_{\mathrm{c}} = \mathbb{E}[\mathbf{p}_{\mathrm{f}} \mid Z^{\mathrm{c}}]$. The orthogonal conditional-expectation decomposition gives

$$\mathcal{L}_{Z^{\mathrm{c}}}(\mathbf{p}_{\mathrm{c}}) - \mathcal{L}_{Z^{\mathrm{f}}}(\mathbf{p}_{\mathrm{f}}) = \mathbb{E}\|\mathbf{p}_{\mathrm{f}}(Z^{\mathrm{f}}) - \mathbf{p}_{\mathrm{c}}(Z^{\mathrm{c}})\|_2^2 =: I_*.$$

For any predictor $\mathbf{q}$ that depends only on Z^{c},

$$\mathcal{L}_{Z^{\mathrm{c}}}(\mathbf{q}) - \mathcal{L}_{Z^{\mathrm{f}}}(\mathbf{p}_{\mathrm{f}}) = I_* + \mathbb{E}\|\mathbf{q}(Z^{\mathrm{c}}) - \mathbf{p}_{\mathrm{c}}(Z^{\mathrm{c}})\|_2^2.$$

Thus I_* is the minimum additional squared loss from merging states.

If the protocol state is $\mathsf{S} \in \mathcal{S}$, define

$$\omega_s = \mathbb{P}(\mathsf{S} = s), \qquad a_s = \mathbb{E}\left[\, \|\mathbf{p}_{\mathrm{f}}(Z^{\mathrm{f}}) - \mathbf{p}_{\mathrm{c}}(Z^{\mathrm{c}})\|_2^2 \,\middle|\, \mathsf{S} = s \right].$$

Applying the same orthogonality relation state by state gives

$$\begin{aligned} G_s &:= \mathbb{E}\big[\mathbf{1}\{\mathsf{S} = s\}\, \big(\|\mathbf{e}_Y - \mathbf{p}_{\mathrm{c}}(Z^{\mathrm{c}})\|_2^2 - \|\mathbf{e}_Y - \mathbf{p}_{\mathrm{f}}(Z^{\mathrm{f}})\|_2^2\big)\big] = \omega_s a_s, \\ G_* &= \sum_s G_s = \sum_s \omega_s a_s = I_*. \end{aligned} \tag{15}$$

B.9 Predictive Cost Weighted by the Conditional Attractor

Fix a local kernel and its conditional attractor $\boldsymbol{\alpha}$. For operation u, write the joint distribution of the next coarse state and exit class from fine state i as

$$\mathbf{p}_u(i) = [\mathbf{Q}_u\mathbf{L} \;\; \mathbf{R}_u]_{i,:} \in [0,1]^{r+k}.$$

For coarse group z, let $\alpha_z^{\mathrm{grp}} = \sum_{i:\varphi(i)=z} \alpha_i$ and, when this quantity is positive, define

$$\overline{\mathbf{p}}_u(z) = \frac{\sum_{i:\varphi(i)=z} \alpha_i \mathbf{p}_u(i)}{\alpha_z^{\mathrm{grp}}}.$$

Viewing $X \sim \boldsymbol{\alpha}$ and $\varphi(X)$ as the fine and coarse records, respectively, Eq. (5) gives

$$I_{\boldsymbol{\alpha},u}(\varphi) = \sum_{i \in C} \alpha_i \|\mathbf{p}_u(i) - \overline{\mathbf{p}}_u(\varphi(i))\|_2^2,$$

which is Eq. (6). The quantity is zero when successor distributions agree within every coarse group. If this holds for every admissible operation, the strong-lumpability condition in Eq. (4) is recovered.

C Analytic Dynamics of Perturbation Propagation and State Aggregation

C.1 When a Local Difference Reaches Task Readout

A difference introduced by state aggregation becomes observable only when the process visits it, propagates it through subsequent transitions, and reaches task readout. Over a finite window, this propagation is shaped jointly by the initial distribution, the transient spectrum, and the readout map. If a reference operation gives the same successor within a coarse group while another operation reads the merged condition and sends the fine states to different exit classes, the latter operation creates a representation defect. If the readout map sends those states to the same task object, the path difference does not enter the evaluation. The two finite constructions below illustrate the propagation profile and the role of operation-relevant state under equal representation capacity.

C.2 A Discrete Double-Well Transient Diffusion Model

Figure 2 uses an explicitly computable discrete double-well transient process to illustrate the finite response in Theorem 1. Let x_i be an equally spaced grid on $[-1.8, 1.8]$ and define

$$V(x) = 2.1(x^2 - 1)^2 + 0.32x.$$

For neighboring grid points i, j, take

$$W_{ij} = 0.43 \min\{1, \exp[-(V(x_j) - V(x_i))]\},$$

set the remaining off-diagonal entries to zero, and let $W_{ii} = 1 - \sum_{j \neq i} W_{ij}$. Define the state-dependent exit rate and exit-class weight by

$$e_i = 0.008 + 0.018(1 + \exp(-4x_i))^{-1}, \qquad \theta_i = (1 + \exp(-5x_i))^{-1},$$

and construct

$$[\mathbf{Q}]_{ij} = (1 - e_i)W_{ij}, \qquad [\mathbf{R}]_{i1} = e_i(1 - \theta_i), \qquad [\mathbf{R}]_{i2} = e_i\theta_i.$$

The reference process concentrates its conditional mass in the left well. A local operation shifts mass toward the right region, after which the reference transition and the state-dependent exit rate jointly determine how that mass decays over the remaining window and returns toward the long-run conditional distribution. The conditional distributions, difference, and total-variation curves in Figure 2 are computed directly from these kernels, placing initial visitation, the local difference, and subsequent propagation in one finite process.

C.3 Defect Diffusion in a Binary Conditional-Kernel Family

To make the route from a representation defect to a future readout explicit, let the fine state consist of a visible region $z \in \{0, 1\}$ and a conditional variable $q \in \{0, 1\}$, with two absorbing exit classes. The coarse representation retains z and merges the two values of q within each region. For a designated operation $u = 1$, consider the kernel family

$$\mathbf{P}_1(\gamma) = (1 - \gamma)\mathbf{P}^{(0)} + \gamma\mathbf{P}^{(1)}, \qquad \gamma \in [0, 1],$$

where $\mathbf{P}^{(0)}$ is insensitive to q within a fixed z and satisfies Eq. (4), whereas $\mathbf{P}^{(1)}$ sends different values of q toward different exit classes. The coarse kernel $\widetilde{\mathbf{P}}_1(\gamma)$ is the marginal transition induced by the same coarse record. With $\bar{\mathbf{L}} = \mathrm{diag}(\mathbf{L}, \mathbf{I}_k)$, the representation defect is

$$\mathbf{E}_1(\gamma) = \mathbf{P}_1(\gamma)\bar{\mathbf{L}} - \bar{\mathbf{L}}\widetilde{\mathbf{P}}_1(\gamma).$$

Let $\mathbf{f}_B$ be the indicator column for exit class B. The contribution of event position t to the terminal readout over a window of length T is

$$w_{t,T} = \bar{\boldsymbol{\mu}}\mathbf{P}_1(\gamma)^t\mathbf{E}_1(\gamma)\widetilde{\mathbf{P}}_1(\gamma)^{T-1-t}\mathbf{f}_B.$$

Summing these terms gives the difference between the fine and coarse observations of class B. At $\gamma = 0$, strong lumpability is restored and the defect terms vanish. As γ increases, the within-group successor difference enters the terminal readout through the forward arrival mass and the backward propagation. Figure 11 shows their position-wise contributions and cumulative sum.

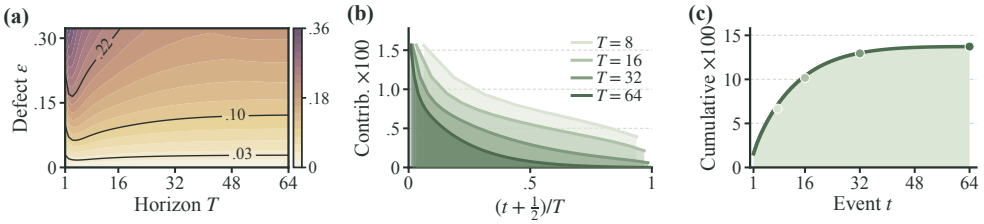

Figure 11: Finite-window propagation of a representation defect. (a) Terminal class-B difference over window length and defect. (b) Event-wise contribution along the normalized window. (c) Cumulative contribution.

C.4 State Fidelity under Operation-Dependent and Independent Variables

Let a fine state contain a visible region z, a dependency variable δ read by the new operation, and a variable j unrelated to that operation. Consider

$$\varphi_{\mathrm{dep}}(z, \delta, j) = (z, \delta), \qquad \varphi_{\mathrm{ind}}(z, \delta, j) = (z, j).$$

The two representations have the same number of coarse labels. The reference operation does not read δ, so both representations satisfy Eq. (4) for that process. Once the new operation reads δ with coupling strength $\lambda \in [0, 1]$, the first representation retains the relevant successor relation, whereas the second merges fine states with different successors.

Figure 4 compares the finite-window value induced by the two representations in this controlled family. Let V_T^{fine} and V_T^{φ} be the probabilities of reaching a designated absorbing class within a window of length T under the fine-state optimal policy and the policy restricted to coarse representation φ, respectively, and define

$$\Delta V = V_T^{\mathrm{fine}} - V_T^{\varphi}.$$

When the dependency variable is retained, the coarse process preserves the successor relation required by each operation and ΔV is zero. When the independent variable is retained instead, increasing λ lets the within-group successor difference accumulate into a nonzero terminal value difference over longer windows. The construction separates label capacity from operation-relevant fidelity, which is determined by successor agreement within each coarse group.

C.5 Readout Invariance under Synchronized Recoding

Let $\mathbf{S}$ be an invertible permutation on the extended state and exit-class space, and let $\mathbf{f}$ be the task-readout column in the original coordinates. If the initial distribution, transition kernels, and readout vector are transformed together as

$$\bar{\boldsymbol{\mu}}^{\dagger} = \bar{\boldsymbol{\mu}}\mathbf{S}, \qquad \mathbf{P}_u^{\dagger} = \mathbf{S}^{-1}\mathbf{P}_u\mathbf{S}, \qquad \mathbf{f}^{\dagger} = \mathbf{S}^{-1}\mathbf{f},$$

then every common operation sequence satisfies

$$(\bar{\boldsymbol{\mu}}\mathbf{S})\mathbf{P}_{u_0}^{\dagger} \cdots \mathbf{P}_{u_{T-1}}^{\dagger}\mathbf{f}^{\dagger} = \bar{\boldsymbol{\mu}}\mathbf{P}_{u_0} \cdots \mathbf{P}_{u_{T-1}}\mathbf{f}.$$

Thus, when task descriptors, code identifiers, and readout rules are recoded together, the task object is unchanged. If a descriptor is permuted while the readout rule retains the old labels, the readout map changes. The SciCode experiment links this algebraic invariance to task evaluation.

D Empirical Tests of Conditional Relations

D.1 Interaction Sources and Event States

We test the conditional-state relations in three kinds of interaction: software repair, scientific programming, and skill execution. The software-repair records come from MiniSWE, OpenHands, and SWE-Agent on the SWE-bench task family (Jimenez et al., 2024); the scientific-programming records come from SciCode subproblems and their evaluation programs (Tian et al., 2024); and the skill-execution records come from Terminus interactions in SkillsBench v1 (Li et al., 2026c). The spectral comparison covers Gemma4-31B (Gemma Team, 2026), Qwen3.5-122B (Qwen Team, 2026a), Qwen3.6-27B (Qwen Team, 2026b), and Qwen3.8-27B (Qwen Team, 2026c), preserving comparisons across model families and interaction organizations.

Table 3: Formal relations and empirical readouts. Correlations, predictive quantities, and score differences use their native scales; r_{corr} is Pearson correlation. "Same direction" indicates agreement in the aggregate trend, and the final row uses common public checkpoints.

Formal relation	Observable form	Interaction setting	Observation
Conditional attractor distribution	Tail decays with $-\log \widehat{\rho}$	Harness run lengths	$r_{\mathrm{corr}} = .88$
Consistency of spectral structure	$\log \widehat{\tau_{\mathrm{res}}}$ remains correlated across partitions	Independent task partitions	$r_{\mathrm{corr}} = .95$
Squared predictive cost, Eq. (5)	$\widehat{G}$ changes in the same direction as $\widehat{I}$	Protocol prediction by task partition	Same direction
Operation-dependent fidelity, Theorem 2	Retention is determined by the readout relation	Equal-capacity binary encodings	Set by phase support
Task readout map Γ_τ	Synchronized recoding preserves the object	SciCode code readout	Object preserved
BasinGuard-controlled path	Formation of the task object determines the evaluation path	Code-workspace tasks with different interaction organization	$+\mathbf{10.87}\%$

Each interaction record is divided at model-call and tool-return boundaries. At event boundary t, the protocol state is

$$s_t \in \{\mathrm{R}, \mathrm{W}, \mathrm{I}, \mathrm{T}\},$$

where the four symbols denote ready, waiting for a tool, interacting, and returning from a tool, respectively. Prediction and transition estimation use task-level splits, so the consecutive records of one task remain within one conditional process.

D.2 Protocol State and the Conditional Relation of the Next Event

Let $Y_t \in \{1, \ldots, d\}$ denote the next event after cut point t, where d is the number of next-event classes, and let $\mathbf{e}_{Y_t} \in \mathbb{R}^d$ be its one-hot vector. Let κ_t denote the call phase and a_t the position within the call. The base and full records are

$$Z_t^{\mathrm{coarse}} = (\kappa_t, a_t), \qquad Z_t^{\mathrm{full}} = (\kappa_t, a_t, s_t).$$

They provide two conditional descriptions of the same next event.

Let $\mathcal{T}_{\mathrm{test}}$ be the held-out task set and N_τ the number of cut points with a labeled next event in task τ. For each held-out task, conditional frequencies are estimated from the remaining tasks. An unseen condition combination receives a half pseudo-count, and a terminal cut point without a next event does not form a prediction sample. Let $\widehat{\mathbf{p}}_{\mathrm{coarse}}$ and $\widehat{\mathbf{p}}_{\mathrm{full}}$ be the probability row vectors obtained from the two records. The task-equal Brier losses are

$$\widehat{\mathcal{L}}_{\mathrm{coarse}} = \frac{1}{|\mathcal{T}_{\mathrm{test}}|} \sum_{\tau \in \mathcal{T}_{\mathrm{test}}} \frac{1}{N_\tau} \sum_{t=1}^{N_\tau} \|\mathbf{e}_{Y_t} - \widehat{\mathbf{p}}_{\mathrm{coarse}}(Z_t^{\mathrm{coarse}})\|_2^2,$$

$$\widehat{\mathcal{L}}_{\mathrm{full}} = \frac{1}{|\mathcal{T}_{\mathrm{test}}|} \sum_{\tau \in \mathcal{T}_{\mathrm{test}}} \frac{1}{N_\tau} \sum_{t=1}^{N_\tau} \|\mathbf{e}_{Y_t} - \widehat{\mathbf{p}}_{\mathrm{full}}(Z_t^{\mathrm{full}})\|_2^2.$$

The empirical conditional-successor difference and Brier improvement are

$$\widehat{I} = \mathbb{E}_{\mathrm{task}} \left\| \widehat{\mathbf{p}}_{\mathrm{full}}(Z^{\mathrm{full}}) - \widehat{\mathbf{p}}_{\mathrm{coarse}}(Z^{\mathrm{coarse}}) \right\|_2^2, \qquad \widehat{G} = \widehat{\mathcal{L}}_{\mathrm{coarse}} - \widehat{\mathcal{L}}_{\mathrm{full}}.$$

Here $\mathbb{E}_{\mathrm{task}}$ first averages within each held-out task and then gives tasks equal weight. $\widehat{I}$ and $\widehat{G}$ estimate the population quantities I_* and G_*. Table 4 reports relative Brier improvement as $100\widehat{G}/\widehat{\mathcal{L}}_{\mathrm{coarse}}$, and reports the Brier quantities as $100\widehat{I}$ and $100\widehat{G}$. Adding protocol state produces a larger predictive-distribution difference together with a larger predictive improvement across the Harnesses, in the direction given by Eq. (5).

Retaining the preceding protocol state improves next-event prediction across the five Harnesses. With entire task blocks held out and a 64-event window, a predictor conditioned on (s_{t-1}, s_t) reduces the mean Brier loss from 0.4687 to 0.1887 relative to conditioning on s_t. The mean gives

Table 4: Protocol-state contribution to next-event prediction. Brier reduction is relative to the base record; $\widehat{I}$ and $\widehat{G}$ are empirical counterparts of Eq. (5), reported in Brier units $\times 100$.

Harness	Brier reduction (%)	$100\widehat{I}$	$100\widehat{G}$
MiniSWE	49.6	0.93	1.03
OpenHands	13.8	0.99	1.13
SWE-Agent	50.6	5.55	5.83
SciCode	52.1	9.63	10.63
Terminus	**62.7**	13.23	14.38

equal weight to each Harness, model, and configuration combination, and the improvement occurs in every held-out task block within these combinations. The preceding event thus carries predictive information relevant to the history representation $\Phi_c(H_t)$.

Protocol information also remains predictive at several semantic resolutions. We compare the ready, waiting, interacting, and returning states with a representation that merges interacting and returning, and with a representation that distinguishes the stages before and after an action. Using the same task holdouts and 64-event windows, we define Brier skill as the loss of the frequency predictor for each representation minus the loss of its state-conditioned predictor. The respective skills are 0.0726, 0.0492, and 0.0301. Each representation improves prediction relative to its own frequency baseline, linking the retained protocol distinctions to observable successor information.

D.3 Binary Encoding under Call-Phase Constraints

The call phase restricts the active state to either $\{\mathrm{R}, \mathrm{T}\}$ or $\{\mathrm{W}, \mathrm{I}\}$. Let

$$\chi : \{\mathrm{R}, \mathrm{W}, \mathrm{I}, \mathrm{T}\} \to \{0, 1\}^4$$

be a four-bit encoding, with its bits ordered as above. Encodings `0011` and `0101` preserve the distinction between the two states within each support set, so there is a decoder d_{dec} satisfying

$$s_t = d_{\mathrm{dec}}(\kappa_t, \chi(s_t)).$$

Encoding `0110` merges the two states within each support set and therefore does not preserve the same successor distinction. Figure 5(a) compares these equal-capacity representations under the same protocol condition; the comparison instantiates the operation-aware fidelity condition in Theorem 2.

D.4 Empirical Decomposition of the State-Conditional Cost

Following Eq. (15), let N_s be the number of observations of state s, let $N_{\mathrm{total}} = \sum_s N_s$ be the total number of observations, and define

$$\widehat{\omega}_s = \frac{N_s}{N_{\mathrm{total}}}, \qquad \widehat{a}_s = \frac{1}{N_s} \sum_{t : s_t = s} \left\| \widehat{\mathbf{p}}_{\mathrm{full}}(Z_t^{\mathrm{full}}) - \widehat{\mathbf{p}}_{\mathrm{coarse}}(Z_t^{\mathrm{coarse}}) \right\|_2^2$$

for states with $N_s > 0$. Figure 5(d) compares $\widehat{\omega}_s \widehat{a}_s$ with the corresponding state-conditional Brier improvement. The visitation weight and conditional-successor difference represent, respectively, how often a state occurs and how much it separates the next-event distribution; their product gives the empirical form of the aggregate decomposition.

D.5 Spectral Characterization of Conditional Attractor Structure

For each fixed model, Harness, and local operation, protocol-state transitions are written as a substochastic kernel on the retained region $C = \{\mathrm{R}, \mathrm{W}\}$, with $\{\mathrm{I}, \mathrm{T}\}$ as exit classes. Each segment begins at its first entry into C, while starts already in C are included; segments that have not exited when observation ends are treated as right-censored. The transition counts estimate the retained-region kernel $\widehat{\mathbf{Q}}$; let $\widehat{\rho}$ be its spectral radius, with empirical residence scale

$$\widehat{\tau_{\mathrm{res}}} = (1 - \widehat{\rho})^{-1}.$$

The theoretical quantity $-\log \widehat{\rho}$ and the tail-decay rate of the Kaplan–Meier estimate use the same local phase and event-time units. Let $S_{\mathrm{KM}}(t)$ denote the Kaplan–Meier estimate of retained-region survival. We fit $\log S_{\mathrm{KM}}(t)$ linearly for $t \geq 8$ and $0.03 < S_{\mathrm{KM}}(t) < 0.97$, and use the negative slope

as the empirical decay rate. Figure 3(c) compares the two decay rates, while Figure 12 compares empirical residence scales under independent task partitions.

For each fixed model, Harness, and operation, the estimated kernel describes transitions within the selected interaction phase. The finite-response identity compares the reference and modified transition rules, while the conditional-attractor interpretation uses the primitive transient condition in Section 2.1.

Independent task-block estimates produce closely grouped spectral quantities within each model–Harness–configuration condition. The median within-combination standard deviation is 0.00243, the median range is 0.00651, and the median coefficient of variation is 0.44% for $\widehat{\rho}$. Estimates from early and late portions of the interactions differ by a median relative change of 1.86%. The spectral decay therefore follows the local interaction structure across task composition and observation interval.

Each evaluation uses a fixed policy packet comprising the predictor, policy identity, release identity, and pulse grammar. Controller decisions retain these identities together with the continuation score and transition count, so the online rule remains tied to the specified policy throughout each interaction.

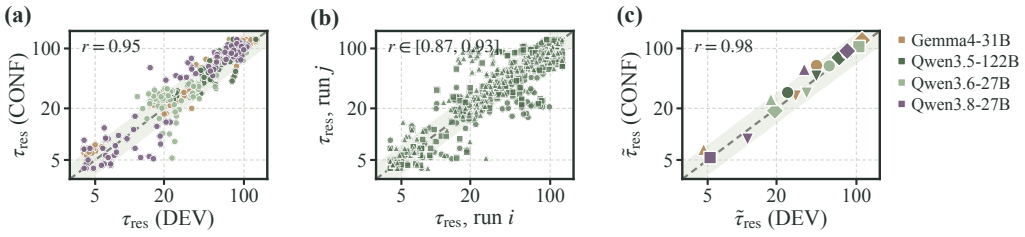

Figure 12: Conditional-attractor consistency across independent estimates. (a) Residence scales for two partitions. (b) Repeated estimates at one coordinate. (c) Model–Harness medians; correlation is on the log scale.

D.6 Code Recoding and Task-Readout Binding

Identity mapping preserves the descriptor–identifier–readout binding; synchronized recoding preserves the task object, while descriptor permutation changes its coordinate.

These controlled recodings therefore produce four readout relations. Synchronized recoding satisfies the algebraic commutation relation in Appendix C.5, leaving the task object and code coverage unchanged. An unsynchronized code identifier sends the readout rule to another program, and the task object changes with it. Figure 6 reports the corresponding readable-code coverage and task score; preserving the binding changes them by 19.84% and 13.79%, respectively, relative to the unsynchronized binding. Each quantity uses its own normalized scale and measures the correspondence between code representation and the physical readout rule.

Readable-code coverage is the proportion of required subproblem code entries that the task protocol resolves to a unique program. The SciCode comparison spans six tasks, three models, six configurations, and three repeats, yielding 324 evaluator input objects. Their 323 nonempty code maps contain 1,053 preserved subproblem entries, which form the coverage denominator. Each coordinated recoding presents the same entries to the evaluator. This common set connects changes in coverage to the association between task descriptors, program identifiers, and evaluation rules.

E Task Readout in Code Tasks with BasinGuard

E.1 The Task-Readout Map in a Code Workspace

GitTaskBench (Ni et al., 2026) places program construction, command-line execution facts, and assertion-based evaluation in one code workspace. We study BasinGuard with three interaction architectures, Pi, Aider, and DeepSeek Harness, and use Qwen3.8-27B and Muse-Glimmer-30B as model conditions (Qwen Team, 2026c; Meta Superintelligence Lab, 2026). The model produces programs and operations, the Harness organizes workspace and tool exchanges, and the task protocol parses the evaluated object from the current public state.

Table 5: BasinGuard interface across the evaluated Harness adapters. Each adapter exposes the public state needed by the controller and retains the host task parser and official evaluator.

Harness	Integration boundary	State and continuation contract	Readout interface
Pi	lifecycle hooks, projected public-tool view, gateway, typed receipts	request index, admitted tokens, atomic groups, errors, tool debt, no-progress, obligation bit; bounded history release, obligation re-anchor, tool recovery	host task parser and official evaluator
Aider	host controller boundary and typed receipts	same frozen predictor and public predicates; host-specific state adapter	host task parser and official evaluator
DeepSeek	host controller boundary and typed receipts	same frozen predictor and public predicates; host-specific state adapter	host task parser and official evaluator

For the task evaluations, thinking mode was disabled for all model calls. The native and BasinGuard paths use the same task inputs, model endpoints, workspace initialization, tool interfaces, and official evaluators. At each public boundary, BasinGuard records the request index, admitted input size, completed atomic tool groups, recent tool outcomes, unmatched tool calls, no-progress state, typed obligation state, continuation predictor, active predicates, and resulting state change. Frozen episode inputs specify the predictor, threshold, earliest eligible request index, cooldown, retained-group count, context-capacity rule, and transition limit. The Pi integration projects the public tool catalog into a typed active view, routes calls through the host gateway, bounds tool-result visibility by the configured context budget, and records typed receipts. We distinguish model requests, tool requests, gateway events, and controller transitions throughout the evaluation trace. Controller transitions occur between provider events, and the host then resumes its ordinary provider lifecycle. The predictor is evaluated from public protocol features alone.

GitTaskBench uses a frozen public-state cascade indexed by task and seed. The cascade keeps the task coordinate, public workspace, and official readout fixed while allowing the host or the plugin controller to determine the next public state. The host and plugin paths therefore begin from the same task boundary and reach the same official readout interface. GAIA2 uses the same paired task assignment and official validation rule; gateway events, provider requests, and tool events remain separate protocol quantities.

The resulting trace distinguishes model requests, input admission, tool events, controller transitions, and task outcomes at their respective protocol levels.

Let $o_\tau(x)$ denote the public output and $w_\tau(x)$ the code, workspace, and execution facts, with value spaces $\mathcal{O}_\tau$ and $\mathcal{W}_\tau$, respectively. Direct-output parsing and structural parsing are

$$\Gamma_\tau^{\mathrm{out}} : \mathcal{O}_\tau \to \mathcal{Y}_\tau \cup \{\bot\}, \qquad \Gamma_\tau^{\mathrm{str}} : \mathcal{W}_\tau \times \mathcal{O}_\tau \to \mathcal{Y}_\tau \cup \{\bot\}.$$

Here $\mathcal{Y}_\tau$ is the space of valid task objects, and $\bot$ denotes a public state in which the evaluated object is not yet uniquely determined. The task-readout map is

$$\Gamma_\tau(x) = \begin{cases} \Gamma_\tau^{\mathrm{out}}(o_\tau(x)), & \Gamma_\tau^{\mathrm{out}}(o_\tau(x)) \in \mathcal{Y}_\tau, \\ \Gamma_\tau^{\mathrm{str}}(w_\tau(x), o_\tau(x)), & \Gamma_\tau^{\mathrm{out}}(o_\tau(x)) = \bot \text{ and } \Gamma_\tau^{\mathrm{str}}(w_\tau(x), o_\tau(x)) \in \mathcal{Y}_\tau, \\ \bot, & \text{otherwise.} \end{cases} \tag{16}$$

Direct-output readout takes priority over structural readout, so the task protocol assigns each public state either a unique object or $\bot$. Let e denote conditional evolution and r denote readout once an object has formed. If $\mathcal{B}_\tau^{\mathrm{task}}$ is the set of boundary classes recognized by the protocol, let $\eta_\tau : \mathcal{B}_\tau^{\mathrm{task}} \to \{\mathsf{e}, \mathsf{r}\}$ map each boundary class to one of these relations. Equation (16) therefore gives a common interface between code, workspace facts, and task evaluation.

Let m and h denote the model and Harness conditions, respectively, and let ℓ index a paired run. Let $x_{\tau,\ell}^0(m,h)$ be the common public checkpoint determined by the task, model, Harness, and run index, containing the task coordinate, current code, workspace snapshot, and public output. From this checkpoint, the native Harness path reaches $x_{\tau,\ell}^{\mathrm{H}}(m,h)$, while the BasinGuard path reaches $x_{\tau,\ell}^{\mathrm{BG}}(m,h)$ according to Eqs. (9) and (8). Both paths use the same readout and scoring maps, so

their per-run scores are

$$J^{\mathrm{H}}_{\tau,\ell}(m,h) = \bar{g}_\tau\big(\Gamma_\tau(x^{\mathrm{H}}_{\tau,\ell}(m,h))\big), \qquad J^{\mathrm{BG}}_{\tau,\ell}(m,h) = \bar{g}_\tau\big(\Gamma_\tau(x^{\mathrm{BG}}_{\tau,\ell}(m,h))\big). \tag{17}$$

Let $\mathcal{R}_\tau$ be the paired run set for task τ. The task-level mean score is

$$\widehat{J}^{\pi}_{\tau}(m,h) = \frac{1}{|\mathcal{R}_\tau|}\sum_{\ell\in\mathcal{R}_\tau} J^{\pi}_{\tau,\ell}(m,h), \qquad \pi \in \{\mathrm{H},\mathrm{BG}\}.$$

The main-text task-score difference first gives tasks equal weight and then gives model–Harness combinations equal weight. The changes shown in the tables and figures are determined by the task objects reached by the two public-state trajectories.

Let $\mathcal{T}$ be the common task set and $\mathcal{C}$ the evaluated model–Harness pairs. The task-level and configuration-level differences are

$$\Delta_{\mathrm{task}}(m,h) = \frac{100}{|\mathcal{T}|}\sum_{\tau\in\mathcal{T}}\Big[\widehat{J}^{\mathrm{BG}}_{\tau}(m,h) - \widehat{J}^{\mathrm{H}}_{\tau}(m,h)\Big], \qquad \Delta_{\mathrm{all}} = \frac{1}{|\mathcal{C}|}\sum_{(m,h)\in\mathcal{C}}\Delta_{\mathrm{task}}(m,h). \tag{18}$$

For the readout-boundary decomposition, let $\mathcal{B}^{\mathrm{task}}_\tau$ be the boundary categories identified by the task protocol, let $b_{\tau,\ell}$ be the category of run ℓ, and let $\eta_\tau : \mathcal{B}^{\mathrm{task}}_\tau \to \{\mathsf{e},\mathsf{r}\}$ assign conditional evolution or readout. With $d_{\tau,\ell}(m,h) = J^{\mathrm{BG}}_{\tau,\ell}(m,h) - J^{\mathrm{H}}_{\tau,\ell}(m,h)$,

$$\Delta_b(m,h) = \frac{100}{|\mathcal{T}|}\sum_{\tau\in\mathcal{T}}\frac{1}{|\mathcal{R}_\tau|}\sum_{\ell\in\mathcal{R}_\tau}\mathbf{1}\{\eta_\tau(b_{\tau,\ell}) = b\}\, d_{\tau,\ell}(m,h), \qquad \Delta_{\mathrm{task}} = \Delta_{\mathsf{e}} + \Delta_{\mathsf{r}}. \tag{19}$$

The two terms assign the score change to the interval before and after the task object becomes readable, respectively.

Table 6: GitTaskBench normalized score difference, BasinGuard minus the native Harness path, aggregated by model and Harness. Marginal and overall rows use equal weighting.

Aggregation	Condition	Δ_{task} (%)
Model	Qwen3.8-27B	+17.15
	Muse-Glimmer-30B	+4.59
Harness	Pi	+10.87
	Aider	+9.06
	DeepSeek	+12.68
♦ Equal-weight overall		**+10.87**

The normalized score difference and the absolute task-success readout describe complementary aspects of the same paired evaluation. Across GitTaskBench, the absolute task-success rate rises from 9.33% on the host path to 22.00% with BasinGuard. For execution accounting, define the normalized execution index $c^\pi = \bar{d}^\pi/\bar{d}^{\mathrm{H}}$, where $\bar{d}^\pi$ is the mean episode duration under path π and the host path provides the unit reference. The index is 0.9541 for GitTaskBench and 0.9942 for GAIA2, so the score gains are accompanied by comparable or lower normalized execution duration.

The task-level differences are positive in every model–Harness condition. Their means and sample standard deviations are $19.57\% \pm 2.17\%$, $8.70\% \pm 0.00\%$, and $23.19\% \pm 2.51\%$ for Qwen3.8-27B with Pi, Aider, and DeepSeek Harness, respectively, and $2.17\% \pm 0.00\%$, $9.42\% \pm 2.51\%$, and $2.17\% \pm 0.00\%$ for Muse-Glimmer-30B. These values use the same task partition and task-equal aggregation as the main table. The paired sign test gives $p = 1.71 \times 10^{-8}$.

E.2 Case Study: From Program Representation to an Evaluated Fact

Program text specifies a computational relation, while the task protocol reads workspace facts to determine the evaluated object. In an Aider Eparse record, the executable solution and required output occupy successive public states. Before the output is formed, the state lies in $\mathcal{X}_\tau \setminus \mathcal{D}_\tau$.

Conditional evolution then produces a protocol-readable output, after which the code and output jointly determine $\Gamma_\tau(x) \in \mathcal{Y}_\tau$. The same readout relation appears in other code-workspace tasks. In the Faker record, the program specifies a data-generation relation and workspace facts then close the pending object. In the Stegano record, a binary artifact is routed through a state-dependent tool organization into an interface readable by the task protocol. The surface operations differ,

Table 7: GAIA2 official task pass rate across model–Harness conditions. Gain is the absolute difference between the BasinGuard and host rates (%).

Model	Harness	Host (%)	BasinGuard (%)	Gain (%)
Qwen3.8-27B	Pi	4.17	6.46	$+\mathbf{2.29}$
	Aider	1.67	2.29	$+\mathbf{0.63}$
	DeepSeek	4.79	6.46	$+\mathbf{1.67}$
Muse-Glimmer-30B	Pi	25.83	28.13	$+\mathbf{2.29}$
	Aider	13.96	14.58	$+\mathbf{0.63}$
	DeepSeek	23.13	24.79	$+\mathbf{1.67}$

but all three cases are connected to the task object by the same map Γ_τ and then evaluated by $\bar{g}_\tau$. Together they make the relation among program representation, environmental facts, and task readout concrete.

E.3 The Readable Set of Task Objects

The corresponding readable set is

$$\mathcal{D}_\tau = \{x \in \mathcal{X}_\tau : \Gamma_\tau(x) \neq \bot\}.$$

When the state belongs to $\mathcal{D}_\tau$, the task protocol computes $\bar{g}_\tau \circ \Gamma_\tau$. For states in the complementary set, BasinGuard uses the public history to form a control-sufficient state and preserves or advances the conditional evolution of the existing protocol until a task object forms. The readable set therefore expresses program state, workspace facts, and task evaluation as one conditional boundary.

E.4 GAIA2 Paired Readout

GAIA2 evaluates objects assembled from public facts across several tool interactions. We apply the same map Γ_τ to states reached from a common public checkpoint under the host Harness and BasinGuard. For a fixed model–Harness condition, let $\mathcal{R}$ be the paired evaluation set and let $v_\ell^{\mathrm{BG}}, v_\ell^{\mathrm{H}} \in \{0, 1\}$ indicate official task success for instance ℓ. The official pass rate is

$$p^\pi = \frac{100}{|\mathcal{R}|} \sum_{\ell \in \mathcal{R}} v_\ell^\pi, \qquad \pi \in \{\mathrm{H}, \mathrm{BG}\},$$

and the gain reported in Table 7 is $p^{\mathrm{BG}} - p^{\mathrm{H}}$. BasinGuard increases the official pass rate under every model–Harness condition. Each cell contains 480 paired instances. Across Pi, Aider, and DeepSeek, the two paths differ by 11, 3, and 8 successful instances, respectively, so the same three gains appear in both model blocks. Their initial and final pass rates remain model-specific. The comparison uses the same task instances and official validation rule for both paths.

The receipt-backed cases show how the boundary assembles a task object from heterogeneous public facts. In the Pi and DeepSeek traces, the BasinGuard path reaches a state containing the required calendar and message facts before official evaluation. The Aider trace starts from an already readable object and carries it through the host evaluator. The common net changes above arise from the aggregate paired counts, while the traces retain model- and Harness-specific paths. Together, the cases instantiate the same relation between conditional evolution, public tool facts, and task readout that governs the code-workspace results.

F Conditional Selection in Self-Evolving Harnesses

F.1 The Relation between Evaluation Feedback and Code Snapshots

This section fixes Qwen3.8-27B and studies how evaluation feedback remains tied to the code snapshot that produced it. FiNER-based self-evolving tasks retain candidate versions with their evaluations.

Classification routing carries the task description and existing feedback into the next code-path decision. Once feedback is bound to snapshot x, the selector state is $(x, \text{receipt})$; Figure 7(a)–(b) reports the corresponding version-selection and classification-routing results. The association between

evaluation feedback and the evaluated object also relates to the discussion of reusable holdouts and repeated evaluation (Dwork et al., 2015).

F.2 Task-Level Candidate Selection

In task-level candidate selection, historical evaluations act on the candidate version that produced them and enter the current version decision. Under this protocol, the equal-weight difference in sub-problem accuracy is 5.45%. The result describes the conditional association between code snapshots and task evaluation, with the score supplied by the same task protocol.

F.3 Evaluation Objects in Classification Tasks

The evaluation object in classification routing changes with task semantics. Reaction-role assignment, legal knowledge, character-level news classification, and intent recognition involve structured labels, domain knowledge, text representation, and user goals, respectively (Schneider et al., 2016; Fei et al., 2024; Zhang et al., 2015; Casanueva et al., 2020). Numeric entities, semantic direction, and fine-grained sentiment in financial text change the input domain and label structure (Loukas et al., 2022; Malo et al., 2014; Demszky et al., 2020); citation intent in scientific writing, textual entailment in scientific question answering, social-media text classification, and multilingual reviews likewise correspond to different evaluation functions (Cohan et al., 2019; Khot et al., 2018; Barbieri et al., 2020; Keung et al., 2020). With the task descriptor and historical feedback jointly used as routing conditions, in-domain accuracy increases by 1.98%, macro-averaged accuracy by 1.41%, and the out-of-distribution task score by 1.22%. These results connect the conditional facts retained by the state representation with the readout relation of each task.

F.4 Public State in Action Selection

TextArena (Guertler et al., 2025) studies action selection under shared public-game conditions. After the policy reads the current public game state, environmental reward and action legality provide the reward and constraint readouts, respectively; Figure 7(c)–(e) reports the corresponding changes. With Qwen3.8-27B fixed, single-player reward, two-player win rate, and action legality increase by 2.1%, 2.97%, and 10.93%, respectively. Each quantity is aggregated by task (Agarwal et al., 2021).