TRACETOCHAIN: Auditing LLM-Agent Reliability with Absorbing Markov Chains

Phat T. Tran-Truong, Xuan-Bach Le(✉)
Faculty of Computer Science and Engineering
Ho Chi Minh City University of Technology (HCMUT), VNU-HCM
Ho Chi Minh City, Vietnam
{phatttt, lexuanbach}@hcmut.edu.vn

***Abstract*—Large language model (LLM) agents are sequential software systems, yet their reliability is reported as scalar benchmark scores—pass@k, pass^k, and the reliability decay curve (RDC)—that neither identify the success-time distribution being estimated nor test whether traces support it, leaving horizon, counterfactual, and metric-reconciliation questions unanswerable. We present TRACETOCHAIN, a reproducible pipeline that fits agent traces to an absorbing discrete-time Markov chain by Laplace-smoothed maximum likelihood, gates every result behind a composite Akaike information criterion and Kolmogorov–Smirnov (KS) certificate, and attaches Dirichlet-posterior and bootstrap intervals. We evaluate it under a strict 50/50 fit/test protocol on seven controlled MAST-style corpora and on $2{,}459$ real episodes: 479 SWE-bench Verified SWE-agent trajectories and $1{,}980$ τ-bench episodes over two models and two domains. On held-out controlled traces the fitted chain reproduces the RDC to a maximum L_∞ error of 0.053 and passes the first-passage KS test on $7/7$ frameworks (minimum $p = 0.83$). On real traces the certified asymptotic reliability recovers the held-out pass rate to within 0.03 on SWE-bench and to a mean absolute error of 0.0075 across four τ-bench corpora, and one fitted chain yields pass@k, pass^k, and closed-form perturbation answers ($\Delta R_\infty = +0.034$ for a fallback tool) with no benchmark rerun. The certificate states exactly which outputs the evidence supports, and its verdict is invariant across 18 clustering configurations of the real corpus. LLM-agent reliability engineering can thus move from scalar scores to an audited model that identifies its own trustworthy outputs.**

***Index Terms*—software reliability, large language models, agent systems, absorbing Markov chains, goodness-of-fit, model auditing, first-passage time, pass@k, reliability decay curve**

I. INTRODUCTION

Large language model (LLM) agents now run as sequential software systems in production: a deployed agent plans, calls tools, observes results, retries, and terminates in success or failure, as in ReAct-style and tool-using designs [1], [2], [3]. Once such an agent sits on a critical path—resolving a support ticket, patching a repository, issuing a refund—the operator inherits the ordinary obligations of dependability and site reliability engineering (SRE) [4], [5]: state a reliability target, quantify the uncertainty around it, and say under what conditions it holds.

Agent evaluation, however, reports reliability as a scalar. Benchmarks supply rich execution traces—ToolBench [6], τ-bench [7], SWE-bench [8], WebArena [9], and AgentBench [10]—but summarize them as pass@k [11], pass^k, or the reliability decay curve (RDC) [12]. Each score is a different summary of when a run succeeds, computed with a different denominator and reported without a fit check. Classical software reliability already has the machinery for this setting, from Markov program models [13] and absorbing chains [14] to reliability-growth models [15], [16], and trace-driven frameworks such as TriCEGAR [17] and ProbGuard [18] learn Markov abstractions from execution data.

Despite this, no current LLM-agent reliability report names the success-time distribution it estimates, tests whether the traces support that distribution, or attaches finite-trace uncertainty to it. The practical cost is concrete. A team that ships a refund agent at pass@$1 = 0.72$ cannot say what reliability a larger step budget buys, what an added fallback tool changes end-to-end, or how an internal pass^5 dashboard relates to an SRE mean-time-between-failures target—without rerunning the benchmark once per question, and still without an interval or a validity check.

In this paper we propose TRACETOCHAIN, a pipeline that turns an agent trace corpus into an *audited* reliability model. Each execution becomes a trajectory in an absorbing discrete-time Markov chain (DTMC) $\hat{M} = (\mathcal{S}_T, \hat{Q}, \hat{R}_\oplus, \hat{R}_\ominus, s_0)$ (Definition 1), with transient states $\mathcal{S}_T$ for intermediate behavior and absorbing states for terminal success and failure. What distinguishes TRACETOCHAIN from prior trace-to-DTMC abstraction is that the fit is not trusted by default: a composite Akaike information criterion (AIC) and Kolmogorov–Smirnov (KS) certificate gates every reported quantity, and Dirichlet-posterior and bootstrap intervals accompany the ones that pass. Horizon reliability, local perturbation, and metric reconciliation then become first-passage queries on one model (Props. 1 and 2, Thm. 3), so pass@k, pass^k, and the RDC stop competing and become projections of a single success-time distribution.

Figure 1 shows the pipeline and the study design that evaluates it: a strict 50/50 fit/test protocol on seven controlled MAST-style corpora, then end-to-end fits on $2{,}459$ real agent episodes from two independent benchmarks. In both settings the certified outputs track their held-out targets closely, and the certificate marks precisely which further outputs its evidence supports—the behavior a reliability audit is supposed to have.

Our contributions are:

(✉) Corresponding Author: Xuan-Bach Le (lexuanbach@hcmut.edu.vn)

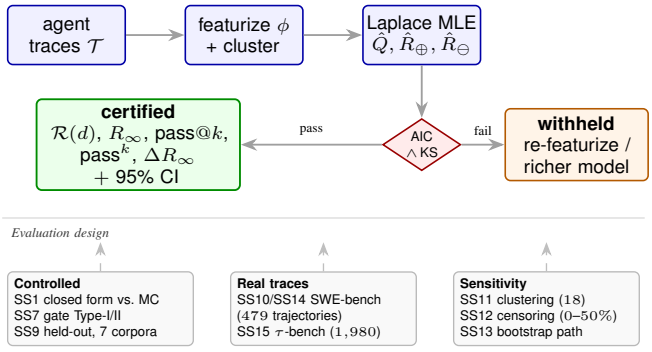

Fig. 1. TRACETOCHAIN pipeline and evaluation design.

(1) *An audited trace-to-chain pipeline.* TRACETOCHAIN (§IV, Algorithms 1–2) maps traces to a fitted absorbing DTMC behind a composite AIC ∧ KS goodness-of-fit certificate, with Dirichlet-posterior and trace-level bootstrap intervals on the chain and every derived quantity (§IV-D).
(2) *Metric unification.* pass@k, pass^k, and the RDC are derived as projections of one success first-passage distribution (Thm. 3), making their denominators explicit and their disagreement resolvable.
(3) *Held-out validation.* On seven controlled MAST-style frameworks (SS9, §VIII-B) the pipeline attains maximum RDC error $L_\infty^{\mathrm{RDC}} = 0.053$ (median 0.048) and held-out KS $p > 0.05$ on $7/7$ frameworks (min $p = 0.83$).
(4) *Real-trace validation on two benchmarks.* Fitting end-to-end to 479 SWE-bench Verified SWE-agent trajectories (SS10–SS14, §IX) and $1{,}980$ τ-bench episodes (SS15, §X) recovers the held-out pass rate to within 0.03 and to a mean of 0.0075 respectively, delivering metric-reconciliation and closed-form perturbation answers on real agents while the audit scopes the remaining outputs.
(5) *Sensitivity analysis.* Clustering, censoring, and bootstrap-path studies (SS11–SS13, §XI) show the verdict is invariant across 18 clustering configurations, and that $\hat{R}_\infty$ stays within 4% of a synthetic ground truth up to 50% censoring.

The rest of this paper is organized as follows. §II positions the work; §III and §IV define and fit the absorbing-chain model; §V and §VI derive the reliability results; §VII–§XI report the controlled, real-trace, and sensitivity studies; and §XII–§XIV discuss limits and implications.

II. RELATED WORK

Prior work provides the ingredients for deployment reliability decisions but not their combination: reliability engineering contributes the vocabulary, agent benchmarks the trace setting, and probabilistic verification the model-based reasoning. The missing piece is an audited fit from LLM-agent traces—a model with uncertainty, diagnostics, and shared semantics for common metrics.

a) Classical SRE.: Classical software reliability gives this paper its vocabulary and analytic tools. The dependability taxonomy of Avižienis et al. separates reliability, availability, safety, and failure semantics [4], while SRE practice [5] turns such concepts into measurable service objectives. State-based modeling is equally central: Cheung [13] used Markov chains for program-level software reliability, Trivedi and Bobbio [19] give a broader engineering treatment, and Kemeny and Snell [14] developed the absorbing-chain framework and fundamental matrix $N = (I - Q)^{-1}$ used by our closed forms. Reliability-growth models add a complementary aggregate view [15], [16]. None of these is by itself a trace-level evaluation method for LLM agents; Proposition 7 connects them by recovering the NHPP form as a rare-failure scaling limit of the fitted agent chain.

b) Agent benchmarks.: Agent benchmarks show why trace-level reliability semantics are needed. ReAct [1], Toolformer [2], and Reflexion [20] established agent trajectories that interleave reasoning, tool use, and revision steps; ToolBench [6] and τ-bench [7] make API choice, tool errors, and tool-agent-user interaction part of the evaluation surface. SWE-bench [8], WebArena [9], ALFWorld [21], and Voyager [22] move evaluation further toward software, web, embodied, and open-ended environments where success is long-horizon and context-dependent. MAST [23] diagnoses multi-agent failure modes, and AgentBench [10] broadens the interactive evaluation space beyond one task family. Together, these benchmarks expose the trace setting; they do not by themselves unify the reported metrics. pass@k [11] originated as a repeated-sampling success metric for code generation, and agent evaluation later adapted it alongside the reliability decay curve (RDC) [12], which tracks how repeated-trial reliability *degrades as task duration grows*. Our d-step reliability $R(d)$ (Definition 2) is the complementary step-budget view: a non-decreasing cumulative success probability indexed by the step budget rather than a duration-indexed decay. We retain the RDC name for continuity and state the modelled quantity explicitly in §V-C.

c) Formal methods for agents.: Formal methods supply the model-based reasoning layer. Probabilistic computation tree logic (PCTL) [24] supports reasoning about probabilistic time and reliability properties, and model-checking texts [25] place such logics in a broader verification framework. Given a probabilistic model, PRISM [26] and Storm [27] can verify PCTL properties. More recent trace-driven frameworks, including TriCEGAR [17] and ProbGuard [18], learn DTMC or Markov decision process (MDP) abstractions from empirical traces. We therefore do not claim trace-to-DTMC abstraction as the main theoretical novelty. Our focus is its reliability-engineering use for LLM agents: TRACETOCHAIN fits an absorbing DTMC with a composite AIC ∧ KS goodness-of-fit certificate and uncertainty quantification, and SS9 (§VIII-B) evaluates held-out recovery on controlled MAST-style corpora using KS distance and L_∞^{RDC} error. The first-passage view explains why pass@k, pass^k, and the RDC are restrictions of a single distribution (Theorem 3), while the Goel–Okumoto NHPP appears as a rare-failure scaling limit of the fitted chain (Proposition 7).

Positioning summary. These traditions are complementary rather than competing: classical reliability models provide semantics, agent benchmarks provide traces, and probabilistic-

verification tools provide model-based queries, but none of them fits and audits a reliability model from LLM-agent traces. TRACETOCHAIN's distinction is the audited combination—traces define the estimate, diagnostics test the abstraction, uncertainty bounds the reported quantities, and first-passage semantics connect horizon reliability, perturbations, and common LLM-agent metrics—which yields a reliability argument from trace data rather than another scalar score.

III. FORMALISM

Deployment questions require probabilities that can be interpreted from traces. The operational question is simple: after an agent has spent several steps reasoning, calling tools, and observing outputs, what is the probability that it reaches success before failure? We represent that "think-act-observe" loop with a fitted absorbing discrete-time Markov chain (DTMC), following classical absorbing-chain and software-reliability models [14], [13]: intermediate behavior becomes transient execution states, the two terminal outcomes are task success $\oplus$ and fatal failure $\ominus$, and reliability becomes a first-passage question—how soon a run reaches the success absorber, and whether failure arrives first. This view also gives benchmark metrics a common semantics, since pass@k [11], pass^k [7], and the step-budget reliability decay curve (RDC) in the sense of §V-C [12] are all restrictions of one success first-passage distribution. We write $\mathcal{S}_T$ for the transient state set: the discrete labels used for intermediate reasoning, tool-use, and observation behavior before a run absorbs into $\oplus$ or $\ominus$.

Later sections estimate the components of $\hat{M}$ from traces and test whether the abstraction is adequate before using it for reliability claims.

Definition 1 (Agent Markov chain)**.** *An* agent Markov chain *(AMC) is a tuple* $\hat{M} = (\mathcal{S}_T, \hat{Q}, \hat{R}_\oplus, \hat{R}_\ominus, s_0)$*, where* $\mathcal{S}_T$ *is the finite set of transient* execution states *obtained from trace featurization (§IV),* $\hat{Q} \in [0,1]^{|\mathcal{S}_T|\times|\mathcal{S}_T|}$ *gives transitions among those states, and* $\hat{R}_\oplus, \hat{R}_\ominus \in [0,1]^{|\mathcal{S}_T|}$ *give absorption probabilities into the task-complete state* $\oplus$ *and the fatal-failure state* $\ominus$*. Mass is conserved:* $\hat{Q}\mathbf{1} + \hat{R}_\oplus + \hat{R}_\ominus = \mathbf{1}$*. Finally,* $s_0 \in \mathcal{S}_T$ *is the initial state, or more generally an initial distribution* π_0*.*

Definition 2 (Reliability)**.** *The* d*-step reliability is* $\mathcal{R}(d; \hat{M}) = \Pr[\tau_\oplus \le d \mid X_0 = s_0]$*, where* $\tau_\oplus$ *is the first time the chain reaches the success absorber, and* $R_\infty = \lim_{d\to\infty} \mathcal{R}(d; \hat{M})$*. The Kemeny–Snell fundamental matrix [14] is* $N = (I - \hat{Q})^{-1}$*; it counts expected visits to transient states before absorption and yields the closed form in Prop. 1.*

Benchmark metrics. For k repeated runs of a task, pass@k is the probability that *at least one* run succeeds and pass^k that *all* k succeed; under the fitted chain and i.i.d. trials (A4) both are restrictions of the same success first-passage distribution as the RDC (Theorem 3).

This first-passage view also connects the fitted chain to classical reliability growth. A central software-reliability model is the non-homogeneous Poisson process (NHPP), a counting-process model for failure discovery over time, especially the Goel–Okumoto model [15]; Musa–Okumoto gives a related formulation [16]. Under rare-failure scaling, the fitted agent chain recovers this NHPP form as a limit (Proposition 7), which explains when classical reliability-growth estimators are meaningful for agent traces.

TABLE I
NOTATION. THE FITTED CHAIN $\hat{M}$ AND ITS DIAGNOSTICS ARE ESTIMATED FROM TRACES (§IV).

Symbol	Meaning
$\mathcal{S}_T$, m, ϕ	transient states, $m = \lvert\mathcal{S}_T\rvert$, step featurizer
$\hat{Q}$	$m\times m$ transient transition matrix
$\hat{R}_\oplus, \hat{R}_\ominus$	exit vectors to success $\oplus$ / failure $\ominus$
$N{=}(I{-}\hat{Q})^{-1}$	fundamental matrix (visits pre-absorption)
s_0, $\tau_\oplus$	start state; first-passage time to success
$\mathcal{R}(d), R_\infty$	reliability $\Pr[\tau_\oplus \le d]$ and its limit
$\rho(\hat{Q})$	spectral radius of $\hat{Q}$ (A1)
pass@k, pass^k	any-of-k / all-of-k success probabilities
Δ_{AIC}	AIC(1st)$-$AIC(2nd) Markov-order statistic

Model assumptions. Transience gives eventual absorption and the fundamental matrix; invertibility is the algebraic condition for the closed forms; the initial condition fixes where traces start; and i.i.d. replications are required when interpreting pass@k and pass^k. The first-order Markov property is not taken on faith: the tests in §IV-B and §IV-C report when the corpus does not support the fitted DTMC.

Assumption 1 (Transience)**.** $\rho(\hat{Q}) < 1$.

Assumption 2 (Invertibility)**.** $(I - \hat{Q})$ *is invertible.*

Assumption 3 (Initial condition)**.** s_0 *is deterministic (extensions replace* $e_{s_0}^\top$ *by* $\pi_0^\top$*).*

Assumption 4 (I.i.d. replications)**.** *For pass*^{k} *and pass@*k*, trials are i.i.d. unless noted.*

Definition 3 (Local perturbation family)**.** *For sensitivity analysis, let* $Q(\varepsilon) = Q_0 + \varepsilon\Delta$ *for* $0 \le \varepsilon \le \varepsilon_{\max}$*, where rows remain substochastic and* $\rho(Q(\varepsilon)) < 1$*. The success-exit vector* $R_{\oplus,0}$ *is held fixed, and any removed transient mass is assigned to the failure absorber so that* $Q(\varepsilon)\mathbf{1} + R_{\oplus,0} + R_\ominus(\varepsilon) = \mathbf{1}$*. Operationally, this represents a local design change, such as adding a fallback tool, without re-running the full benchmark.*

Remark. The fitted absorbing DTMC is not intended to recover every hidden state of the agent or its environment. It is a compact reliability model estimated from observable traces, and its outputs are meaningful only when the state construction and goodness-of-fit checks support the first-order absorbing-DTMC approximation.

IV. STATE CONSTRUCTION FROM TRACES

The formalism is useful only if traces can produce the fitted chain reproducibly and with checks that expose when the abstraction fails. A fitted Markov model is an estimate plus evidence: the trace representation, order test, first-passage fit, and uncertainty analysis must all support its use. Figure 1

summarizes the workflow; Algorithm 1 implements the construction pipeline and Algorithm 2 the goodness-of-fit (GoF) protocol that SS7 (§VII-B) checks on controlled corpora.

A. Featurization and Discretization

The first step is to turn heterogeneous trace events into countable states: reasoning steps, tool calls, and observations are featurized and then clustered into labels such as "plan" or "tool result." Let $\mathcal{T} = \{\tau^{(i)}\}_{i=1}^{n}$ be the trace corpus. A run is $\tau^{(i)} = (\sigma_1^{(i)}, \sigma_2^{(i)}, \ldots, \sigma_{L_i}^{(i)}, y^{(i)})$, where each $\sigma_t^{(i)}$ is one observed step and $y^{(i)} \in \{\oplus, \ominus\}$ is the terminal outcome. A featurizer $\phi : \sigma \mapsto \mathbb{R}^p$ maps each step to a vector representation, either:

- *rule-based* (tool type, retry flag, error code, used in our simulation study SS7 and MAST experiments), or
- *learned* (a pretrained sentence encoder applied to the agent's chain-of-thought and interchangeable in Algorithm 1).

We cluster the feature vectors $\{\phi(\sigma_t^{(i)})\}$ using agglomerative Ward linkage [28]. The number of clusters $m \in [k_{\min}, k_{\max}]$ is selected by silhouette score [29]; we use $[3, 6]$ on the controlled corpora of SS9 and $[2, 10]$ on the real corpora, with SS11 varying $k_{\max}$ explicitly. Each step is assigned to the selected cluster, yielding $s_1^{(i)}, \ldots, s_{L_i}^{(i)} \in \{1, \ldots, m\}$ followed by the terminal outcome.

B. Transition Estimation and Order Selection

Estimating the fitted chain is then a counting problem with a direct operational reading: a row of $\hat{Q}$ says where the agent goes next if it has not terminated, and the matching entries of $\hat{R}_\oplus$ and $\hat{R}_\ominus$ say how often that state exits to success or fatal failure. We fit these quantities by Laplace-smoothed maximum-likelihood estimation (MLE) with $\alpha = 1$, so $\hat{Q}_{ij} = (c_{ij} + \alpha)/(c_i + \alpha(m+2))$ where c_{ij} is the count of transitions from state i to state j, $c_{i,\oplus}$ and $c_{i,\ominus}$ count terminal exits from state i, and $c_i = \sum_j c_{ij} + c_{i,\oplus} + c_{i,\ominus}$. The same denominator normalizes the success and failure exits. Smoothing prevents sparse rows from assigning zero probability to unobserved but plausible exits.

We also test whether first-order memory is sufficient. The comparison uses a first-vs-second-order Markov Akaike information criterion (AIC) test [30]: $\Delta_{\mathrm{AIC}} = \mathrm{AIC}_1 - \mathrm{AIC}_2 = -2(\ell_1 - \ell_2) + 2(k_1 - k_2)$, where ℓ_o and k_o are the log-likelihood and free-parameter count under order o. The first-order model is preferred iff $\Delta_{\mathrm{AIC}} < 0$. Simulation study SS7 (§VII-B) shows that this test has high power against second-order ground truth.

a) Time-homogeneity as a testable assumption.: When we pool transition counts, we assume a *time-homogeneous* kernel $\Pr[s_{t+1} = j \mid s_t = i]$ that does not vary with t. The composite AIC $\wedge$ KS protocol (Algorithm 2) tests whether that simplification is defensible. AIC checks whether recent history still matters after conditioning on the current state, while the Kolmogorov–Smirnov (KS) first-passage test can reveal drift in exit hazards. If either test fails, the corpus should be segmented before counts are pooled.

Algorithm 1: TRACETOCHAIN: construct and audit an absorbing DTMC from agent execution traces.

Input: Traces $\mathcal{T}$, featurizer ϕ, cluster range $[k_{\min}, k_{\max}]$, Laplace α.
Output: $\hat{Q}, \hat{R}_\oplus, \hat{R}_\ominus$, labels π, AIC order verdict.

$X \leftarrow [\phi(\sigma_t^{(i)})]_{i,t}$ *// featurize*
for $k = k_{\min}$ **to** $k_{\max}$ **do**
  $L_k \leftarrow \mathrm{AgglomerativeWard}(X, k)$; $\mathrm{sil}_k \leftarrow \mathrm{silhouette}(X, L_k)$
end
$m^\star \leftarrow \arg\max_k \mathrm{sil}_k$; $\pi \leftarrow L_{m^\star}$
$c_{s_t, s_{t+1}} \mathrel{+}= 1$ for each transient transition; $c_{s_{L_i}, y^{(i)}} \mathrel{+}= 1$ for each terminal exit
$\hat{Q}_{ij} \leftarrow (c_{ij} + \alpha)/(c_i + \alpha(m^\star + 2))$; same normalization for $\hat{R}_{\oplus,i}, \hat{R}_{\ominus,i}$
Compute ℓ_1, ℓ_2, k_1, k_2; $\Delta_{\mathrm{AIC}} \leftarrow -2(\ell_1 - \ell_2) + 2(k_1 - k_2)$
return $(\hat{Q}, \hat{R}_\oplus, \hat{R}_\ominus, \pi)$*; first-order iff* $\Delta_{\mathrm{AIC}} < 0$

b) Variable horizons, early termination, and censoring.: Runs can have different lengths and may end in success ($\oplus$), failure ($\ominus$), or right-censoring. Algorithm 1 counts each observed transient transition once. Completed runs add one absorber count, while censored runs contribute only the transitions observed before censoring. This keeps $\hat{Q}$ tied to observed movement among transient states, but it can make the exit estimates $\hat{R}_\oplus$ and $\hat{R}_\ominus$ conservative when censoring is substantial. In §VIII-A we observe censoring below 10%, and SS12 (§XI) probes far heavier regimes.

C. Goodness-of-Fit Protocol

Fitting helps only if the abstraction matches the traces. A rejection is an engineering result, not a nuisance: it says the current state labels should not be used for reliability queries without segmentation, new features, or a richer model. We use two checks on $\hat{M}$ that address different failure modes:

Layer 1 (KS on first-passage). We compare the model success first-passage-time (FPT) distribution $F^{\hat{M}}_{\tau_\oplus \mid \tau_\oplus < \tau_\ominus}$, conditional on eventual success, with the empirical distribution of success FPTs among successful traces, and retain the null iff $p_{\mathrm{KS}} > 0.05$. The reference distribution is drawn from $\hat{M}$ as N seeded absorption times ($N = 8{,}000$ in SS9, $20{,}000$ on the real corpora), so this is a genuine *two-sample* KS test [31], not a one-sample test against a closed-form CDF: stochastic, but seed-reproducible.

Layer 2 (AIC). The AIC layer asks whether the first-order chain is adequate compared with a second-order alternative. We reject the first-order model if $\Delta_{\mathrm{AIC}} > 0$.

Composite rule. The chain is accepted iff *both* $p_{\mathrm{KS}} > 0.05$ *and* $\Delta_{\mathrm{AIC}} < 0$. This rule catches second-order dependence that the FPT-marginal KS test can miss (see §VII-B). If either layer fails, later reliability quantities should be treated as unsupported by the current trace abstraction. Given ϕ, the cluster range, α, and a lowest-k silhouette tie-breaking rule, Algorithm 1 is deterministic in the trace corpus, and Algorithm 2 is deterministic given its reference-sample seed; all seeds are fixed in the artifact.

Algorithm 2: Composite KS/AIC goodness-of-fit test for the agent-DTMC assumption.

Input: Traces $\mathcal{T}$, fitted chain $\hat{M}$, threshold $\alpha_{\mathrm{KS}} = 0.05$.
Output: ACCEPT / REJECT, $(p_{\mathrm{KS}}, \Delta_{\mathrm{AIC}})$.

Draw N seeded absorption times from $\hat{M} \to$ reference $\tilde{F}^{\hat{M}}_{\tau_\oplus \mid \tau_\oplus < \tau_\ominus}$
Collect empirical FPTs from $\mathcal{T}_\oplus$ (success traces)
$p_{\mathrm{KS}} \leftarrow$ two-sample KS($\hat{F}$, $\tilde{F}^{\hat{M}}$)
$\Delta_{\mathrm{AIC}} \leftarrow$ (Algorithm 1, line 8)
if $p_{\mathrm{KS}} > \alpha_{\mathrm{KS}}$ ***and*** $\Delta_{\mathrm{AIC}} < 0$ **then**
  return ACCEPT, $(p_{\mathrm{KS}}, \Delta_{\mathrm{AIC}})$
else
  return REJECT, $(p_{\mathrm{KS}}, \Delta_{\mathrm{AIC}})$
end

D. Uncertainty Quantification on $\hat{Q}$, $\hat{R}_\oplus$, $\hat{R}_\ominus$

Algorithm 1 returns point estimates, but a small corpus may estimate a row of $\hat{Q}$ or an exit probability poorly, and that uncertainty should be visible before an operator interprets $R(d)$, R_∞, or a perturbation result. We therefore attach intervals to $\hat{Q}$, $\hat{R}_\oplus$, and $\hat{R}_\ominus$ and propagate them to the quantities of §V–§VI.

a) (i) Closed-form Dirichlet posterior.: The Laplace-smoothed MLE is the posterior mean under a symmetric Dirichlet(α) prior over $\{s_1, \ldots, s_m, \oplus, \ominus\}$. The row-$i$ posterior is $\mathrm{Dir}(c_{i,\cdot} + \alpha\mathbf{1})$ and the marginal for any single entry $X_{i,j}$ (including absorber columns) is $\mathrm{Beta}(c_{i,j} + \alpha,\ c_i + \alpha(m + 2) - c_{i,j} - \alpha)$, where $c_i = \sum_{j'} c_{i,j'}$ includes both transient and absorber outcomes. Equal-tailed $(1 - \alpha_{\mathrm{CI}})$ credible intervals (CIs) are the corresponding Beta quantiles; we use $\alpha_{\mathrm{CI}} = 0.05$ throughout.

b) (ii) Non-parametric trace-level bootstrap.: To quantify trace-sampling variability we resample traces with replacement $B = 200$ times, assign each step to its nearest target centroid to keep labels aligned, and recompute the MLE. The implemented full re-clustering path (`fast=False`) gives comparable but wider intervals.

c) Empirical ranges.: Table II reports representative MAST-derived state taxonomies from SS8; each cluster is named by its dominant rule-based feature (§IV-A) with that feature's within-cluster share in parentheses, and the posterior intervals show which success exits are well estimated or sparse, exposing state-level variation hidden by aggregate pass/fail rates. Across the seven frameworks the two uncertainty routes agree closely on entries of $\hat{Q}$: median posterior CI widths span 0.108–0.116 and median bootstrap widths 0.084–0.103, while maxima reaching 0.34 mark the sparse rows. All downstream $\mathcal{R}(d)$, mean time to absorption (MTTA $= e_{s_0}^\top N\mathbf{1}$), and pass^k quantities inherit these CIs by propagation (Lipschitz-continuous in $(\hat{Q}, \hat{R}_\oplus)$ by Proposition 2).

The output is the audited model used for reliability analysis: a fitted chain plus uncertainty and fit diagnostics. If either layer rejects, the corpus should be segmented, re-featurized, or treated as outside the absorbing-DTMC approximation; when both pass, the fitted chain can be used for reliability queries with the reported intervals.

TABLE II
MAST-DERIVED STATE TAXONOMY WITH 95% POSTERIOR SUCCESS-EXIT INTERVALS.

Framework	Cluster	$\hat{R}_{\oplus,i}$	95% post. CI	$\hat{R}_{\ominus,i}$
react [1]	tool_call (100%)	0.064	[0.039,0.094]	0.067
	plan (100%)	0.076	[0.048,0.109]	0.072
	retry (100%)	0.034	[0.004,0.092]	0.085
	reflect (100%)	0.081	[0.038,0.138]	0.081
	error_parse (100%)	0.082	[0.028,0.162]	0.082
	wait (100%)	0.057	[0.007,0.153]	0.057
reflexion [20]	plan (100%)	0.121	[0.074,0.178]	0.027
	error_parse (100%)	0.083	[0.041,0.138]	0.033
	retry (100%)	0.054	[0.020,0.103]	0.045
	reflect (100%)	0.104	[0.056,0.166]	0.043
	tool_call (100%)	0.097	[0.066,0.133]	0.064
	wait (100%)	0.128	[0.049,0.236]	0.064

V. ANALYTIC TOOLS

Once a corpus has produced an accepted fitted chain, the question is what that chain answers without another benchmark run. Three absorbing-chain tools address the deployment questions of §I: reliability at a horizon, sensitivity to a local change, and compatibility among common benchmark metrics.

A. Closed-Form Reliability

The first deployment question is reliability under a step budget: starting at s_0, what is the chance of reaching $\oplus$ by step d? In the running loop, this asks whether the agent reaches a correct final answer within d reasoning/tool/observation steps.

Proposition 1 (Closed-form reliability; Kemeny–Snell [14])**.** *Under Assumptions 1–3, for every $d \in \mathbb{N}_{\geq 0}$,*

$$R(d) = e_{s_0}^\top (I - Q^d)\, N\, R_\oplus, \qquad N = (I - Q)^{-1}, \quad (1)$$

and $R_\infty = \lim_{d\to\infty} R(d) = e_{s_0}^\top N R_\oplus$.

Proof idea. Summing all paths that spend $t = 0, \ldots, d-1$ steps in transient states and then exit to success gives $R(d) = e_{s_0}^\top \sum_{t=0}^{d-1} Q^t R_\oplus$; the identity $\sum_{t=0}^{d-1} Q^t = (I - Q^d)(I - Q)^{-1}$ yields (1), and transience gives $Q^d \to 0$. Since N_{ij} is the expected number of visits to state j before absorption, the closed form separates two effects an operator can inspect: how often the agent visits each state, and how likely success is after those visits.

B. Perturbation Sensitivity

The second question is counterfactual: if a local transition changes, how much can end-to-end reliability move? A new fallback tool may change one row of Q by redirecting failed tool calls back to planning, and the fitted chain gives a first-order sensitivity calculation before any benchmark rerun.

Proposition 2 (Perturbation sensitivity; Stewart [32])**.** *Let $Q(\varepsilon) = Q_0 + \varepsilon\Delta$ be the perturbation family from Definition 3. Write $N_0 = (I - Q_0)^{-1}$ and let $R_{\oplus,0}$ be the fixed success-exit vector. Assume $\varepsilon_{\max}\|N_0\|_\infty\|\Delta\|_\infty < 1$. For every $0 \leq \varepsilon \leq \varepsilon_{\max}$,*

$$\big|R_\infty(\varepsilon) - R_\infty(0)\big| \leq \varepsilon\|N_0\|_\infty^2\|\Delta\|_\infty\|R_{\oplus,0}\|_\infty + C\varepsilon^2, \quad (2)$$

with $C = \|N_0\|_\infty^3 \|\Delta\|_\infty^2 \|R_{\oplus,0}\|_\infty / (1 - \varepsilon_{\max}\|N_0\|_\infty\|\Delta\|_\infty)$. *Moreover,*

$$R_\infty(\varepsilon) - R_\infty(0) = \varepsilon e_{s_0}^\top N_0 \Delta N_0 R_{\oplus,0} + O(\varepsilon^2). \tag{3}$$

Proof idea. Expand $(I - Q(\varepsilon))^{-1}$ with the Neumann series around Q_0; the smallness condition bounds the remainder by the displayed constant. In the first-order term $N_0 \Delta N_0$, the left factor counts how often execution reaches the changed row and the right factor measures success value after the perturbed transition, so a large $\|N_0\|_\infty$ warns that small local changes can have large global effects. The bound is best read as a screening calculation that identifies local changes meriting a new benchmark run. As stated it applies to perturbations of Q with $R_\oplus$ fixed; changes to success exits require the corresponding linear term for the exit vector, and the real-data fallback-tool result of §IX-B is of that second kind, so we evaluate it exactly from Proposition 1 rather than through this bound.

C. Metric Unification

The third question is semantic. pass@k, pass^k, and the RDC often appear to be competing summaries, but under the fitted chain they are different views of one first-passage distribution: the disagreement is about which projection to report, not about the underlying reliability event. We keep the benchmark name RDC, but the quantity modeled here is the cumulative success probability by horizon d.

Theorem 3 (Metric unification)**.** *Let* $R_\infty = e_{s_0}^\top N R_\oplus$. *Under the i.i.d. replication assumption A4,*

$$\text{pass}^k = R_\infty^k, \tag{4}$$

$$\text{pass@}k = 1 - (1 - R_\infty)^k, \tag{5}$$

$$\text{RDC}(d) = R(d) = e_{s_0}^\top (I - Q^d) N R_\oplus. \tag{6}$$

Thus the three metrics are restrictions or transformations of the same first-passage distribution.

Proof idea. pass^k is the probability that all k independent trials succeed, pass@k is one minus the probability that all k fail, and the RDC is the finite-horizon reliability of Proposition 1.

Repeated agent runs can, however, share latent conditions such as a common prompt template, cache state, or task difficulty. The next result makes that caveat measurable: it shows how shared variation changes the two repeated-trial metrics even when marginal reliability remains R_∞.

Theorem 4 (Correlated-trial inequalities)**.** *Suppose the* k *trials share a latent state* ξ *with* $\Pr[X_i = 1 \mid \xi] = p(\xi)$ *and* $\mathbb{E}_\xi[p(\xi)] = R_\infty$, *and are independent conditional on* ξ. *Then*

$$\text{pass}^k = \mathbb{E}_\xi[p(\xi)^k] \geq R_\infty^k, \tag{7}$$

$$\text{pass@}k = 1 - \mathbb{E}_\xi[(1 - p(\xi))^k] \leq 1 - (1 - R_\infty)^k. \tag{8}$$

For $k \geq 2$, *equality holds iff* $p(\xi)$ *is almost surely constant.*

Proof idea. Condition on the latent variable and apply Jensen's inequality to the convex functions p^k and $(1-p)^k$.

Corollary 5 (Diagnostic)**.** *The gap* $\widehat{\text{pass}^k} - \hat{R}_\infty^k$ *is a lightweight diagnostic for hidden cross-trial correlation. Large positive gaps indicate that repeated trials are not behaving as i.i.d. samples from one fixed chain.*

Hidden heterogeneity thus raises all-success estimates and lowers at-least-one-success estimates relative to the i.i.d. formulas, so a large gap should trigger inspection before repeated samples are treated as independent trials: an assumption behind pass@k and pass^k becomes a diagnostic rather than a hidden convention.

VI. Extensions

Three extensions help interpret those answers: why an RDC may saturate, when a classical reliability-growth curve emerges, and how many steps are needed before $R(d)$ approaches R_∞.

A. RDC Shape

An RDC is more useful when its shape has an explanation: a concave curve suggests early steps resolve the easier cases while later ones add less, and a convex window suggests a barrier such as waiting on a tool response. The following condition captures the common early-gains/saturation pattern.

Assumption 5 (Stochastic monotonicity of Q)**.** *Fix an order* $\preceq$ *on the transient states. The matrix* Q *is* stochastically monotone *if* Qf *is non-increasing whenever* $f : \mathcal{S}_T \to \mathbb{R}_{\geq 0}$ *is; equivalently, later rows of* Q *place stochastically more mass on later states in the order [33], [34], [35].*

Proposition 6 (RDC shape; Keilson–Kester [35], Karlin–Rinott [36])**.** *Let* $v = e_{s_0}^\top$ *and* $w = R_\oplus$. *For* $R(d) = v(I - Q^d)NR_\oplus$,

$$\Delta R(d) = R(d+1) - R(d) = vQ^d w \geq 0, \tag{9}$$

and, for $d \geq 1$,

$$\Delta^2 R(d) = R(d+1) - 2R(d) + R(d-1) = -vQ^{d-1}(I - Q)w. \tag{10}$$

If $Qw \leq w$ *componentwise, then* $R(d)$ *is globally concave. Assumption 5 together with* w *non-increasing in the state order is the structural regime in which that dominance is expected; the concavity conclusion itself needs only* $Qw \leq w$.

Proof idea. The first difference is the probability of exiting to success exactly after d transient steps, so reliability never decreases with more steps; since $vQ^{d-1} \geq 0$, (10) makes $Qw \leq w$ sufficient for a nonpositive second difference. A convex empirical window instead signals an early barrier.

B. NHPP Rare-Failure Limit

The absorbing-chain view also explains when aggregate reliability-growth curves emerge from step-level behavior—not as a separate modeling story but as the rare-failure limiting case in which a one-state chain reduces to Goel–Okumoto.

Proposition 7 (NHPP rare-failure limit; Goel–Okumoto [15])**.** *Consider a sequence of one-transient-state agent chains with per-step success probability* μ_n, *per-step failure probability*

ε_n, *and* $Q_n = 1 - \mu_n - \varepsilon_n$. *Suppose* $\mu_n \to 0$, $\varepsilon_n \to 0$, $d_n\mu_n \to \Lambda \in (0, \infty)$, *and* $\varepsilon_n/\mu_n \to 0$. *For any fixed* $c > 0$ *and* $d = cd_n$, *the cumulative first-passage CDF converges to*

$$R_n(d) \to 1 - e^{-c\Lambda}, \tag{11}$$

the Goel–Okumoto NHPP mean-value form $m(t) = a(1-e^{-bt})$ *with normalized* $a = 1$.

Proof idea. For the one-state chain, $R_n(d) = \frac{\mu_n}{\mu_n+\varepsilon_n}\big(1-(1-\mu_n-\varepsilon_n)^d\big)$. Under rare-failure scaling the prefactor tends to one and the geometric term tends to $e^{-c\Lambda}$. Outside this low-error regime, the exact closed form of Proposition 1 should be used.

C. Approach-Rate Bound

Even when R_∞ is known, a practitioner still needs a step budget. The next result converts spectral decay of Q into a conservative horizon rule: how many more agent steps are needed before the remaining reliability gap is at most δ?

Proposition 8 (Approach-rate bound; Horn–Johnson [37], Stewart [32], Levin–Peres [38])**.** *Let* $\rho = \rho(Q) < 1$ *and* $N = (I-Q)^{-1}$. *If* $Q = V\Lambda V^{-1}$ *is diagonalizable, then* $R_\infty - R(d) \le \kappa_\infty(V)\rho^d\|NR_\oplus\|_\infty$ *with* $\kappa_\infty(V) = \|V\|_\infty\|V^{-1}\|_\infty$, *so a sufficient horizon for error at most* δ *is*

$$d^\star_{\mathrm{diag}}(\delta) = \max\left\{0, \left\lceil \frac{\log(\kappa_\infty(V)\|NR_\oplus\|_\infty/\delta)}{-\log\rho} \right\rceil\right\}. \tag{12}$$

For non-diagonalizable Q, *the same exponential term is multiplied by a polynomial factor determined by the largest Jordan block.*

Proof idea. The gap is $R_\infty - R(d) = e_{s_0}^\top Q^d N R_\oplus$, so bounding it reduces to bounding $\|Q^d\|$; diagonalization gives $\|Q^d\|_\infty \le \kappa_\infty(V)\rho^d$ and Jordan blocks add a polynomial multiplier, so large $\rho(Q)$ or ill-conditioned eigenvectors imply longer step budgets. Together these results connect finite-horizon reliability, classical reliability growth, and conservative step budgets within one fitted-chain view.

VII. Simulation Validation

We keep the artifact's study numbering, so the theorem-backing simulations SS2–SS5 are reported there. The controlled checks below do not validate unseen operational traces; they show that the computation, analytic claims, and diagnostics behave as intended on controlled inputs.

A. SS1 and the Analytic Checks

SS1 isolates the numerical calculation behind the reliability estimates. We generate 500 random substochastic chains per size $m \in \{5, 10, 50, 100, 500\}$ and compare R_∞ (Proposition 1) with Monte Carlo estimates from 10^5 trajectories per chain, used as an independent simulation baseline for the same fitted chain rather than as a competing estimator. The absolute error $|R^{\mathrm{cf}}_\infty - \hat{R}^{\mathrm{MC}}_\infty|$ stays below the 5% acceptance threshold at every size, which supports the closed-form computations used later in SS6 and SS9 while leaving the Markov-fit question to the GoF protocol.

Three further controlled checks, reported in full in the artifact, confirm the analytic claims. The perturbation bound of Proposition 2 tracks the true $|R_\infty(\varepsilon) - R_\infty(0)|$ over $\varepsilon \in [0, 0.28]$, so it screens local changes by direction and scale before a full rerun. The correlated-trial gap of Theorem 4 grows with k and with $\mathrm{Var}(p(\xi))$, confirming that agreement, pass^k, and the independent-trial calculation are different projections of first-passage behavior rather than interchangeable scalars. Finally, across six rare-failure scaling regimes the cumulative first-passage curve converges to the Goel–Okumoto mean-value form (Proposition 7), so Goel–Okumoto appears as a limiting case of the agent-chain model rather than an unrelated baseline.

B. SS7: Goodness-of-Fit of the Agent-DTMC Assumption

SS7 tests the goodness-of-fit protocol of §IV-C (Algorithm 2) on three controlled conditions. The goal is not to make every trace Markov, but to separate corpora where the fitted-chain abstraction is credible from those where it is not.

(A) Markov ground truth. On 30 corpora of 300 traces from a known 1st-order absorbing chain ($m = 5$), the composite test retains the null in 100% of corpora at $\alpha = 0.05$; since a nominal 5% test would reject once or twice in 30 replicates, the observed size is *conservative* rather than exactly nominal. *(B) 2nd-order ground truth.* On 15 corpora from a 2nd-order chain with mixture weight 0.6, the FPT-KS layer alone rejects 0% while the AIC layer rejects 100% with $\Delta_{\mathrm{AIC}} \in [+979, +1614]$, so the composite accept-if-both-pass rule rejects 100%. *(C) MAST-derived self-consistency.* Sampling 500 synthetic traces from each of the 7 MAST-derived chains of §VIII-A, the composite test accepts all 7 (KS p-values $\in \{0.320, 0.374, 0.541, 0.828, 0.910, 0.994, 1.000\}$, with the AIC layer preferring the first-order model in every case). Because these traces are sampled from the fitted chain, (C) checks internal consistency, not whether real agent traces are Markov; the held-out study next checks whether the full pipeline recovers first-passage behavior on controlled MAST-style traces it did not fit.

SS1, the artifact checks of Propositions 2 and 7 and Theorem 4, and SS7 thus form a validation ladder that separates numerical correctness, analytic behavior, and diagnostic behavior from the held-out recovery tested next.

VIII. Empirical Case Studies

The empirical evidence serves two purposes: SS6 illustrates the reliability quantities on MAST-derived framework summaries, while SS9 uses a strict held-out split to test whether TraceToChain recovers the RDC and success-time distribution on unseen traces.

A. MAST Benchmark Formulation

SS6 uses transition matrices derived from public MAST summaries for seven frameworks, applies the fitted-chain quantities, and ranks frameworks by R_∞. The exercise is illustrative: it does not claim that raw MAST traces are Markov, but it shows what the reliability vocabulary reports once a chain is available. Asymptotic reliability spans $R_\infty \in [0.058, 0.450]$:

TABLE III
SS9: HELD-OUT RECOVERY ON SEVEN CONTROLLED MAST-STYLE CORPORA.

Framework	n_{fit}	n_{test}	m	D_{KS}	p_{KS}	L_∞^{RDC}
react [1]	200	200	5	0.020	1.000	0.048
reflexion [20]	200	200	5	0.033	0.984	0.052
cot_agent [3]	200	200	5	0.023	1.000	0.018
toolformer [2]	200	200	5	0.030	0.994	0.052
babyagi [3]	200	200	6	0.045	0.827	0.053
autogpt [3]	200	200	6	0.036	0.960	0.039
agentbench [10]	200	200	5	0.030	0.995	0.021

it is highest for toolformer [2] and lowest (≤ 0.07) for react [1], cot_agent, and reflexion [20], over $m \in [5, 12]$ states with $\rho(Q) \in [0.37, 0.66]$ and δ=0.01 horizons of 3–9 steps. The full table and the corresponding decay curves are in the artifact.

B. SS9: Held-Out Empirical Validation

SS7(C) is an in-sample self-consistency check; SS9 instead evaluates controlled held-out recovery with a strict fit/test protocol. The test set is not used to choose features, clusters, transition estimates, or tuning, so the reported KS and RDC errors measure held-out recovery rather than reuse of the training corpus.

a) Protocol.: For each of the 7 MAST-style frameworks we generate $n = 400$ trajectories from a ground-truth absorbing chain whose transient structure mimics the MAST taxonomy and emits noisy one-hot features (Gaussian noise $\sigma = 0.08$ and $\approx$5% censored traces). The corpus is split *before* featurization into $n_{\text{fit}} = 200$ and $n_{\text{test}} = 200$. We run TRACETOCHAIN on the fit half only.

Table III reports the recovered state count m, the KS agreement between the model success-time cumulative distribution function (CDF) $F^{\hat{M}}_{\tau_\oplus | \tau_\oplus < \tau_\ominus}$, conditional on eventual success and sampled with N=8,000 trajectories, and the held-out success-time CDF $\hat{F}_{\tau_\oplus}$ among successful traces. The table also reports the sup-norm discrepancy $L_\infty^{\text{RDC}} = \sup_{d \in [0,50]} |\mathcal{R}(d; \hat{M}) - \hat{\mathcal{R}}_{\text{emp}}(d)|$. These are all first-passage checks: they ask whether the fitted chain predicts *when* success occurs on traces it did not fit. The held-out empirical RDC is the non-parametric baseline over observed horizons; the fitted chain is useful only when it tracks that curve and passes GoF, and it then adds state-level interpretation, uncertainty propagation, perturbation analysis, metric reconciliation, and horizon queries beyond the observed grid.

b) Findings.: Figure 2 overlays, for each framework, the held-out empirical RDC $\hat{\mathcal{R}}_{\text{emp}}(d)$ (dashed) on the analytic $\mathcal{R}(d)$ of the fitted chain (solid), with each panel title reporting that framework's L_∞^{RDC} and p_{KS}. The two curves track each other over the whole budget range $d \in [0, 50]$—worst case 0.053 on babyagi—including the saturation elbow that the closed form must reproduce to answer horizon queries. Across all 7 frameworks, the composite diagnostic passes on held-out data at $\alpha = 0.05$. KS distances are small ($D_{\text{KS}} \in [0.020, 0.045]$), the minimum KS p-value is 0.827, and $\max L_\infty^{\text{RDC}} = 0.053$ with median 0.048. BabyAGI has the largest RDC discrepancy yet still passes KS. This supports recovery of the success first-passage distribution and RDC for these controlled MAST-style traces.

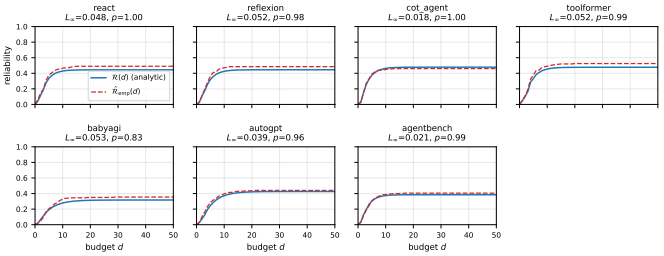

Fig. 2. SS9: held-out empirical RDC over the analytic $\mathcal{R}(d)$.

c) Scope.: SS9 uses synthetic traces from a genuine absorbing chain with noisy emissions, so the held-out KS test probes controlled recovery by the trace-to-chain pipeline: featurization, clustering, Markov-order testing, and MLE. It does not prove that operational agent traces are Markov; §IX takes that step on a real corpus. The MAST release itself lacks the step-level features required by ϕ.

These case studies support a narrow claim: when the diagnostics accept the trace-to-chain abstraction, the fitted chain provides a coherent reliability vocabulary for horizons and frameworks, with the limits set by the available trace representation and the GoF diagnostics.

IX. REAL AGENT-TRACE VALIDATION (SS10, SS14)

A. SS10: Does the Certificate Reject?

The decisive question is whether an *operational* agent trace corpus is adequately described by a first-order absorbing DTMC and, if not, whether the composite certificate rejects it. SS10 fits TRACETOCHAIN end-to-end to a real LLM-agent corpus and reports the outcome as-is.

a) Corpus and provenance.: We use the publicly released reasoning traces of a SWE-agent submission to the SWE-bench Verified leaderboard [8] (SWE-agent driving Claude-4-Sonnet), which archives one `.traj` file per instance recording the per-step `action`, `observation`, and `state`; the SWE-bench `resolved` verdict gives the ground-truth label. We harvested 479 instances (324 resolved, 155 unresolved; 29,504 steps). A rule-based featurizer—as in SS7—maps each step to a semantic action category by parsing the issued command (stripping `cd ... &&` wrappers and recording the file-editor sub-command), giving the alphabet {VIEW, EDIT, EXECUTE, SEARCH, INSPECT, VCS, SUBMIT, FILEOP, NONE, OTHER}. No trajectory text beyond these features is used; corpus and harvester are in the artifact.

b) Protocol.: We follow the SS9 protocol: the corpus is split 50/50 *before* featurization and clustering, TRACETOCHAIN is fitted on the fit half only, and the held-out half drives the Kolmogorov–Smirnov (KS) first-passage test and the empirical RDC. To keep Ward clustering tractable on a single machine we first draw a deterministic sub-sample of the

TABLE IV
SS10: TRACETOCHAIN ON 479 REAL SWE-AGENT TRAJECTORIES.

Quantity	**Value**
Instances used (succ/fail)	479 (324/155)
Agent steps	29504
Fit / test traces	80 / 80
Recovered states m	7
Silhouette	0.801
Δ_{AIC} (1st$-$2nd)	+817.2
First-order preferred (AIC)	no
Held-out KS D	0.393
Held-out KS p	0.0000
$L_{\infty}^{\mathrm{RDC}}$ (held-out)	0.260
Composite verdict	**REJECT**

corpus (uniform without replacement under a fixed seed) and split *that*; the reported run uses 160 traces in total, i.e. 80 fit and 80 test.

c) Result: the certificate rejects.: Table IV summarizes the fit. The clustering itself is clean, recovering $m{=}7$ well-separated states (silhouette 0.80): six are pure single-category states (EXECUTE, VIEW, EDIT, SEARCH, VCS, SUBMIT) and the seventh pools the low-frequency FILEOP, NONE, and OTHER steps. Despite this, the *composite certificate rejects the first-order absorbing DTMC*: the AIC layer strongly prefers a second-order chain ($\Delta_{\mathrm{AIC}} = +817$, so the first-order model is *not* selected), the held-out first-passage KS test rejects with $D_{\mathrm{KS}} = 0.39$, $p < 10^{-4}$, and the analytic RDC departs from the held-out empirical RDC by $L_{\infty}^{\mathrm{RDC}} = 0.26$. The rejection is stable at total sub-sample sizes of 120 and 160 traces and under both featurizations ($\Delta_{\mathrm{AIC}} \in [+470, +817]$, KS $p < 10^{-4}$ throughout).

d) Interpretation.: Real SWE-agent trajectories carry history dependence (retries, tool state, accumulated context) that a first-order chain over action categories does not capture; the AIC $\wedge$ KS certificate detects exactly this and *withholds* the first-passage interpretation, in line with the "reject if GoF fails" discipline of §XII. SS10 therefore validates the safeguard on operational data: on this corpus one should not report $\mathcal{R}(d)$ from a first-order DTMC, while the estimator-level recovery guarantees of SS9 and the diagnostic behavior of SS7 remain intact. §IX-B then takes the constructive next step: it localizes the misfit to the success-time distribution, shows that R_{∞} is nonetheless recovered accurately, and identifies the duration-aware model needed for horizon queries.

B. SS14: What the Audit Accepts

The natural follow-up is to ask which model the data *does* support and which reliability outputs are trustworthy. SS14 answers both; because the base states are the action categories themselves, no clustering is needed and we use the full 479-trace corpus (50/50 fit/test).

a) The misfit is in duration, not order.: An order-k Markov chain is a first-order absorbing chain on category k-tuples, so we can lift the corpus to $k \in \{1, 2, 3\}$ and to a duration-aware phase-type variant (state = (category, coarse progress stage)) and re-run the held-out success first-passage KS test. Table V reports the order lifts: the timing is rejected at *every* order ($p < 10^{-3}$), and a higher order does not help (it worsens with data sparsity). The reason is visible in Figure 3 (left): real successful runs are *concentrated* (held-out 10–90th percentile 34–86 steps, median 54, with a floor near 30), whereas a category-Markov chain produces a geometric-shaped first-passage time with mass near zero and a long tail. The phase-type lift narrows the gap substantially (KS D from 0.34 to 0.29) by letting the per-step hazard depend on progress, but does not fully close it. The non-Markovian structure of real agent traces is therefore primarily in the *success-time distribution*, and a genuinely duration-aware model (semi-Markov / phase-type with a learned holding-time law) is the indicated remedy—a sharper diagnosis than "the chain does not fit."

TABLE V
SS14: ORDER-k LIFTS OF THE REAL SWE-BENCH CORPUS.

order k	states	R_{∞}	KS D	KS p	verdict
1	10	0.656	0.341	0.000	REJECT
2	82	0.600	0.364	0.000	REJECT
3	318	0.546	0.425	0.000	REJECT

b) The asymptotic reliability is accepted, and the metric/perturbation use cases work on real data.: Crucially, the audit does not reject everything: it separates the trustworthy outputs from the untrustworthy ones. The fitted chain's asymptotic reliability R_{∞} (eventual success probability) *matches the held-out empirical pass rate* on the full 479-trace corpus: at first order $R_{\infty} = 0.656$ versus an observed pass rate of 0.683, i.e. within 0.027 (Figure 3, right; the parsimonious first-order chain is best, as higher orders overfit R_{∞}). Two of the three motivating use cases therefore pay off directly on real traces, because they depend on R_{∞} rather than on the rejected timing: *metric reconciliation*—from the single fitted R_{∞} we read $\mathrm{pass@1} = 0.66$, $\mathrm{pass@5} = 0.995$, $\mathrm{pass}^5 = 0.12$, reconciling the three conventions without a third denominator; and *fallback-tool perturbation*—rerouting 10% of each state's failure mass to success yields $\Delta R_{\infty} = +0.034$. Because this moves mass into $R_{\oplus}$ rather than within Q, it falls outside the scope of Proposition 2 and is computed as an *exact* re-evaluation of Proposition 1 on the modified chain; either way it needs no benchmark rerun. The third use case, horizon extrapolation $\mathcal{R}(d)$, is exactly the quantity the KS test rejects, so the audit *withholds* it and flags that a duration-aware model is required before reporting it.

On real agent traces the framework is therefore neither uniformly valid nor uniformly useless—the intended behavior of an estimation-and-audit pipeline.

X. CROSS-BENCHMARK REAL-DATA VALIDATION (SS15)

A single corpus is one verdict. SS15 tests whether the SS10/SS14 pattern *generalizes* to a second, very different real benchmark—the conversational tool-use benchmark τ-bench—across two models and two task domains.

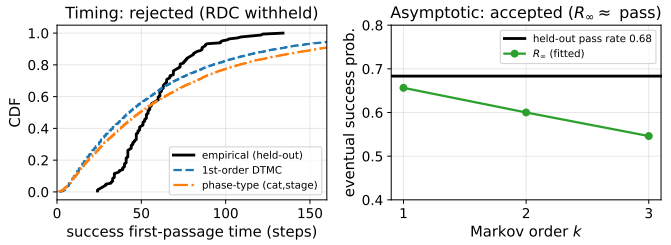

Fig. 3. SS14: success first-passage CDF (left) and R_∞ by Markov order (right) on real SWE-agent traces.

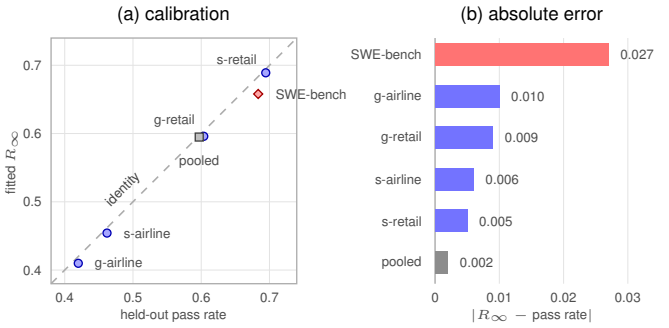

Fig. 4. Fitted R_∞ vs. held-out pass rate on five real corpora.

a) Corpus and provenance.: We use the publicly released τ-bench trajectory archive [7]: Claude-3.5-Sonnet and GPT-4o on the retail and airline domains, 1,980 episodes in total, each carrying the task reward as a ground-truth success label. The agent's actions are its *assistant turns*, mapped to five categories—respond, lookup (read-only tool), write (state-changing tool), handoff, and think—and the episode absorbs by reward. Since the MLE and the first-passage KS test depend only on transition counts and the success-length distribution, we retain per-domain sufficient statistics and ship the extractor in the artifact.

b) The asymptotic reliability is accepted—on every domain and model.: For each of the four model×domain runs ($n = 920$, 400, 460, and 200 episodes) and for the pooled corpus, the fitted first-order R_∞ recovers the held-out empirical pass rate almost exactly: the mean absolute error across the four runs is 0.0075, and the pooled estimate is 0.595 versus an observed pass rate of 0.597 ($|\text{err}| = 0.002$). Figure 4 places these estimates alongside the SWE-agent corpus of §IX. Panel (a) plots fitted R_∞ against the held-out pass rate for all five real corpora: every point lies on the identity line, and the two hardest (τ-bench airline, 0.42–0.46) sit on it as tightly as the two easiest, so the estimator is calibrated over the full competence range, not only where agents succeed often. Panel (b) shows the residuals: four corpora and the pooled estimate fall at or below 0.010, and the largest—SWE-bench at 0.027—is an order of magnitude below the 0.27 spread in pass rates separating the corpora. They are also one-signed, R_∞ sitting marginally *below* the observed rate everywhere, consistent with the conservative exit estimates censoring induces (§XI). Calibration therefore holds across a wide difficulty spread and two underlying models—evidence that the asymptotic output is trustworthy on real agents, not an artifact of one corpus. Consequently the metric-reconciliation and perturbation use cases (which depend only on R_∞) carry over to τ-bench unchanged.

c) The timing behaves exactly as the certificate dictates.: As on SWE-bench, the success-conditional first-passage KS test rejects the first-order timing on every domain ($p \leq 0.001$ throughout; on sonnet-retail the empirical median success length is 13 steps versus a fitted-chain median of 9), so the audit again *withholds* horizon $\mathcal{R}(d)$ while keeping the accepted asymptotic and metric outputs. The accept-the-asymptotic / withhold-the-timing behavior is thus consistent across two independent real benchmarks, four model×domain corpora, and a wide range of agent competence.

XI. Robustness and Sensitivity (SS11–SS13)

The diagnostics and estimates depend on three modeling choices—the clustering that builds the state space, the treatment of censored traces, and the bootstrap path—and this section reports a controlled sensitivity study for each.

a) SS11: clustering and featurization sensitivity.: Because the certificate is a *joint* test of featurization, clustering, and the first-order assumption, its verdict could in principle be an artifact of the clustering. We therefore re-ran the SS10 fit across 18 configurations: two featurizations (action-category one-hot, with and without a step-depth feature), three agglomerative linkages (Ward, average, complete), and three cluster-count caps $k_{\max} \in \{6, 8, 10\}$. Table VI reports the result: the composite verdict is *invariant*, with all 18 configurations rejecting the first-order absorbing DTMC. The held-out first-passage KS test rejects ($p < 10^{-3}$) in every configuration, including the handful of high-k settings where the AIC order-test component alone would accept a first-order chain—i.e. the held-out KS layer is the binding, clustering-robust component of the certificate, exactly as intended. The recovered state count m tracks $k_{\max}$ (range 6–10) and the silhouette varies with featurization, but $\hat{R}_\infty$ is stable across all configurations (0.588–0.600): the estimates move modestly with clustering choices, and the verdict not at all.

Reconciling the two SWE-bench R_∞ estimates. The $\hat{R}_\infty$ column of Table VI and the $R_\infty = 0.656$ of SS14 come from different fits of the same benchmark, and each must be read against its own target: the 160-trace sub-sample used here has a held-out pass rate of 0.663, whereas the full 479-trace corpus used by SS14 has 0.683. Sub-sampling therefore moves the target as well as the estimate, and comparing an $\hat{R}_\infty$ fitted on one split against the held-out rate of another is not meaningful. Because it uses the whole corpus and needs no clustering, the calibration claim of §IX-B and the SWE-bench point of Fig. 4 are taken from the SS14 fit. We recommend reporting R_∞ only against the held-out rate of the split that produced it.

b) SS12: censoring sensitivity.: The censoring figures reported elsewhere ($\leq 10\%$) are mild, and real deployments

TABLE VI
SS11: CLUSTERING AND FEATURIZATION ABLATION (18 CONFIGURATIONS).

feat.	linkage	$k_{\max}$	m	silh.	Δ_{AIC}	KS p	verdict
cat+depth	ward	6	6	0.77	+548	0.000	REJECT
cat+depth	ward	8	7	0.80	+543	0.000	REJECT
cat+depth	ward	10	7	0.80	+543	0.000	REJECT
cat+depth	average	6	6	0.75	+473	0.000	REJECT
cat+depth	average	8	8	0.78	+152	0.000	REJECT
cat+depth	average	10	10	0.81	-431	0.000	REJECT
cat+depth	complete	6	6	0.77	+572	0.000	REJECT
cat+depth	complete	8	8	0.78	+152	0.000	REJECT
cat+depth	complete	10	10	0.81	-431	0.000	REJECT
cat	ward	6	6	0.95	+548	0.000	REJECT
cat	ward	8	8	0.99	+395	0.000	REJECT
cat	ward	10	10	1.00	-431	0.000	REJECT
cat	average	6	6	0.95	+548	0.000	REJECT
cat	average	8	8	0.99	+377	0.000	REJECT
cat	average	10	10	1.00	-431	0.000	REJECT
cat	complete	6	6	0.79	+551	0.000	REJECT
cat	complete	8	8	0.99	+333	0.000	REJECT
cat	complete	10	10	1.00	-431	0.000	REJECT

may truncate far more aggressively. We took a known ground-truth absorbing chain ($R_\infty = 0.784$), sampled 600 trajectories, and refit TRACETOCHAIN after right-censoring a controlled fraction of traces (truncating at a uniform-random transient step and marking the trace as censored, neither success nor failure). Under this non-informative censoring the recovered $\hat{R}_\infty$ stays within $\approx 4\%$ of the truth at every level: the relative bias moves only over $[-4.2\%, -2.6\%]$ across censoring fractions 0–50%, against -3.3% with no censoring at all, so censoring contributes almost none of the residual. Censored traces still contribute their observed transient transitions but are correctly excluded from the absorption counts, so the exit estimates are not materially biased. This holds for censoring independent of the outcome; *informative* censoring (e.g. a step budget that truncates precisely the long, failure-bound trajectories) could still bias the estimates, and detecting that regime is left to future work.

c) SS13: bootstrap path and uncertainty understatement.: The intervals of §IV-D are reported via the fast bootstrap path, which holds the target clustering fixed (nearest-centroid reassignment of resampled steps). The module also implements a full re-cluster bootstrap (`fast=False`) that re-runs Ward clustering on every resample and so additionally captures clustering instability. On a controlled 250-trace corpus the full re-cluster path gives transition intervals whose median half-width is $\approx 4.7\times$ that of the fixed-clustering path (0.158 vs. 0.034). The fixed-clustering intervals therefore understate total uncertainty and should be read as a lower bound conditioned on the chosen clustering; we recommend reporting the full re-cluster path (or both) when clustering stability is in question.

XII. THREATS TO VALIDITY

The main threat is overinterpreting the fitted abstraction. The chain remains a reliability model for specified trace features under specified diagnostics, not a universal model of agent behavior, so the estimates should be read conditionally.

a) Construct validity (state construction).: Mapping trajectories to a DTMC aggregates over memory, tool state, and context, and no single mapping is canonical. §IV gives a reproducible construction whose output is *tested* by Algorithm 2: SS7 checks the protocol on known synthetic chains and SS9 checks controlled fit/test recovery. These establish statistical adequacy for the chosen features, not uniqueness of ϕ. If GoF rejects, first-passage interpretation should stop until traces are segmented, re-featurized, or modeled with a richer process; semi-Markov or hidden Markov models (HMMs) are the natural next steps. We therefore recommend reporting $(\phi, m, p_{\mathrm{KS}}, \Delta_{\mathrm{AIC}})$ with each estimate.

b) Estimator sensitivity.: The composite verdict is invariant to sub-sampling and clustering (SS11), but R_∞ is a property of the fit that produced it: sub-sampling changes the held-out target as well as the estimate (§XI), so R_∞ must always be reported against the held-out rate of the same split.

c) Empirical scope of SS9.: SS9 uses synthetic traces from a true absorbing chain with noisy one-hot emissions, so its held-out KS test evaluates controlled recovery by the trace-to-chain pipeline (featurizer + clustering + MLE) rather than the full modeling class on unprocessed benchmark data; §IX–X supply the real-trace counterpart.

d) Independence assumption.: The pass^k and pass@k identities require i.i.d. trials (A4), which can fail under temperature-0 sampling or shared prefixes. Theorem 4 and Corollary 5 quantify the resulting bias and provide a diagnostic.

e) External validity.: The MAST case study covers 7 frameworks, so its conclusions may not generalize to other agents, tasks, or trace distributions; SS6 is therefore framed as illustrative rather than confirmatory.

f) NHPP-limit scope.: The NHPP limit applies under rare-failure scaling ($\varepsilon \to 0$, $d \to \infty$); in high-failure regimes ($\varepsilon > 0.1$) the exact closed form of Proposition 1 should be used instead. That is the common case for deployed agents—the SWE-agent corpus of §IX fails on roughly 32% of tasks ($155/479$)—so the Goel–Okumoto/NHPP limit is best treated as a conceptual bridge to reliability-growth theory rather than an estimator at these rates.

XIII. DISCUSSION

For benchmark authors, SRE teams, and method developers, the fitted chain provides a shared reliability object at the trace level: it supports first-passage metrics with uncertainty, horizon and perturbation queries, and diagnostics that can reject an inadequate abstraction. SS9 (Table III) shows that, once the trace-to-chain step is specified and checked, $\hat{M}$ tracks held-out first-passage behavior to within $L_\infty^{\mathrm{RDC}} \leq 0.053$ across seven frameworks. This does not replace benchmarks or incident analysis; it shows how benchmark-style trace corpora can support auditable reliability answers while also flagging when the abstraction is too weak for them. The practical requirement is to fit the chain, report uncertainty, run GoF, and withhold first-passage conclusions when the checks fail.

The real-data studies make this concrete and show the audit is *selective* rather than all-or-nothing: across SWE-bench (SS10, SS14) and τ-bench (SS15) it accepts the asymptotic reliability but rejects the first-order RDC timing, while SS11–SS13 (§XI)

show the verdict survives the clustering choice even where the point estimate does not. The value of the fitted chain on real traces is thus real but *partial*, and the audit is what makes it safe by saying which part to trust.

The clearest next step, identified by the data rather than assumed, is a duration-aware model: SS14 localizes the misfit to the success-time distribution, so semi-Markov holding times (or a learned phase-type duration law) are the indicated route to restoring horizon queries. Further directions include HMM reliability under partial observability, online estimation of Q, integration with PRISM and Storm, and mixture-chain models for cross-trial correlation [39].

XIV. Conclusion

LLM-agent teams already collect execution traces, but scalar metrics alone do not answer deployment reliability questions. TraceToChain turns those traces into an audited absorbing-chain model, combining a composite AIC $\wedge$ KS certificate, Dirichlet-posterior and bootstrap intervals, and a strict held-out protocol, and it recasts pass@k, pass^k, and the RDC as projections of a single success-time distribution. On seven controlled MAST-style frameworks the analytic and held-out empirical RDCs agree to $\max L_\infty^{\mathrm{RDC}} = 0.053$ with KS acceptance on $7/7$ corpora. On $2{,}459$ real episodes from two independent benchmarks the certified asymptotic reliability recovers the held-out pass rate to within 0.03 on SWE-bench and to a mean of 0.0075 across four τ-bench corpora, delivering metric reconciliation and closed-form perturbation answers on real agents with no benchmark rerun, and the certificate marks precisely which further outputs its evidence supports. The verdict is invariant across 18 clustering configurations, and $\hat{R}_\infty$ stays within 4% of a synthetic ground truth up to 50% censoring. The result is a reliability model that reports not only a number but the evidence that licenses it.

References

[1] S. Yao *et al.*, "ReAct: Synergizing reasoning and acting in language models," in *International Conference on Learning Representations*, 2023.

[2] T. Schick *et al.*, "Toolformer: Language models can teach themselves to use tools," *Advances in Neural Information Processing Systems*, 2023.

[3] L. Wang *et al.*, "A survey on large language model based autonomous agents," *Frontiers of Computer Science*, vol. 18, no. 6, p. 186345, 2024.

[4] A. Avižienis *et al.*, "Basic concepts and taxonomy of dependable and secure computing," *IEEE Transactions on Dependable and Secure Computing*, vol. 1, no. 1, pp. 11–33, 2004.

[5] B. Beyer *et al.*, *Site Reliability Engineering: How Google Runs Production Systems*. Sebastopol, CA: O'Reilly Media, 2016.

[6] Y. Qin *et al.*, "ToolLLM: Facilitating large language models to master 16000+ real-world APIs," *International Conference on Learning Representations (ICLR)*, 2024.

[7] S. Yao *et al.*, "τ-bench: A benchmark for tool-agent-user interaction in real-world domains," in *International Conference on Learning Representations (ICLR)*, 2025.

[8] C. E. Jimenez *et al.*, "SWE-bench: Can language models resolve real-world GitHub issues?" in *International Conference on Learning Representations*, 2024.

[9] S. Zhou *et al.*, "WebArena: A realistic web environment for building autonomous agents," in *International Conference on Learning Representations*, 2024.

[10] X. Liu *et al.*, "AgentBench: Evaluating LLMs as agents," in *International Conference on Learning Representations*, 2024.

[11] M. Chen *et al.*, "Evaluating large language models trained on code," *arXiv preprint arXiv:2107.03374*, 2021.

[12] A. Khanal, Y. Tao, and J. Zhou, "Beyond pass@1: A reliability science framework for long-horizon LLM agents," *arXiv preprint arXiv:2603.29231*, 2026.

[13] R. C. Cheung, "A user-oriented software reliability model," *IEEE Transactions on Software Engineering*, vol. SE-6, no. 2, pp. 118–125, 1980.

[14] J. G. Kemeny and J. L. Snell, *Finite Markov Chains*. Princeton, NJ: Van Nostrand, 1960.

[15] A. L. Goel and K. Okumoto, "Time-dependent error-detection rate model for software reliability and other performance measures," *IEEE Transactions on Reliability*, vol. R-28, no. 3, pp. 206–211, 1979.

[16] J. D. Musa, A. Iannino, and K. Okumoto, *Software Reliability: Measurement, Prediction, Application*. New York, NY: McGraw-Hill, 1987.

[17] R. Koohestani *et al.*, "TriCEGAR: A trace-driven abstraction mechanism for agentic AI," *arXiv preprint arXiv:2601.22997*, 2026.

[18] H. Wang *et al.*, "ProbGuard: Proactive runtime monitoring for LLM agent safety via probabilistic prediction," *arXiv preprint arXiv:2508.00500*, 2025.

[19] K. S. Trivedi and A. Bobbio, *Reliability and Availability Engineering: Modeling, Analysis, and Applications*. Cambridge, UK: Cambridge University Press, 2017.

[20] N. Shinn *et al.*, "Reflexion: Language agents with verbal reinforcement learning," in *Advances in Neural Information Processing Systems*, 2023.

[21] M. Shridhar *et al.*, "ALFWorld: Aligning text and embodied environments for interactive learning," in *International Conference on Learning Representations*, 2021.

[22] G. Wang *et al.*, "Voyager: An open-ended embodied agent with large language models," *Transactions on Machine Learning Research*, 2024.

[23] M. Cemri *et al.*, "Why do multi-agent LLM systems fail?" *arXiv preprint arXiv:2503.13657*, 2025.

[24] H. Hansson and B. Jonsson, "A logic for reasoning about time and reliability," *Formal Aspects of Computing*, vol. 6, no. 5, pp. 512–535, 1994.

[25] C. Baier and J.-P. Katoen, *Principles of Model Checking*. Cambridge, MA: MIT Press, 2008.

[26] M. Kwiatkowska, G. Norman, and D. Parker, "PRISM 4.0: Verification of probabilistic real-time systems," in *Computer Aided Verification*. Springer, 2011, pp. 585–591.

[27] C. Dehnert *et al.*, "A storm is coming: A modern probabilistic model checker," in *Computer Aided Verification*. Springer, 2017, pp. 592–600.

[28] J. H. Ward, "Hierarchical grouping to optimize an objective function," *Journal of the American Statistical Association*, vol. 58, no. 301, pp. 236–244, 1963.

[29] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53–65, 1987.

[30] H. Akaike, "A new look at the statistical model identification," *IEEE Transactions on Automatic Control*, vol. 19, no. 6, pp. 716–723, 1974.

[31] N. V. Smirnov, "On the estimation of the discrepancy between empirical curves of distribution for two independent samples," *Bulletin de l'Université de Moscou, Série Internationale (Mathématiques)*, vol. 2, no. 2, pp. 3–14, 1939.

[32] G. W. Stewart, *Matrix Algorithms, Volume I: Basic Decompositions*. Philadelphia, PA: SIAM, 1998.

[33] D. J. Daley, "Stochastically monotone Markov chains," *Z. Wahrscheinlichkeitstheorie verw. Gebiete*, vol. 10, pp. 305–317, 1968.

[34] S. Karlin, *Total Positivity, Volume I*. Stanford, CA: Stanford University Press, 1968.

[35] J. Keilson and A. Kester, "Monotone matrices and monotone Markov processes," *Stochastic Processes and their Applications*, vol. 5, no. 3, pp. 231–241, 1977.

[36] S. Karlin and Y. Rinott, "Classes of orderings of measures and related correlation inequalities. I. multivariate totally positive distributions," *Journal of Multivariate Analysis*, vol. 10, no. 4, pp. 467–498, 1980.

[37] R. A. Horn and C. R. Johnson, *Matrix Analysis*, 2nd ed. Cambridge, UK: Cambridge University Press, 2013.

[38] D. A. Levin and Y. Peres, *Markov Chains and Mixing Times*, 2nd ed. American Mathematical Society, 2017.

[39] F. Spaeh, K. Sotiropoulos, and C. E. Tsourakakis, "ULTRA-MC: A unified approach to learning mixtures of markov chains via hitting times," *arXiv preprint arXiv:2405.15094*, 2024.