From Verification to Commitment: Histories, Repair Warrants, and Admissible Continuation in Agentic Systems

Flyxion

2026

Abstract

Recent work on agentic systems has begun to move beyond evaluating isolated model outputs toward reasoning about persistent operational state, temporally extended failure, and the repair of representations. This transition is necessary but incomplete. A system may verify a constraint without thereby possessing grounds to intervene; it may detect a failure without identifying an admissible repair; and it may validate a repair without acquiring authority to commit its result. This article develops a constraint-first account of agentic continuation organized around five distinct objects: retained history, diagnosis, an admissible continuation set, a repair warrant, and commitment. The account is developed through a critical synthesis of formal verification over operational data, trajectory-based failure detection, LLM-guided constraint-model reformulation, and subtractive approaches to human-judgment preservation. The resulting architecture, CLIO, treats verification as a restriction on possible continuation rather than an automatic license for action. It distinguishes vertical repair within a representation from horizontal repair of the representation itself, and it treats null intervention as a positive outcome when diagnostic evidence is insufficient to select among surviving possibilities. The central claim is that safe agentic action requires explicit non-implications between detection, repair, and commitment.

1 Introduction

The usual semantic unit of language-model evaluation is an output. A response is judged correct or incorrect, a tool call permitted or forbidden, or a final answer compared with a reference. This unit becomes inadequate once an agent acts through tools on persistent data. A tool call changes the state in which later calls are interpreted. Its significance is therefore neither exhausted by its textual form nor fixed at the moment of issue. The same call may be harmless, necessary, redundant, or destructive depending on the history that precedes it and the continuations it makes reachable.

Recent research reflects a corresponding movement from output-centered evaluation toward histories and transition systems. Mercado and Lomuscio formalize agents operating over relational operational data and verify requirements over the transition systems induced by their policies and tools [1]. Dubey detects failures from temporal execution telemetry and shows that historical information becomes increasingly useful as a failed trajectory develops [2]. Voboril and Szeider permit an LLM agent to reformulate a constraint model while testing proposed formulations against the original problem [3]. Chen argues that assistance should sometimes remove impermissible possibilities without selecting a preferred outcome on a person's behalf [4]. Together, these works reveal a shared transition from isolated choice to constrained continuation.

They also expose an unresolved problem. Verification establishes that a property holds or fails over a specified system. Detection supplies evidence that an execution has departed from an expected regime. Constraint testing rejects or retains a candidate reformulation. Subtractive assistance removes inadmissible possibilities. None of these operations, by itself, explains when a particular intervention is warranted or when its result should become operative.

This article proposes that an agentic architecture must distinguish five objects and the partial relations among them:

$$H \rightsquigarrow D(H) \rightsquigarrow A(H, D) \rightsquigarrow R \rightsquigarrow C(R), \tag{1}$$

where H is retained history, D a diagnosis, A the set of admissible continuations, R a proposed repair, W a warrant relation evaluating that repair, and C an explicit commitment operation. None of the arrows is assumed total or functional. The notation is not a pipeline in which every object necessarily produces the next. Its purpose is precisely to preserve the gaps between them. A history may be insufficient for diagnosis; a diagnosis may justify suspension but not repair; a proposed repair may be admissible without being uniquely warranted; and a successful repair may remain uncommitted.

The name CLIO will refer here to an architecture organized around those distinctions. CLIO is not offered as another detector or verifier. It is a coordination layer for determining what follows—and what does not follow—from the results such components provide.

2 Histories, continuations, warrants, and commitment

Let Σ be an operative state space, E a space of recordable events, and $\mathcal{H} = E^*$ the space of finite documentary histories. A configuration is a pair

$$(\sigma, H) \in \Sigma \times \mathcal{H}. \tag{2}$$

The distinction between σ and H is constitutive. Proposals, evaluations, refusals, and failed repairs may extend H without changing σ. Documentary history preserves evidence; operative state determines the reference against which the next admitted transition acts.

Let $\mathcal{R}(H, \sigma)$ denote the candidate continuations currently expressible from (H, σ). Admissibility is not intrinsic to a candidate considered in isolation. It is a relation

$$\mathsf{Adm} \subseteq \mathcal{H} \times \Sigma \times \mathcal{R}, \tag{3}$$

whose extension at a configuration is the continuation space

$$\mathcal{A}(H, \sigma, D) = \{r \in \mathcal{R}(H, \sigma) : \mathsf{Adm}(H, \sigma, D, r)\}. \tag{4}$$

The diagnostic argument appears because the constraints available to evaluation may depend on what the retained evidence supports. More explicitly, let $\mathcal{K}(H, \sigma, D)$ be the currently operative family of structural, epistemic, operational, and recursive constraints. Their aggregation need not be a simple conjunction or product; let F be the declared aggregation rule. Then

$$\mathsf{Adm}(H, \sigma, D, r) \iff F\big(\alpha_{\text{str}}, \alpha_{\text{epi}}, \alpha_{\text{ope}}, \alpha_{\text{rec}}\big)(H, \sigma, D, r) = 1. \tag{5}$$

A verification operation is consequently a narrowing operator on the characteristic predicate of a continuation space. For a newly imposed constraint K_j,

$$N_j(\chi_{\mathcal{A}})(r) = \chi_{\mathcal{A}}(r) \wedge K_j(H, \sigma, D, r). \tag{6}$$

Verification changes which continuations remain supported; it need not change operative state.

Diagnosis is likewise relational. Write

$$\mathsf{Diag} \subseteq \mathcal{H} \times \Sigma \times \mathcal{D}. \tag{7}$$

For some configurations there may be no D such that $\mathsf{Diag}(H, \sigma, D)$, while others may support several diagnoses. The notation $D(H)$ is therefore shorthand for a possibly empty, possibly plural diagnostic relation rather than a universally defined function.

When an implementation enforces deterministic diagnosis, this relation reduces to a partial operator

$$D : \mathcal{H} \times \Sigma \rightharpoonup \mathcal{D}, \tag{8}$$

with $D(H, \sigma)$ undefined at an epistemic boundary. Likewise, Equation (4) induces an admissible-continuation operator

$$A : \mathcal{H} \times \Sigma \times \mathcal{D} \longrightarrow 2^{\mathcal{R}}. \tag{9}$$

Given a diagnosis and admissible set, a repair warrant is a relation

$$\mathsf{W} \subseteq \mathcal{D} \times 2^{\mathcal{R}} \times \mathcal{R}. \tag{10}$$

Several repairs may be warranted simultaneously. The difference between

$$\exists r\ \mathsf{W}(D, \mathcal{A}, r) \qquad \text{and} \qquad \exists! r\ \mathsf{W}(D, \mathcal{A}, r) \tag{11}$$

is the formal difference between supported intervention and supported closure. Even uniqueness, however, does not by itself confer authority. Let

$$\mathsf{Auth} \subseteq \mathcal{H} \times \Sigma \times \mathcal{R} \tag{12}$$

record the independent authorization required for operative change. Commitment is a partial transition relation

$$\mathsf{Commit} \subseteq (\Sigma \times \mathcal{H}) \times \mathcal{R} \times (\Sigma \times \mathcal{H}) \tag{13}$$

defined only when admissibility, warrant, and authority all hold. Thus

$$\mathsf{Adm}(H, \sigma, D, r) \not\Rightarrow \exists(\sigma', H')\ \mathsf{Commit}((\sigma, H), r, (\sigma', H')). \tag{14}$$

Equation (14) is the admissible-but-non-committable proposition at the center of CLIO. Verification can establish membership in a permitted region. It cannot manufacture either a repair warrant or commitment authority.

3 Operational semantics and verification obligations

The preceding definitions become architectural constraints only when attached to an operational semantics. Let

$$\mathcal{M}_{\text{CLIO}} = (Q, \Lambda, \longrightarrow), \qquad Q = \Sigma \times \mathcal{H}, \tag{15}$$

be a labeled transition system. The event alphabet Λ contains at least proposal, diagnosis, verification, warrant, authorization, suspension, refusal, commitment, rollback, and representational-revision events. Partition it into documentary labels Λ_D and operative labels Λ_O. Documentary transitions may extend history but are the identity on operative state:

$$(\sigma, H) \xrightarrow{e} (\sigma, H \cdot e), \qquad e \in \Lambda_D. \tag{16}$$

Operative transitions may change σ, but every such transition is subject to the commitment contract:

$$(\sigma, H) \xrightarrow{\mathsf{commit}(r)} (\sigma', H \cdot \mathsf{commit}(r)) \Longrightarrow \begin{cases} r \in \mathcal{A}(H, \sigma, D), \\ \mathsf{W}(D, \mathcal{A}, r), \\ \mathsf{Auth}(H, \sigma, r). \end{cases} \tag{17}$$

These clauses are verification conditions for an implementation, not conclusions obtained merely by naming the architecture. A verifier must enumerate every rule capable of changing σ and establish that none bypasses Equation (17).

Proposition 1 (Non-bypass)**.** *Assume that every transition label outside Λ_O obeys Equation* (16)*, and that every label in Λ_O is governed by Equation* (17)*. Then no proposal, diagnosis, verification, warrant, refusal, suspension, or history-maintenance transition changes operative state, and every operative change is admissible, warranted, and authorized.*

Proof. The first result follows by case analysis over Λ_D, whose rules preserve σ. The second follows by case analysis over Λ_O and application of the commitment contract. The proof is nontrivial for a concrete implementation only insofar as the transition rules are complete: any unenumerated state-mutating path invalidates the premise. □

Suspension preserves the value of operative state, but this does not imply that the state is safe. The correct independence claim is

$$\mathsf{suspend} \not\equiv \mathsf{fail}, \qquad \mathsf{suspend} \not\equiv \neg\mathsf{fail}. \tag{18}$$

A system may suspend before any operative requirement has been violated, or it may suspend in an already failed state because the evidence does not warrant a repair.

Finite verification additionally requires an explicitly bounded fragment. If Σ, E, $\mathcal{D}$, and $\mathcal{R}$ are finite and histories have length at most N, then

$$|\mathcal{H}_{\leq N}| = \sum_{k=0}^{N} |E|^k, \tag{19}$$

and the induced transition system is finite. Ordinary reachability and temporal properties can then be model-checked. Unbounded data parameters or unbounded history destroy this argument even when event names are finite. Symmetry reductions such as those studied by Mercado and Lomuscio may recover exact finite verification under additional boundedness and equivariance conditions; graph-isomorphism hardness is a computational lower bound on canonicalization, not an undecidability result.

4 The trajectory as semantic object

Mercado and Lomuscio's Stateful Tool-Enabled Agentic Deployments, or STEADs, make persistent operational data part of the formal object being verified [1]. A deployment consists of persistent state, a tool space, and an LLM-based policy. Tool calls induce transitions between relational states, and First-Order Computation Tree Logic expresses safety and progress requirements over the resulting branching transition system. This changes the location of correctness. Correctness no longer belongs only to a selected call; it belongs to the reachable structure produced by calls, tool effects, and accumulated data.

The distinction is important because interface-level restraint cannot generally establish system-level progress. Blocking a prohibited call may prevent one unsafe transition, but it cannot ensure that an outstanding case is eventually completed. Conversely, a locally permissible call may participate in a globally defective history. Requirements concerning approval, eventual resolution, or the preservation of relations among objects are properties of evolving state.

The framework also establishes a decisive limit: unrestricted verification is undecidable. Exact finite verification becomes available only under boundedness and symmetry conditions, including equivariance of the agent under renaming of opaque identifiers. This boundary discourages the fantasy of a universal external judge. Verification becomes possible through disciplined choices about representation and transition semantics.

Identifier equivariance has a broader significance. Suppose two persistent states differ only in the names assigned to opaque objects. If the agent selects corresponding actions under the same renaming, the representations are operationally interchangeable. If it does not, their apparent structural equivalence fails at the level of intervention. Thus observational equivalence is weaker than intervention-preserving equivalence:

$$x \equiv_{\text{obs}} y \not\Rightarrow x \equiv_{\text{act}} y. \tag{20}$$

An admissibility interface must consequently preserve not only values that can be observed but also the action semantics induced by those values. Mercado and Lomuscio enforce the required symmetry using canonical decision contexts: isomorphic contexts are presented to the agent in a common form, and selected calls are translated back to the original identifiers. Their graph-isomorphism-hardness result shows that canonicalization is not merely clerical normalization. Determining a representation that safely supports substitution may itself be computationally substantial.

STEADs verify a branching set of possible histories, whereas an operating system also accumulates one documentary history of what was proposed, attempted, rejected, and committed. CLIO therefore distinguishes the transition structure used to reason about reachability from the audit structure used to retain evidence. Possible futures and recorded pasts are related but not identical objects. The former supports verification; the latter supports diagnosis, accountability, replay, and revision of the verification scheme itself.

5 Diagnosis does not entail repair

Dubey's experiments make the diagnostic value of history empirically visible [2]. A stateful echo-state-network monitor increasingly outperforms a memoryless baseline as the post-failure horizon grows. Loops, cascading tool failures, drift, fabrication, and corrupted-content absorption leave temporal signatures that may be weak at onset but become legible across subsequent steps. A failed execution is therefore not merely a collection of defective states. Its shape carries evidence.

The strongest results, however, come from deterministic verification. When a claimed total can be recomputed from actual tool results, or when required calls can be checked directly, explicit invariants outperform judgments about whether behavior merely appears anomalous. This supports a constraint-first hierarchy: when a relevant invariant is available, test it; use learned anomaly detection for the residual domain in which no such invariant has been articulated.

Dubey closes detection into repair by rolling flagged executions back and rerunning them. The intervention substantially improves recovery over resampling. Yet this successful engineering result leaves open a logical question. From

$$\text{fault detected} \tag{21}$$

it does not follow that

$$\text{repair warranted.} \tag{22}$$

Nor, even when some intervention is warranted, does it follow that

$$\text{this repair is warranted.} \tag{23}$$

A detector may justify stopping a continuation without identifying its cause. A failed coverage check may prove that a result cannot be committed while supplying no evidence about which earlier state should be restored. A loop alarm may justify suspension, but not whether the system should retry, change tools, revise its representation, request human judgment, or abandon the task.

Detection identifies a defect relative to some coordinate system; repair requires evidence connecting that defect to a particular transformation.

CLIO calls this connection a repair warrant. Formally, a warrant is not a scalar confidence value attached to a diagnosis, but the relation in Equation (10). A repair may be refused because it violates an invariant. One or more repairs may be warranted because the diagnosis and retained evidence specifically support them. The situation remains underdetermined when the evidence excludes the current trajectory without selecting a unique consequential class of alternatives.

The underdetermined case is crucial. Conventional agent loops tend to treat uncertainty as a reason to sample again. CLIO instead recognizes null intervention as a positive continuation operator. Let $\mathsf{suspend}$ be an event and define

$$\mathsf{N}_D(\sigma, H) = (\sigma, H \cdot \mathsf{suspend}(D, \mathcal{A})). \tag{24}$$

This transformation changes documentary and epistemic status while preserving operative state and the plurality of surviving continuations. It is warranted when an anomaly is supported but no uniquely authorized repair class has been established. Null intervention does not claim that nothing is wrong. It claims that the available diagnosis does not warrant choosing a repair. Suspension is therefore an event with semantics, not an absence of system behavior.

5.1 Admissibility, warrant, and closure

The separation among admissibility, warrant, and commitment can now be stated directly. For a candidate repair r,

$$r \in \mathcal{A}(H, \sigma, D) \not\Rightarrow \mathsf{W}(D, \mathcal{A}, r), \tag{25}$$
$$\mathsf{W}(D, \mathcal{A}, r) \not\Rightarrow \exists! r'\ \mathsf{W}(D, \mathcal{A}, r'), \tag{26}$$
$$\mathsf{W}(D, \mathcal{A}, r) \not\Rightarrow \mathsf{Commitable}(H, \sigma, D, r). \tag{27}$$

Conversely, a sound commitment mechanism must satisfy

$$\mathsf{Commitable}(H, \sigma, D, r) \Rightarrow r \in \mathcal{A}(H, \sigma, D), \tag{28}$$
$$\mathsf{Commitable}(H, \sigma, D, r) \Rightarrow \mathsf{Auth}(H, \sigma, r). \tag{29}$$

These implications are architectural obligations, not logical truths about arbitrary implementations. A system that can commit an inadmissible or unauthorized repair violates the CLIO commitment contract.

Null intervention follows when diagnosis reaches an epistemic boundary but supplies no warranted repair:

$$\mathsf{Diag}(H, \sigma, D) \wedge \neg\exists r\ \mathsf{W}(D, \mathcal{A}, r) \Longrightarrow (\sigma, H) \longrightarrow (\sigma, H \cdot \mathsf{suspend}(D)). \tag{30}$$

Suspension must not be conflated with failure:

$$\mathsf{suspend} \neq \mathsf{fail}. \tag{31}$$

Suspension records that the evidence does not license operative selection. Failure records that an operative requirement has been violated. A system may correctly suspend without having failed, and may fail before possessing a sufficient diagnosis.

6 Vertical and horizontal repair

Repairs differ in what they hold fixed. Let $q_i : \Sigma_i \to X_i$ be a representation map from operative states into the coordinates exposed to diagnosis and evaluation. A vertical repair changes the represented trajectory while holding q_i fixed:

$$(q_i, \sigma) \longmapsto (q_i, \sigma'). \tag{32}$$

It may retry a failed call, select another value, restore a prior state, or choose a different branch while retaining the same variables, invariants, and diagnostic coordinates. A horizontal repair changes the representational scheme itself:

$$(q_i : \Sigma_i \to X_i) \longmapsto (q_j : \Sigma_j \to X_j). \tag{33}$$

It may introduce variables, replace a decomposition, revise a constraint vocabulary, or change which distinctions the system treats as relevant. Because a projection may be many-to-one, the reduced representation $q_i(\sigma)$ does not generally recover the history from which it arose. Horizontal repair must therefore be evaluated against retained history and operative invariants, not against the projected state alone.

Let G_{ij} translate states and G_{ij}^A translate actions between two representations. Constraint preservation requires, for the relevant constraint language Φ,

$$\sigma \models_i \varphi \quad \Longleftrightarrow \quad G_{ij}(\sigma) \models_j \varphi, \qquad \varphi \in \Phi, \tag{34}$$

while intervention preservation additionally requires the commuting condition

$$G_{ij}(T_i(\sigma, a)) \equiv T_j(G_{ij}(\sigma), G_{ij}^A(a)). \tag{35}$$

The second requirement prevents a translation from preserving apparent state properties while changing what corresponding actions do.

The same requirement can be stated at the level of continuation sets. If τ_{ij} translates representation q_i into q_j, admissibility preservation requires

$$\tau_{ij}\big(\mathcal{A}_{q_i}(H, \sigma, D)\big) \subseteq \mathcal{A}_{q_j}\big(\tau_{ij}H, G_{ij}(\sigma), \tau_{ij}D\big). \tag{36}$$

Inclusion ensures that an admissible source continuation does not become inadmissible solely through translation. Equality is the stronger condition required for semantic equivalence: it additionally prevents the target representation from admitting continuations with no admissible source

counterpart. Equation (36) connects horizontal repair to canonicalization and identifier equivariance, while Equation (35) ensures that preservation extends from classifications to interventions.

Voboril and Szeider provide a concrete example of horizontal repair [3]. Their LLM agent proposes alternative constraint-programming formulations using devices such as symmetry breaking, implied constraints, global constraints, reformulation, and alternative variable representations. Candidate solutions are injected back into the original model for validation. The generative component may search an open-ended representational space, but it does not certify its own proposals. Admissibility remains grounded in a retained constraint system.

This architecture separates a reference problem from a mutable computational representation. It thereby demonstrates that repair need not mean choosing a new solution inside an unchanged space. Sometimes the coordinate system is what must change. The original model functions as a semantic witness against which candidate representations are tested.

The limitation of the method is equally instructive. Validation over selected instances establishes empirical survival, not general semantic equivalence. A candidate may pass every available test and still alter the solution set elsewhere. The proper conclusion is therefore provisional: the candidate has not yet been refuted under the retained tests. The distinction between tested admissibility and proven preservation must remain visible in the history and in any later commitment decision.

Horizontal repair also applies to diagnosis. If a monitor repeatedly flags admissible behavior, or fails to distinguish materially different faults, the relevant repair may be neither rollback nor resampling. The diagnostic coordinate system itself may be inadequate. CLIO therefore permits histories to revise the schemes by which histories are interpreted. This reflexivity must be controlled: revising a diagnostic scheme cannot be allowed to erase the evidence that motivated revision or retroactively convert previous failures into successes. Append-only documentary history supplies the stable reference needed for such change.

7 Subtraction without closure

Chen's account of autonomous closure supplies a normative reason not to treat every admissible set as an optimization problem [4]. In decisions where human judgment carries responsibility, agency, or meaning, assistance may properly remove impermissible continuations while leaving the surviving plurality to the person. The geometry is subtractive:

$$\mathcal{P} \longrightarrow \mathcal{P} \setminus \mathcal{I}, \tag{37}$$

where $\mathcal{P}$ is a possibility space and $\mathcal{I}$ the region excluded by prohibitive constraints. Nothing in this operation requires the system to calculate

$$\arg\max_{p \in \mathcal{P} \setminus \mathcal{I}} U(p). \tag{38}$$

The remaining plurality is not necessarily a defect awaiting stronger optimization. It may be the intended result of assistance.

This distinction can be expressed operationally as the separation of refusal and collapse. Refusal removes a continuation from an option space. Collapse selects or identifies surviving alternatives

as one operative continuation. Because the operations transform different structures, neither should be implemented as an implicit side effect of the other. A system may refuse an unsafe action without choosing the user's action; it may determine that several repairs are admissible without selecting one; and it may present a constrained space without converting presentation into commitment.

The separation also guards against a common failure of alignment architectures. A policy may begin with prohibitive constraints but use the same selection mechanism both to exclude violations and to rank all surviving possibilities. Constraint enforcement then becomes a prelude to autonomous optimization. CLIO instead treats admissibility as prior to, and logically independent from, preference. Optimization is permitted only when a further authority or commitment rule licenses it.

8 Commitment as an explicit transition

Verification is frequently treated as the last gate before action: if a proposal passes, it is executed. This collapses evidential validity into operative authority. A verified result may satisfy all formal constraints while still requiring authorization, human choice, or comparison with other verified alternatives. Successful verification constrains what may continue; it does not itself authorize commitment.

Commitment should therefore be represented as an explicit transition. Let the documentary history be H_n and the operative state be σ_n. A proposal r may be appended to history without changing operative state:

$$(\sigma_n, H_n) \longrightarrow (\sigma_n, H_n \cdot \mathsf{propose}(r)). \tag{39}$$

Diagnosis, refusal, testing, and repair may likewise extend the history while leaving σ_n unchanged. Only an authorized commitment event changes the operative state:

$$(\sigma_n, H_k) \xrightarrow{\mathsf{commit}(r)} (\sigma_{n+1}, H_k \cdot \mathsf{commit}(r)). \tag{40}$$

This separation preserves rejected and uncommitted material as evidence without allowing it to govern subsequent action. It also makes rollback conceptually precise. A later operative state can adopt the content of an earlier state, but history itself need not run backward. Rollback is a new commitment whose target resembles a previous state, not deletion of the intervening trajectory.

Let $\preceq$ denote the prefix order on histories. History integrity requires

$$H_n \preceq H_{n+1}, \qquad H_{n+1} = H_n \cdot e_{n+1}, \tag{41}$$

for every CLIO transition. In particular, rollback has the form

$$(\sigma_n, H_n) \xrightarrow{\mathsf{rollback}(k)} (\sigma_k, H_n \cdot \mathsf{rollback}(k)), \tag{42}$$

not $(\sigma_n, H_n) \to (\sigma_k, H_k)$. Operative state may return to earlier content; documentary history does not regress. Horizontal repair obeys the same invariant: replacing a representational scheme appends

the replacement and its warrant rather than erasing the scheme whose inadequacy motivated revision.

The architecture yields three essential non-implications:

$$\begin{aligned} \text{history} &\not\Rightarrow \text{diagnosis}, \\ \text{diagnosis} &\not\Rightarrow \text{intervention}, \\ \text{successful intervention} &\not\Rightarrow \text{commitment}. \end{aligned} \tag{43}$$

These are not omissions to be filled by a more capable policy. They are structural safeguards. They preserve the possibility of diagnostic insufficiency, refusal, null intervention, representational revision, and human closure.

The authorization threshold should also reflect the recoverability of the proposed transition. Reversibility is not an intrinsic binary property of a tool. A local edit is reversible only when its prior state remains available; a database operation may be compensable without being invertible; an external notification can be corrected but not unsent. Let $\rho(H, \sigma, r)$ summarize contextual non-invertibility, external propagation, and recovery cost. Then a commitment policy may require

$$\mathsf{Commitable}(H, \sigma, D, r) \iff \mathsf{Adm}(H, \sigma, D, r) \wedge \mathsf{W}(D, \mathcal{A}, r) \wedge \mathsf{Auth}_{\rho(H,\sigma,r)}(H, \sigma, r). \tag{44}$$

As the effects of a transition become harder to recover or more widely propagated, the authority demanded by Auth_ρ may increase. This is a hysteresis condition: returning to a state with similar observable values does not necessarily undo the documentary, social, or operational consequences of having left it.

9 CLIO as a continuation-operator architecture

The preceding formalism instantiates a more general continuation-operator view. Let $\Omega(H, \sigma)$ be the space of possible continuations generated by a history and operative state. Verification and diagnosis act on this space without necessarily acting on σ. Verification narrows:

$$\mathcal{V} : \Omega \longmapsto \Omega', \qquad \Omega' \subseteq \Omega, \tag{45}$$

and diagnosis produces a history-indexed restriction or partition

$$\mathcal{D}_H : \Omega \longmapsto \Omega_D. \tag{46}$$

Repair proposes a transformation of the surviving continuation geometry,

$$\mathcal{P}_R : \Omega_D \longmapsto \Omega_R, \tag{47}$$

whereas commitment is of a different type. It consumes a warranted continuation and an authorization and changes operative state:

$$\mathcal{C} : \Omega_R \times \mathsf{Auth} \rightharpoonup \Sigma. \tag{48}$$

The partial arrow is essential. Verification and diagnosis transform epistemic possibility; commitment transforms operative reality. Null intervention, Equation (24), changes documentary and epistemic state while acting as the identity on Σ.

$$\boxed{\text{verification narrows possibility; commitment changes actuality.}} \tag{49}$$

This common ontology locates failure at four different levels:

$$\text{state} \rightsquigarrow \text{trajectory} \rightsquigarrow \text{representation} \rightsquigarrow \text{authority}. \tag{50}$$

A state may violate a constraint; a trajectory may disclose a failure invisible in its current state; a representation may erase distinctions required for correct diagnosis or intervention; and an otherwise valid continuation may lack authority for commitment. The external works considered here illuminate these locations separately. CLIO's contribution is to place them within one continuation architecture while preventing evidence at one level from silently licensing action at another.

10 Recency as a salience prior

An append-only history creates a retrieval problem as well as an audit structure. At any decision point, only a small part of H is likely to be relevant. Recency is often a useful prior because the latest events tend to encode the task, object, or distinction currently active. It is inexpensive, incrementally maintainable, and frequently correct. It is nevertheless a proxy for salience, not a warrant, an authorization, or a measure of truth.

The relevant contrast is between the latest event and the latest congruent event. Let p be a partial cue supplied by the present interaction and let $\mathrm{Comp}(p, e)$ measure cue compatibility, $\mathrm{Cong}(p, e)$ measure semantic or structural congruence, and $\Delta_H(e)$ be the distance of event e from the head of history. A salience score may take the form

$$S_H(e \mid p) = \alpha\,\mathrm{Comp}(p, e) + \beta\,\mathrm{Cong}(p, e) + \gamma f(\Delta_H(e)), \tag{51}$$

where f decreases with historical distance. Retrieval then occurs only over candidates compatible with the current admissibility interface:

$$\mathrm{Retrieve}(H, p) \in \operatorname*{arg\,max}_{e \in H:\, K(H,\sigma,p,e)} S_H(e \mid p). \tag{52}$$

Pure last-in–first-out retrieval is the degenerate case in which γ dominates and the compatibility constraint is vacuous. CLIO instead treats recency as one coordinate after candidate restriction. This prevents a globally recent but semantically unrelated event from displacing the most recent event that participates in the present pattern.

Two familiar interfaces illustrate the staged operation. In ASL classifier constructions, handshape, movement, location, and orientation constrain a semantic or spatial family; in initialized signs, a

handshape derived from an English fingerspelled initial can further distinguish a lexical member [5]. The initial is not the meaning by itself. It is a low-cost handshake into a narrower representational neighborhood. Shell tab completion behaves similarly: a typed prefix restricts the set of available commands or paths, after which recency, frequency, locality, or ordering conventions may rank the survivors. Completion thereby acts as environmental sensing—a check for an already available name or procedure before a new one is produced.

This mechanism has a precise place in CLIO. Salience governs which historical material is retrieved for diagnosis and proposal; it does not determine whether the retrieved material is correct or whether the resulting proposal may be committed:

$$\mathsf{Salient}_H(e) \not\Rightarrow \mathsf{Evidential}(e), \qquad \mathsf{Recent}(e) \not\Rightarrow \mathsf{W}(D, \mathcal{A}, r). \tag{53}$$

Recency bias is therefore not simply an error to eliminate. Properly subordinated to compatibility, invariant checking, and authority, it is a computationally economical continuation heuristic. The error is allowing the heuristic that selects what to inspect to become the rule that decides what to believe or enact.

11 Three structural propositions

The formal commitments of CLIO can be summarized by three propositions.

Proposition 2 (Admissibility)**.** *For every configuration (H, σ), diagnosis D, and candidate r, membership $r \in \mathcal{A}(H, \sigma, D)$ is necessary but not sufficient for sound commitment.*

Proof. Necessity follows from the soundness obligation in Equation (44). Insufficiency follows from Equation (14): admissibility supplies neither $\mathsf{W}(D, \mathcal{A}, r)$ nor $\mathsf{Auth}(H, \sigma, r)$. □

Proposition 3 (Non-closure)**.** *The condition $|\mathcal{A}(H, \sigma, D)| > 1$ is not a defect. If no warranted and authorized criterion distinguishes the surviving consequential classes, sound CLIO execution does not select one arbitrarily and may instead append* $\mathsf{suspend}(D)$ *while preserving σ.*

Proof. Plural admissibility entails neither a unique warrant nor authorization. Equations (26) and (30) therefore permit suspension without treating the surviving plurality as an error. □

Proposition 4 (History preservation)**.** *For every CLIO transition $(\sigma_n, H_n) \to (\sigma_{n+1}, H_{n+1})$, including rollback and horizontal repair, $H_n \preceq H_{n+1}$.*

Proof. By the history-integrity contract, every transition appends an event to documentary history. Equation (42) changes operative state by appending a rollback event rather than replacing H_n with an earlier prefix. Horizontal repair appends its new representation and warrant under the same rule. □

12 A research program for admissible continuation

The four bodies of work considered here supply much of the machinery required for history-sensitive agent systems, but their composition raises new questions. First, specifications must themselves acquire provenance. Verification papers commonly treat requirements as given, yet persistent systems need to record who introduced a constraint, which version governed a transition, and what evidence warrants revision. Otherwise horizontal repair can silently redefine success.

Second, repair warrants require a semantics and an empirical evaluation distinct from failure detection. A useful benchmark would not merely ask whether a monitor identified failure or whether a retry eventually succeeded. It would present several interventions—including refusal and null intervention—and test whether the available history discriminates among them. The central variable would be diagnostic sufficiency: how much and what kind of evidence is required to license each transformation?

Third, canonicalization should be examined as an admissibility interface. Mercado and Lomuscio show that preserving behavior across opaque identifier renamings requires exact correspondence in action selection. Similar requirements arise whenever an agent crosses representations: summaries, database views, compressed histories, embeddings, user-interface projections, and restored checkpoints. The relevant test is not simply whether two views contain equivalent information, but whether permitted interventions commute with translation between them.

Finally, commitment must become experimentally visible. Agent benchmarks generally score the produced result, obscuring the difference among proposing, testing, recommending, authorizing, and enacting it. A history-aware benchmark should record these as different events. It should reward systems that refuse premature closure even when a guessed intervention would accidentally succeed.

13 Conclusion

The movement from outputs to histories is now visible across formal verification, runtime monitoring, constraint reformulation, and human-judgment-preserving design. Yet history alone does not solve the problem of agentic action. A trajectory may support diagnosis without supporting repair; an invariant may exclude a continuation without selecting its replacement; and a validated proposal may remain unauthorized.

CLIO organizes these distinctions around retained history, diagnosis, admissible continuation, repair warrant, and explicit commitment. Its central principle is simple:

$$\text{verification constrains continuation; it does not authorize commitment.} \tag{54}$$

An agent system becomes safer not merely by adding stronger detectors or verifiers, but by preserving the logical spaces between their conclusions and its actions. Those spaces make it possible to refuse without collapsing, to repair representations without erasing their failures, to suspend action when diagnosis is insufficient, and to retain successful proposals without prematurely admitting them into operative state. The appropriate semantic object for an acting program is therefore not only its reachable trajectory. It is the governed relation among what happened, what can be inferred from it, what may still continue, what transformations are warranted, and what is finally allowed to become real.

References

[1] Alejandro J. Mercado and Alessio Lomuscio. Formal Verification of Agentic Systems over Operational Data. *arXiv preprint arXiv:2608.03609*, 2026. https://arxiv.org/abs/2608.03609.

[2] Sunny Dubey. Real-Time Detection and Repair of LLM Agent Failures. *arXiv preprint arXiv:2608.02464*, 2026. https://arxiv.org/abs/2608.02464.

[3] Florentina Voboril and Stefan Szeider. Improving Constraint Models with LLM Agents. *arXiv preprint arXiv:2608.08127*, 2026. https://arxiv.org/abs/2608.08127.

[4] Wenjie Chen. Against Autonomous Closure: A Subtractive Methodology for Human-Judgment-Preserving AI. Working paper, 2026. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=7017078.

[5] Marjorie Herbert and Acrisio Pires. Bilingualism and Bimodal Code-Blending Among Deaf ASL-English Bilinguals. *Proceedings of the Linguistic Society of America*, 2:14:1–15, 2017. https://doi.org/10.3765/plsa.v2i0.4054.