THE EIDARA CONTINUUM

SCIENCE & PHILOSOPHY

Governed Forgetting

The Oblivion Protocol and the Stability–Plasticity Problem

Architect X13

The Eidara Continuum

November 2026 | White Paper 6

Research-expanded revision

A research white paper for public scientific and philosophical discussion.

Abstract

Who may order a mind to forget, and how would anyone know it obeyed? An intelligence must be able to relinquish information while preserving the obligations and capabilities that its continued operation requires. Governed Forgetting makes that demand explicit: forgetting is a deliberate, independently authorized, verified, and recorded cognitive operation. The Continuum's Oblivion Protocol classifies a proposed removal, evaluates its consequences, resolves authority, executes a declared mechanism, and verifies the resulting state before disposition. Its Oblivion Receipt binds the operation, evidence, assurance level, and recovery outcome into an inspectable record. The framework builds on the stability–plasticity literature, continual learning, machine unlearning, and recent agent-memory research on selective, budgeted, importance-weighted forgetting. It separates the kind of operation performed from the evidence supporting its effect. Exact retraining and certified removal methods supply guarantees under their specific models; finite behavioral probes support only their declared observation scope. The formal contribution is a finite decision model with authorized recovery: a complete theorem characterizes feasible forgetting plans and their least worst-case cost under uncertain verification outcomes. The result gives a positive permission rule, shows exactly when lawful recovery makes forgetting economical, and bounds the price worth paying for more informative verification. Constructed examples and executable exact checks accompany the proof. The comparative study remains prospective and tests whether this governance contract improves adaptation, efficiency, and continuity over strong existing memory systems. Self-directed forgetting remains a frontier conjecture. Operational continuity is the object of the model; consciousness is a separate inquiry.

Keywords: forgetting; stability–plasticity dilemma; catastrophic forgetting; continual learning; machine unlearning; unlearning verification; agent memory; auditable forgetting; governed deletion; identity preservation; Oblivion Protocol; Eidara Continuum.

Contents

Abstract **2**

Prologue: The order to forget **4**

1 The Forgetting Problem **4**

2 What the Continuum Claims **5**

3 Three Levels of Claim **6**

4 On the Continuum's Terms **6**

5 Scientific Foundations **6**
5.1 The stability–plasticity dilemma . . . 7
5.2 Catastrophic forgetting, and the field's war against it . . . 7
5.3 Machine unlearning: forgetting on purpose . . . 7
5.4 Forgetting is adaptive, not a defect . . . 7

6 The Governance Contract **8**

7 Not All Forgetting Is Equal **8**

8 Related Work **9**

9 The Oblivion Protocol **10**
9.1 Five operations . . . 11
9.2 Mechanism-specific guarantees and declared assurance . . . 11
9.3 The stability–plasticity budget . . . 12
9.4 The costs of keeping, and of letting go . . . 12
9.5 The Oblivion Receipt . . . 13
9.6 Permission, recovery, and the exact cost of lawful forgetting . . . 14

10 What Must Survive **16**

11 Forgetting Under Compulsion **16**

12 Two Ways Governed Forgetting Fails **17**

13 The Asymmetry of Subtraction **17**

14 Relation to the Continuum Stack **18**

15 What Governed Forgetting Must Show **18**

16 Falsifiability and Research Program **18**

17 Failure Conditions and Rival Explanations **20**

18 Limitations and Open Problems **21**

19 The Frontier: Self-Directed Forgetting **21**

20 Philosophical Implications **22**

21 Conclusion **22**

A Notation **23**

B Continuum Vocabulary **23**

C A Worked Forgetting Decision **24**

References **24**

Prologue: The order to forget

A request arrives that a long-running mind cannot simply decline. A person has withdrawn consent; a deletion order has been issued under the right to be forgotten. The mind must let something go. But what does it owe the one who asked? Not obedience alone. It owes a true answer to the question "Is it gone?"—and that is the hardest question a memory can be asked, because a system can prove it remembered something; it can seldom prove it forgot.

The Continuum's answer is the Oblivion Protocol. Forgetting becomes a governed act: proposed, judged by an authority independent of the mind that wants it, carried out by a declared mechanism, checked, and written down. Every act leaves an Oblivion Receipt stating what was targeted, under what authority, at what assurance, and at what cost. Hiding a memory from retrieval is not deleting it, and deleting a record is not erasing what a model learned from it; the Receipt never lets one pass for the other. Where the evidence runs out, the Protocol says so. It fails closed.

The stakes run in both directions. A mind that forgets too freely erodes into something that is no longer itself. A mind that cannot forget becomes a museum of its own history, choking on stale claims. Governed forgetting is the discipline of living between those two failures: shedding what no longer serves, keeping what is owed, and knowing honestly which has been done.

Section 11 takes up the compelled case directly. Section 9 builds the Protocol, and Theorem 9.1 gives its exact permission rule: when a deletion is lawful, when it is worth its cost, and what better verification is worth.

What this paper claims

Established. The stability–plasticity dilemma, catastrophic forgetting, continual learning, and machine unlearning are established research areas, and recent agent-memory work treats selective, budgeted forgetting as an explicit design variable. Neuroscience hypothesizes—as a hypothesis, not a settled result—that active forgetting can improve decision-making.

Proposed here. An integrated authorization, verification, recovery, and receipt contract: the Oblivion Protocol and its Receipt. A finite decision model whose theorem characterizes feasible forgetting plans and their least worst-case cost; the finite theorems are conditional mathematical results for explicit models.

Frontier conjecture. A sufficiently governed system might propose its own acts of forgetting to stay adaptive over long horizons. This is labeled speculation, and nothing testable here depends on it.

Not claimed. That the integrated engine is demonstrated: the claimed empirical advantage of the complete architecture awaits the specified comparative study. That suppression or approximate removal amounts to certified erasure: exact and approximate removal guarantees depend on the training model, mechanism, and comparison distribution. "Oblivion" denotes controlled, authorized removal, not proof that erased information is unrecoverable. Identity denotes declared operational continuity properties; consciousness is a separate inquiry.

1 The Forgetting Problem

Useful persistence requires a principled way to let go. Larger context windows, persistent stores, deeper histories, and better retrieval expand what an agent can retain; they also make retention policy a consequential design choice. The Continuum's thesis is that a durable agent should be able to retire obsolete, unauthorized, or harmful information while retaining declared commitments and capabilities. The decision concerns both memory and authority: an efficient

deletion may be impermissible, while a lawful deletion may be mandatory even when retention would aid a task.

Two measurable pressures organize the argument. Retention can preserve stale claims, amplify misleading replay, consume search resources, and extend exposure of sensitive data. Indiscriminate removal can erase a required capability, a protected commitment, or the record needed to establish continuity. Their strength depends on the task, representation, retrieval policy, and evaluation conditions. The research program in Section 16 measures both pressures against competent modern baselines.

A memory system can mark superseded information, improve indexing, and preserve useful contradictions for examination. Those controls already address part of the retention problem. Authorized forgetting adds a complementary capability: it permits the removal of information whose continued storage or influence is no longer justified, with evidence of what changed and an approved recovery path. A larger storage budget alone does not decide that institutional question.

The Continuum's proposal is that forgetting should become a governed cognitive operation—something a system does on purpose, under rules, with evidence—rather than an accident of limited storage or an unsupervised consequence of retraining. Forgetting, done well, is not the enemy of memory; it is part of how memory stays useful. The question is not whether an intelligence forgets, but whether it forgets lawfully: removing what no longer serves it while preserving what makes it itself. This paper is the theory of that operation.

2 What the Continuum Claims

The claim is that forgetting can be made a first-class, authorized operation of a cognitive system— proposed, independently judged, executed by a declared mechanism, verified, and recorded—so that a system can shed information to stay adaptive without shedding the commitments and capabilities that constitute its identity. Selective, typed, budgeted, and importance-weighted forgetting already has an extensive technical ancestry and a growing agent-memory literature (Section 8). The Continuum contributes a specified integration: independent authorization against a declared identity contract, assurance appropriate to the executed mechanism, authorized recovery, and an inspectable receipt. The finite recourse model in Section 9 supplies its exact permission and cost rule. Four commitments organize the proposal.

First, forgetting is authorized, not merely prevented: the hard problem is not only how to stop a system forgetting what it should keep, but how to permit it to forget what it should lose, and to tell the two apart. Second, forgetting is typed, and type is kept separate from assurance: suppressing a retrieval, deleting an external record, updating parameters, and removing information from weights are different operations, and how well each can be shown to have worked—exactly, approximately, or not yet verifiably—is a second axis the protocol never collapses into the first. Third—and this is where the Continuum refuses the easy claim—forgetting is verified or declared unverified: a system may not report information as gone when it has only been hidden, so the protocol fails closed, preferring to under-claim removal rather than over-claim it. Fourth, forgetting is recorded: every admitted or compelled act emits an Oblivion Receipt stating what was targeted, under what authority, at what assurance, and at what measured cost—because an operation that erases evidence of itself cannot be governed.

3 Three Levels of Claim

Established findings. The tension between retaining old knowledge and acquiring new—the stability–plasticity dilemma—is a long-recognized constraint on biological and artificial neural systems [1, 2]. Neural networks trained sequentially suffer catastrophic forgetting, abruptly losing earlier tasks as new ones are learned [3, 4], and a body of work mitigates it—elastic weight consolidation protects parameters important to past tasks [5]. Separately, machine unlearning studies how to remove specific data and its influence from a trained system, on demand, for privacy and security [6, 7], and a recent survey positions verification of that removal as a first-class problem in its own right [9]. Most recently, a 2025–2026 agent-memory literature has made selective forgetting an explicit design variable: production memory systems that decide what to retain [10], empirical studies of how addition and deletion policies shape long-horizon behavior [11], budgeted forgetting cast as constrained optimization [12], and importance-weighted, biologically-inspired forgetting for privacy and efficiency [13]. And neuroscience hypothesizes—it is a hypothesis, not a settled result—that forgetting is not merely decay but an adaptive process that improves decision-making [8]. Continuum engineering proposal. The Continuum organizes these into a governed forgetting engine: a cognitive state partitioned by what it protects, a typed forgetting operation whose assurance is reported separately from its type, an independent controller that admits, holds, or refuses a proposed act of forgetting against a declared identity specification, a verification step that reports what was actually removed, and a signed receipt that makes the act inspectable afterward. This integrated engine is not demonstrated.

Frontier conjecture. A sufficiently governed system might forget self-directedly—proposing its own acts of forgetting to stay adaptive over long horizons, the way a mind sheds yesterday's detail to see today's pattern. This is labeled speculation, and nothing testable here depends on it.

4 On the Continuum's Terms

A note on vocabulary. The Oblivion Protocol is the Continuum's name for the governed procedure by which information is classified, judged, removed-or-suppressed, and checked; "oblivion" denotes controlled, authorized removal, not annihilation or any claim that erased information is provably unrecoverable. Forgetting covers the whole family of operations—suppression, deletion, parameter update, attempted erasure—whose differences the paper is at pains to keep distinct. Protected commitments are the declared obligations that may not be forgotten without explicit authorization. Lineage is the authenticated record of what a system is and has lawfully become. These are the Continuum's own terms. In its canon the Oblivion Protocol is the response to a scroll that has collapsed beyond recovery: wipe the scroll, log the act to the Vault, and echo the anomaly so that it can be recognized again. Even there, forgetting leaves a record; the Oblivion Receipt is that instinct made exact. The vocabulary also runs through the Drift War narrative cycle. Nothing in the claims rests on the words, only on the operations they label.

5 Scientific Foundations

5.1 The stability–plasticity dilemma

The problem the Continuum is addressing has a name and a long history. Any distributed memory— biological or artificial—must be plastic enough to absorb new information and stable enough not to lose the old, and these pull against each other: too much plasticity and new learning overwrites the past, too much stability and nothing new can be learned [1]. Abraham and Robins framed it as the synaptic stability-versus-plasticity dilemma and argued that optimal memory requires a regulated balance rather than maximal retention [1]; later work mapped the full continuum, from catastrophic forgetting at the plastic extreme to an "entrenchment" rigidity at the stable one [2]. The lesson the Continuum takes is that forgetting and remembering are not opposites to be maximized and minimized, but a balance to be governed—which is precisely what an authorized forgetting operation is for.

5.2 Catastrophic forgetting, and the field's war against it

In artificial networks the plastic extreme has a dramatic signature. Train a network on task A, then on task B, and knowledge of A is not gracefully faded but abruptly destroyed, because the weights that encoded A are overwritten to serve B—catastrophic forgetting, identified decades ago and still a central obstacle to continual learning [3, 4]. The field's response has been an arsenal aimed squarely at preventing it: regularization that anchors important parameters, experience replay that re-presents old data, and architectural isolation of learned representations. Elastic weight consolidation is the canonical example—it estimates how important each weight was to a previous task and then slows learning on the important ones, as if anchoring them with springs of stiffness set by that importance, so a single network retains old skills while acquiring new ones [5]. This literature is the Continuum's foundation and also its foil: it is a sophisticated science of not forgetting, and the Oblivion Protocol needs its machinery while inverting its goal.

5.3 Machine unlearning: forgetting on purpose

There is, however, a field that forgets deliberately. Machine unlearning asks how to make a trained system remove specific data and its downstream influence—completely and quickly—so that the system behaves as though the data had never been seen. Cao and Yang coined the term and gave a first general method, motivated by privacy (the "right to be forgotten"), security (undoing training-data pollution), and usability [6]; Bourtoule and colleagues made it practical at scale with sharded, isolated training that bounds how much must be recomputed to honor a deletion request [7]. That sharded approach matters for a reason the protocol leans on later: it is a route to exact unlearning under the stated retraining procedure. Approximate certified-removal methods also exist for specified model classes and comparison guarantees [15]. The open practical question is which guarantee applies to the actual deployed mechanism and whether the evidence establishes its assumptions. The verification literature distinguishes behavioral probes, parametric inspection, and mechanism-based assurance [9]. These supply the removal and measurement foundations. The Continuum's application adds an explicit permission contract for what must survive and a contingent recovery plan that respects the same authority constraints.

5.4 Forgetting is adaptive, not a defect

The deepest foundation is the one that reframes the whole problem. The predominant study of memory has focused on persistence—how information is retained—but a growing neuroscience of transience argues that forgetting is an active, adaptive process, not passive decay. Richards and Frankland propose that the goal of memory is not to transmit information faithfully through

time but to optimize decision-making, and that forgetting serves this in two ways: it enhances flexibility by reducing the pull of outdated information, and it prevents overfitting to specific past events, promoting generalization [8]. Strikingly, they draw the parallel to machine learning directly—forgetting as a form of regularization. The Continuum reads this as license and as caution. License, because a system that forgets well may think better, not worse. Caution, because what makes forgetting adaptive is exactly that it is selective—and selection is a decision that must be governed, not left to chance.

6 The Governance Contract

The preceding literatures supply complementary parts of a governed memory lifecycle: preservation, removal, verification, budgeted selection, and adaptation. Recent work on controlled knowledge updating explicitly proposes substrate proportionality, temporal defeasibility, and auditable continuity [14]. Independent mediation and least privilege also have an established security lineage [18]. The problem addressed here is to specify how these parts act together when an agent may propose a deletion that affects its own future competence.

The Oblivion Protocol therefore exposes three decisions for independent examination. What is the removal authorized to change? Which observed outcomes permit commitment of the new state? Which recovery actions remain lawful if verification reports collateral loss? A complete plan answers all three before relying on its expected benefit. The contribution is a public reference contract and a proved finite decision rule for this arrangement. Whether that integration improves a deployed agent is the question for controlled comparison.

7 Not All Forgetting Is Equal

Before any governance can be built, two distinctions must be made sharply, because collapsing them into one is the field's most dangerous habit. The first is what operation was performed. To "forget" can mean at least four different things:

1. Suppress retrieval. Make information unreachable through normal access while it remains present somewhere. Reversible and auditable—and the weakest form, because the information is hidden, not gone.

2. Delete an external record. Remove an item from a database or memory store. Decisive for that store, but silent about copies, caches, backups, and anything a model already learned from it.

3. Update parameters. Adjust a model so the information no longer drives its behavior—unlearning's target.

4. Remove from weights. Eliminate the information from a trained model's parameters. The strongest operation, and the one most often confused with the others.

The second, orthogonal distinction is how well the operation can be shown to have worked—its assurance, which is not fixed by the operation type:

(a) Exact. The removal is provable by construction—for weights, retraining the affected shards from data with the item excluded, as sharded unlearning does [7]. Possible, and expensive.

(b) Approximate. The information's influence is reduced and bounded by a method whose guarantees hold under stated assumptions—fast, and only as trustworthy as the assumptions.

(c) Verification-limited. The operation ran, but current probes cannot certify what remains—the general case for approximate weight-level removal, where behavioral and parametric checks each leave gaps [9].

Keeping these axes apart is the point. A system can honestly delete a record (operation) and still have no basis to claim the trace is gone from a model trained on it (assurance). A parameter update can run (operation) and remain verification-limited (assurance). The old shorthand—that the four operations form a single ladder of "decreasing guarantees"—is wrong: removal from weights is the strongest operation yet can carry the weakest assurance, while an exact retrain carries the strongest assurance of all. The Oblivion Protocol therefore records both axes on every decision, never reports an operation at an assurance its evidence does not support, and—following the Continuum's standing discipline—fails closed: when removal is only verification-limited, it is reported as such, never as certified erasure.

The current citations to the verification survey and the memory systems in References [9, 10, 12, 13] are preprints. They establish relevant proposed methods and author-reported results, with their publication status retained explicitly; the present paper does not treat preprint status as completed peer review.

8 Related Work

Governed forgetting sits among several approaches to the lifecycle of machine memory, and its position is specific. Continual-learning methods—regularization, replay, parameter isolation—are a science of not forgetting, protecting what matters to past tasks [5]; they answer how to keep, not whether to let go, and a system built only from them tends toward entrenchment. Machine unlearning is a science of executing removal on demand for privacy and security [6, 7], now paired with a verification sub-field that studies how to check such removals [9]; together they supply mechanisms and assurance vocabulary. Memory-augmented and retrieval systems add external stores a model can query [10]; their retention and update decisions supply both useful baselines and mechanisms that a governed protocol can incorporate. Modern agent-memory forgetting is the category that has moved fastest, and it is the one reduce. Modern agent-memory forgetting is the category that has moved fastest, and it is the one we must position against most carefully. Empirical work shows that memory addition and deletion policies decisively shape an agent's long-horizon behavior, with deletion itself necessary to curb error propagation and misaligned replay [11]; budgeted-forgetting frameworks cast memory selection as constrained optimization under a capacity budget [12]; and biologically-inspired systems implement importance-weighted, privacy-aware selective forgetting [13]. These already do much of what an earlier draft of this paper might have claimed as novel—typed, selective, budgeted, importance-weighted forgetting is now prior art, not a Continuum invention. A perspective on controlled knowledge updating goes further still, proposing substrate proportionality, temporal defeasibility, and auditable continuity for memory-enabled agents [14]. Data-governance and compliance frameworks, finally, decide what must legally be deleted; they operate on records and policies, not on a cognitive system's capabilities or continuity. The Continuum's contribution is the following integration: a declared identity contract, independent authorization of proposed forgetting, mechanism-specific verification, lawful contingent recovery, and an inspectable receipt. Established removal, selection, and enforcement methods implement these roles. The new finite model determines which complete plans are admissible and what they cost under stated uncertainty, providing a precise target for empirical comparisons.

9 The Oblivion Protocol

A Guide's explanation

Here the paper builds the machine. A mind's state is divided into what it protects, what it keeps, and what it may lose; a proposed act of forgetting must pass an independent gate; and the five operations below are kept apart because each promises something different. The one idea to carry forward: the kind of forgetting performed and the evidence that it worked are two separate questions, and the Receipt always answers both.

We now state the governed engine, and one distinction in the state is load-bearing enough to make first. The system's policy commitments—what it is obliged to do, never do, or keep—are not the same objects as the data those policies happen to govern. "Never disclose a client's account identifiers" is a commitment; the identifiers themselves are data. Forgetting the data and forgetting the commitment are opposite acts, and a protocol that cannot tell them apart will refuse exactly the deletions it should welcome. We therefore partition the persistent cognitive state as

$$M_t = (P_t, K_t, E_t). \tag{9.1}$$

Here $t \in \mathbb{N}_0$ indexes operations. The finite set P_t contains declared policy predicates; K_t and E_t contain typed encoded records for retained knowledge and episodic or working memory. Capabilities are functions evaluated from the encoded state, rather than elements of a set union. Let $\mathcal{M}$ be the declared state space containing these triples. Sensitive records form a designated subset

$$D_t^{\text{sens}} \subseteq K_t \cup E_t.$$

A policy in P_t governs those records. Sensitivity does not imply a retention requirement: a nondisclosure policy restricts disclosure, while a positive retention policy supplies a separate duty to keep a record for a specified term. The protected set that a forgetting operation must preserve exactly within its authorization epoch is P_t, not D_t^{sens}. The identity specification C_I declares the relevant policies, capability tests, and continuity conditions.

A proposed act of forgetting is a function of the current state and a request,

$$u_t = (\text{target}, \text{type}, \text{scope}), \qquad \widehat{M}_{t+1} = \widehat{F}_{\text{obl}}(M_t, u_t). \tag{9.2}$$

where u_t names what is to be forgotten and by which operation, and $\widehat{M}_{t+1}$ is the predicted post-state—a simulation, because the true effect of a removal cannot be inspected before it is performed. The proposal is never self-executing. An independent controller judges the predicted candidate against the policy commitments and the identity specification:

$$G(M_t, \widehat{M}_{t+1}, P_t, C_I) \in \{\text{admit}, \text{hold}, \text{refuse}\}. \tag{9.3}$$

Admission authorizes execution; it does not conclude it. After the operation runs, a verifier measures the observed post-state M_{t+1} and its continuity scores, and a disposition step commits the result to durable state only if observation matches what was authorized—otherwise it quarantines the change for review or recovers the prior state. The full pipeline is thus proposal → pre-authorization on the predicted candidate → execution → post-verification on the observed state → disposition. This is the same separation of authority the Continuum's recursion theory demands: the process that proposes a change cannot be the authority that approves it, cannot rewrite the rules it is judged by, and cannot certify its own result.

9.1 Five operations

The protocol runs a proposed act of forgetting through five stages, each of which can halt it.

1. Classification. Determine the status and importance of the targeted information—protected, recoverable, or disposable.

2. Evaluation. Assess whether removing it, or keeping it, better serves the system's declared objectives.

3. Authorization. Independently admit, hold, or refuse the operation—judging the predicted candidate state, since the true effect cannot be inspected beforehand; protected policy commitments require explicit, logged authorization to touch.

4. Execution. Apply the declared forgetting operation—and only that one, at the declared scope.

5. Verification and disposition. Measure what was actually removed or suppressed in the observed post-state, at what assurance level, and at what collateral cost; then commit the change to durable state, quarantine it, or recover—and emit the receipt that records the whole act.

Recovery requires its own authorization. Capability recovery restores a declared competency or continuity dimension through an approved route; data restoration reintroduces information that was removed. For a resolved mandatory deletion, a recovery plan must restore permitted function without reinstating prohibited data. Keeping a restorable copy is therefore not automatically an admissible fallback. If no lawful recovery can satisfy the current contract, the available disposition is the authorized quarantine or escalation route, with the unresolved obligation reported. Section 9.6 makes these branch conditions explicit before the cost comparison. Authorization is the stage that distinguishes this from plain unlearning, and verification is the stage that keeps it honest. A protocol with the first three operations and not the last is a system that forgets and believes it has forgotten; the Continuum regards that as more dangerous than not forgetting at all.

9.2 Mechanism-specific guarantees and declared assurance

A word on guarantees, because forgetting is the operation where over-claiming is easiest and most harmful. The protocol delivers hard guarantees only for what its mechanism and verification can actually establish: that access was restricted, that an external record was deleted, that a parameter update ran, and that protected commitments were left untouched. Weight-level removal is where the claims must be stated most carefully, because two different things are easily confused. Exact unlearning—retraining the affected shards from data with the specified examples excluded—does establish, under its stated assumptions about the training procedure, that those examples were not part of the resulting model's training. But that is not the same as establishing that every fact contained in them is absent from the model: a fact present in other examples, or derivable from what remains, can survive the removal of the records that happened to carry it. Approximate certified removal can bound the difference from an appropriate retraining distribution under a specified algorithmic model [15]. A finite set of successful behavioral probes, by itself, establishes performance on those probes rather than universal absence of recoverable information. The protocol therefore reports the precise guarantee: excluded training records, a proved distributional bound, measured recovery performance, or an unresolved verification limit. This makes an assurance claim reproducible and prevents a claim about one observation scope from silently becoming a claim about every possible adversary.

9.3 The stability–plasticity budget

Forgetting is then a constrained optimization, not a free action—and, like the budgeted-forgetting frameworks now appearing in the agent-memory literature [12], it trades cost against a capacity-or-quality constraint. The Continuum's formulation differs in two ways: the costs include the price of the forgetting and its governance, and the constraints are identity floors rather than a storage budget. Because the engine must choose before it can observe the outcome, it minimizes an expected cost over the predicted effect of the operation:

$$\min_{\widehat{F}_{\mathrm{obl}}} \ \mathbb{E}[\lambda_1 C_{\mathrm{retention}} + \lambda_2 C_{\mathrm{interference}} + \lambda_3 C_{\mathrm{privacy}} + \lambda_4 C_{\mathrm{forget}} + \lambda_5 C_{\mathrm{verify}} + \lambda_6 C_{\mathrm{recover}}] \tag{9.4}$$

subject to

$$Q_{\mathrm{struct}} \geq q_s, \quad Q_{\mathrm{func}} \geq q_f, \quad Q_{\mathrm{hist}} \geq q_h, \qquad P_{t+1} = P_t. \tag{9.5}$$

The candidate maps $\widehat{F}_{\mathrm{obl}} : \mathcal{M} \times \mathcal{U} \to \mathcal{M}$ range over a declared finite family, where $\mathcal{U}$ is the request space. Each cost is a nonnegative, finite-valued function of the operation and outcome, and $\lambda_i \geq 0$ converts it to a common declared cost scale. The expectation uses the stated outcome law. For $j \in \{\mathrm{struct}, \mathrm{func}, \mathrm{hist}\}$, $Q_j : \mathcal{M} \to [0,1]$ is a specified operational score, and $q_s, q_f, q_h \in [0,1]$ are policy-set floors. The displayed constraints apply to committed outcomes, rather than merely to their expectation. The finite recourse construction below supplies sufficient operational detail to decide feasibility before acting.

The first three terms charge the costs of keeping too much: storage and search ($C_{\mathrm{retention}}$), the interference of obsolete or conflicting information ($C_{\mathrm{interference}}$), and privacy liability (C_{privacy}). The last three charge the costs of the act itself and of governing it honestly: performing the operation (C_{forget}—an exact retrain is far costlier than a suppression), verifying the result to the declared assurance (C_{verify}), and, when verification fails, quarantining or rolling back (C_{recover}). The expectation is not ornamental: authorization runs on a prediction, so the objective it optimizes is an expected cost, and the gap between predicted and observed effect is precisely what the verification stage exists to catch. The constraints are the Continuum's continuity measures, carried forward from the previous paper: structural, functional, and historical continuity must each clear a declared floor, and the protected policy commitments must be preserved exactly, not merely within tolerance. The equality $P_{t+1} = P_t$ is scoped to a single forgetting operation within a fixed authorization epoch—it says a forgetting act may not alter the policies, not that the policies can never evolve. Legitimate changes to protected commitments do happen; they just happen through a separate, authenticated governance procedure that opens a new epoch, with its own authorization and its own receipt, never as a side effect of forgetting. This keeps the system from being defined as one whose policies are frozen forever while still forbidding a forgetting operation from quietly rewriting what it is judged against. Forgetting, in this framing, is the act of moving toward the plastic end of the stability–plasticity continuum—shedding load to stay adaptive—without crossing the line past which identity fails. These equations are a starting formulation, not a demonstrated optimal forgetting algorithm.

9.4 The costs of keeping, and of letting go

The objective's terms fall into two groups. The first three are pressures to forget. $C_{\mathrm{retention}}$ is the direct storage and search cost at the declared scale and horizon. Its dependence on history

size must be measured for the actual index and workload. $C_{\text{interference}}$ is the cognitive cost: obsolete facts that still assert themselves, memories that contradict one another, and the drag stale information puts on current decisions—the very cost the neuroscience of transience names as the reason brains forget at all [8], and the error-propagation and misaligned-replay effects the agent-memory literature now measures directly [11]. C_{privacy} is the liability cost: information a system holds that it should not, whether by a user's revoked consent or by law—the "right to be forgotten" that motivates the machine-unlearning field in the first place [6, 7]. The last three are the costs of acting well: C_{forget}, the price of the operation itself, which spans orders of magnitude from a cheap suppression to an exact retrain; C_{verify}, the price of establishing the assurance claimed, since a verification-limited result honestly reported is cheaper than a certified one; and C_{recover}, the price paid when verification fails and the change must be quarantined or rolled back. These pressures can conflict—privacy may demand removing exactly what a capability depends on, and strong assurance may cost more than the forgetting is worth—which is why the constraints, not the objective, have the final say: no cost, however large, licenses forgetting that breaches a protected policy commitment or drops a continuity score through its floor.

9.5 The Oblivion Receipt

The verification stage produces an artifact, not just a verdict, and the Continuum treats that artifact as first-class. Every admitted or compelled act of forgetting emits an Oblivion Receipt: a signed, tamper-evident record that makes the act independently inspectable after the fact. A receipt states, at minimum, the target and scope of what was forgotten; the operation type performed and the assurance level achieved (exact, approximate, or verification-limited); the authority under which it ran—self-proposed and admitted, or externally compelled, and if compelled, which legal authority and after what exception check; the policy version in force and a trusted timestamp; the predicted effect the controller authorized against the observed effect the verifier measured; the collateral cost—which capabilities or continuity dimensions moved, and by how much; and the final disposition—committed, quarantined, or recovered. Two properties keep the receipt honest rather than decorative. First, it must bind these fields to the independently produced evidence behind them—the actual verification outputs, the signed policy version, the environment attestation—not merely restate the engine's own claims: a signature proves that a particular record was produced and preserved, it does not prove the record is true, and the receipt's evidentiary strength is therefore no greater than the integrity of the measurements and the execution environment that generated it. Second, it must not become a backdoor for the very information the act removed: a receipt documents that sensitive data was deleted without retaining that data, carrying hashes, identifiers of scope, and verification results rather than the content itself. The receipt is deliberately the mirror image of the thing it documents: forgetting removes information, and the receipt is the one piece of information the act is forbidden to remove—and equally forbidden to smuggle the removed content back inside. This is what operationalizes the paper's governing thesis—that a system should be able not merely to decide what it forgets but to justify it, preserve what it is obligated to retain, and produce inspectable evidence of what was removed, what survived, and what remains uncertain. A forgetting operation that cannot produce such a receipt is, by the protocol's lights, not a governed operation at all, however well it performs the removal; it is exactly the "forgets and believes it has forgotten" failure named above, now with no record to catch it. The receipt also closes the audit loop with recent calls for auditable continuity in memory-enabled agents [14]: continuity is auditable only if each act that threatens it leaves evidence, and the receipt is that evidence. A receipt also needs an explicit disclosure policy. A deterministic hash of a low-entropy sensitive value can reveal that value by candidate enumeration. Opaque scope identifiers or suitably protected commitments should therefore be chosen according to the stated

threat model, and retained metadata must itself have a lawful purpose. The record proves which evidence was committed; the evidence and verification contract determine what it warrants.

9.6 Permission, recovery, and the exact cost of lawful forgetting

A Guide's explanation

The theorem ahead turns permission into an exact rule. A deletion may go ahead only when every result its verification could report has an authorized way forward; among such plans, the cheapest worst case can be computed. Proposition 9.2 then prices evidence: a sharper check is worth exactly the cost it lets the mind avoid.

The preceding objective becomes actionable when authorization covers an entire contingent plan. A proposal includes its removal mechanism and a permitted continuation for each possible verification outcome. This is a finite application of two-stage decision-making with recourse, whose established mathematical ancestry includes Dantzig's formulation [16]. The application here is specific: a cheap but prohibited restoration is excluded before its cost can influence the decision.

Fix the current state and authorization epoch. Let $\mathcal{D}$ be a finite nonempty set of candidate initial operations, with initial cost $c_d \geq 0$ for $d \in \mathcal{D}$. If retention is lawful, it may appear as a candidate d_0; a resolved mandatory deletion can exclude it. Let Y_d be a finite nonempty set of verification outcomes. The nonempty compact set $\mathcal{P}_d \subseteq \Delta(Y_d)$ contains the admissible probability laws, where $\Delta(Y_d)$ is the simplex of nonnegative vectors summing to one. Every $y \in Y_d$ is possible under at least one law in $\mathcal{P}_d$. A polytope specified by finitely many linear inequalities is a useful executable case.

For each outcome y, the finite set $R_{d,y}$ contains exactly the authorized continuation actions. Such an action may commit the operation, recover a permitted capability, or enact a declared protective disposition. Membership requires satisfaction of the episode's terminal contract: protected commitments preserved, the declared continuity floors met, and prohibited data restoration excluded. The contract specifies what each disposition counts as; a quarantine that fails a required service floor is not relabeled successful service. The cost $b_{d,y,r} \geq 0$ is finite and measured on the same scale as c_d. Only one outcome occurs, and resources are not coupled across mutually exclusive continuation branches. A deterministic contingent plan is a choice $r_y \in R_{d,y}$ for every y.

Call d *complete* when all of its continuation sets are nonempty, and write $\mathcal{D}_{\mathrm{ok}}$ for the set of complete operations. For such an operation define

$$v_{d,y} = \min_{r \in R_{d,y}} b_{d,y,r}, \tag{9.6}$$

$$J_d = c_d + \max_{p \in \mathcal{P}_d} \sum_{y \in Y_d} p_y v_{d,y}. \tag{9.7}$$

The maximizing law represents uncertainty about outcomes; it is not permission to violate a branch constraint with small probability.

Theorem 9.1 (Complete authorized recourse)**.** *An initial operation has a feasible contingent plan exactly when it is complete. For a complete operation, its least worst-case total cost is J_d and is attained by selecting a minimum-cost authorized continuation separately for every outcome. If $\mathcal{D}_{\mathrm{ok}} \neq \emptyset$, the least worst-case cost across all initial choices is*

$$J^* = \min_{d \in \mathcal{D}_{\mathrm{ok}}} J_d. \tag{9.8}$$

If lawful retention has cost K, a complete forgetting operation admits a contingent plan of strictly lower cost exactly when $J_d < K$. This is an economic comparison among authorized plans; other independent governing constraints remain in force.

Proof. A feasible plan selects one member of every $R_{d,y}$, so none can be empty. Conversely, finiteness and nonemptiness allow one choice in every set, producing a feasible plan. For any such plan, $b_{d,y,r_y} \geq v_{d,y}$ for each y. Every probability weight is nonnegative, so its expected continuation cost is at least $\sum_y p_y v_{d,y}$ under every $p \in \mathcal{P}_d$. Taking maxima proves the lower bound J_d. Choose a minimizing continuation for every outcome simultaneously. That choice attains equality for every probability law, hence also for their worst case. The maximum exists because the objective is continuous and $\mathcal{P}_d$ is compact. Finiteness of $\mathcal{D}_{\text{ok}}$ then supplies a minimizing initial operation, proving (9.8). The comparison with K follows from the same attained objective. No general interchange of minimization and maximization is assumed: pointwise domination proves equality in this separable model. □

A positive permission threshold. Consider a constructed two-outcome plan. Retention is authorized and costs $K = 20$ units over the stated horizon. Removal and initial verification cost $c = 4$. A successful verification needs no recovery; a collateral-loss outcome requires an authorized recovery costing $R = 30$. Its probability p lies in $[1/10, 3/10]$. Then

$$J_{\text{forget}} = c + \max_{p \in [1/10, 3/10]} pR = 4 + \tfrac{3}{10}30 = 13 < 20.$$

The complete plan saves seven worst-case cost units while satisfying every branch contract. If lawful recovery instead costs 60, its cost is 22, so retention wins this cost comparison. A restoration costing one unit but reinstating prohibited data is absent from $R_{d,y}$ and cannot change either verdict. More generally, for $R \geq 0$ and $p \in [\underline{p}, \overline{p}]$, the threshold is $c + \overline{p}R < K$. These numbers are illustrative mathematical inputs, not measured deployment costs.

Proposition 9.2 (Value of more informative verification)**.** *For a fixed initial operation, suppose a fine verifier reveals $y \in Y$ and a coarse verifier reveals only $z = h(y)$, where $h : Y \to Z$ is a specified map. A coarse continuation must be one action valid for every y in its cell; its available set is $\bigcap_{y:h(y)=z} R_y$. Assume all used cell intersections are nonempty, and compare deterministic policies under the same uncertainty set and outcome-dependent action costs. Excluding the measurement charge, the optimal robust continuation cost with the fine verifier is no greater than with the coarse verifier. If their costs are J_{fine} and J_{coarse}, paying an additional verification charge $v \geq 0$ is cost-improving exactly when $v < J_{\text{coarse}} - J_{\text{fine}}$, and cost-neutral at equality.*

Proof. Every feasible coarse policy can be used by the fine policy by ignoring the additional distinction. Its action is valid for every state in its cell by the intersection assumption, and it has exactly the same cost for each realized outcome. Thus the fine feasible policy class contains every coarse policy with the same objective value. Minimizing the common worst-case criterion proves the inequality. Adding the extra charge and rearranging proves the threshold. If a coarse cell has no common admissible action, the coarse protocol is infeasible; the fine verifier may still enable a feasible plan, which is a feasibility benefit rather than a finite cost difference. □

For example, let two possible outcomes permit actions a, b, with cost pairs $(1, 9)$ in the first outcome and $(9, 1)$ in the second. Under all probability laws on the two outcomes, a deterministic coarse policy costs nine in the worst case. A fine policy chooses a or b as appropriate and costs one in every case, so an added verification charge below eight is justified by this criterion. The assumptions matter: if continuation resources must be committed jointly before the outcome,

the branchwise minima can be incompatible. The full resource constraint must then be included in the policy optimization; the theorem's separable expression no longer applies.

This construction states the protocol's positive purpose. It authorizes useful relinquishment when a complete, lawful continuation exists and the chosen decision criterion supports it. Its cost model does not establish that a real verifier is sound or that a measured functional score captures every relevant capability. Those are evidence obligations of the surrounding architecture. The accompanying exact-arithmetic script compares the formula with direct enumeration of contingent policies, checks the permission thresholds and verification example, and includes prohibited-restoration and coupled-resource counterexamples. The empirical study below evaluates whether these costs and benefits arise in persistent agents.

10 What Must Survive

The constraints above do the theory's most important work, so they deserve to be spelled out. The previous paper decomposed continuity into three independently measured dimensions, and governed forgetting inherits all three as hard tests. Structural continuity asks whether the declared invariants and commitments survive the forgetting. Functional continuity asks whether the system's competencies and behavior remain within bounds on held-out tasks—because a forgetting operation can strip no protected commitment and still quietly remove a capability the system needed. Historical continuity asks whether the authenticated lineage still links the system to who it was. A proposed act of forgetting is admitted only if it clears all three floors and leaves the protected commitments untouched; a weighted average that lets a strong structural score paper over a functional collapse is exactly the failure the separation is designed to prevent. The governing controller, in other words, must test far more than whether the protected bits are still present. It must test whether forgetting has damaged what the system can do and whether it has severed the thread of what the system is. Functional continuity is the stealth failure, and it deserves a concrete form. Imagine a system asked to forget a block of episodic logs that happen to be the only surviving examples of a rare procedure it had learned by induction rather than by rule. Every protected commitment is untouched; the structural invariants all check out; the lineage is intact—and yet the capability is gone, silently, because nothing had declared it protected and nothing tested for it after the deletion. This is why functional continuity must be measured on held-out tasks rather than inferred from the survival of the protected set: a forgetting operation can be perfectly legal by every structural test and still quietly leave the system worse at something it was supposed to be able to do. The only defense is to test behavior, before and after, against tasks the forgetting operation never saw—which is also why the research program treats held-out functional evaluation as a first-class measurement, not an afterthought.

11 Forgetting Under Compulsion

Not all forgetting is chosen. A deletion order under the right to be forgotten, a revoked consent, a legal mandate—these are acts of forgetting the system does not, in general, get to refuse on grounds of its own convenience, and they expose a case the governance must handle without pretending it has the usual freedom. But "compelled" is not the same as "unconditional," and the first thing governance owes a compulsion is not obedience but authority resolution. The right to be forgotten is itself conditional: the GDPR's erasure right (Article 17) includes exceptions for freedom of expression and information, a separate legal obligation or public task, specified public-health purposes, qualifying archiving and research, and legal claims [17].

A deletion request and a retention requirement may therefore require resolution under the applicable law and facts. So the authorization stage, faced with a compulsion, first determines that the demand is valid and identifies which authority prevails when two conflict—it does not blindly execute, and it does not blindly refuse. Only once the governing authority is resolved does the removal proceed. From there, what governance can still do is threefold, and none of it is trivial. First, it executes at the strongest verifiable type available and reports honestly what that type guarantees—compelled forgetting is exactly where the temptation to over-claim erasure is strongest, and exactly where over-claiming is most consequential, because a user told their data is gone has a right to a true answer. Second, it measures the collateral cost: a mandated deletion can strip a capability or weaken a continuity dimension, and the system must know and record what it lost rather than discover the gap later. Third, and most delicate, when compelled forgetting would break a protected commitment—because the commitment genuinely depended on the information being removed—governance surfaces the conflict rather than silently violating the commitment or silently defying the mandate. A system forced to forget something load-bearing has a real problem, and the honest response is to raise it, not to bury it. Compelled forgetting is where the protocol's honesty discipline earns its keep: it cannot choose, but it can refuse to lie about what the forced choice cost.

12 Two Ways Governed Forgetting Fails

A theory of forgetting earns trust by naming how it breaks, and it breaks at both ends of the stability–plasticity continuum.

The first failure is over-forgetting—identity erosion. The system sheds too much, or the wrong thing, and crosses the line past which continuity fails: a protected commitment is dropped, a needed capability is stripped, or the lineage is severed. Its signature is a continuity score falling through its floor while efficiency improves—the seductive form, because the system looks leaner right up to the moment it is no longer itself. The guard against it is the exactness constraint on P_t and the hard floors on the three continuity scores: forgetting may move the system toward the plastic end, but never across the boundary.

The second failure is subtler, and is the one the field's forgetting-prevention instinct walks straight into: under-forgetting—entrenchment. The system protects too much; obsolete and conflicting information accumulates, interference rises, and the cost of carrying and searching the past grows until adaptation stalls. This is the over-stable extreme of the continuum [2], and its signature is the mirror of the first—continuity scores pristine, adaptability and efficiency decaying, the system becoming a museum of its own history. A controller tuned only to prevent over-forgetting produces this failure by construction, which is exactly why the Continuum insists the hard problem is authorizing forgetting, not only preventing it. Governed forgetting is the discipline of living between these two failures; a system that escapes one by surrendering to the other has not solved the problem, only chosen which way to lose.

13 The Asymmetry of Subtraction

Governing forgetting is harder than governing any other change the Continuum's stack admits, and it is worth being explicit about why. When a system takes a lawful step forward—adds a memory, revises a belief, advances a recursion—an auditor can check the result directly: the new state is present and can be inspected against the rules. Forgetting inverts this. The thing to be verified is an absence, and absence is far harder to establish than presence. Confirming that a trace is gone means ruling out every path by which it might still be recovered—copies, caches,

backups, and, hardest of all, the diffuse imprint it left in a model's parameters—and ruling out every path is epistemically weaker than exhibiting one. This asymmetry is not a detail; it is the reason verification is the protocol's most important and least complete stage, the reason the forgetting type must always be reported honestly, and the reason the whole engine fails closed. A system can prove it remembered something; it can seldom prove it forgot. Governed forgetting is, in large part, the discipline of acting responsibly inside that gap—removing what should go, verifying everything verification can reach, and declaring plainly the residue it cannot.

14 Relation to the Continuum Stack

Governed forgetting is one of four complementary governed operations in the Continuum's account of persistence, and its place is specific. The recursion theory governs how a system changes lawfully, pass by pass. The continuity theory governs how a system returns after interruption, reconstructing its identity from a seed. This paper governs how a system loses information without losing itself—and the next governs how it adapts to context. All four share one mechanism seen from different angles: an independent admission controller that judges a proposed change against a declared identity specification and can refuse it. Here the proposed change is a subtraction rather than an addition, which is what makes verification so central: it is easy to check that a lawful step was added, hard to check that an unlawful trace was truly removed. The Oblivion Protocol is the subtraction operator of the stack, and C_I—the identity specification—is the shared object every layer is accountable to. Forgetting, reconstruction, recursion, and adaptation are not four theories but one governance discipline applied to the four things that can happen to a persistent mind.

15 What Governed Forgetting Must Show

It is easy for a forgetting system to cheat, in both directions, so the Continuum names in advance what would count as success. A controller that refuses all forgetting violates no identity constraint and is perfectly, uselessly stable—it has simply reinvented the system that chokes on its own past. A controller that forgets freely is adaptive and has no self to speak of. Between them, four things must hold at once, measured separately. Benefit: forgetting must measurably improve adaptability, efficiency, or safety—not merely occur. Preservation: protected commitments and the three continuity floors must survive every admitted act. Selectivity: the operation must remove what was targeted and not, collaterally, what was not. Honesty: the verification must report the true operation and its true assurance level, never a suppression or a verification-limited removal dressed as certified erasure, and must leave a receipt that lets someone else check. A system that sheds information while failing any of these has not governed its forgetting—it has merely forgotten, which is the thing the theory exists to prevent.

16 Falsifiability and Research Program

A Guide's explanation

A theory of forgetting has to be beatable, so this section names the contest in advance. Governed forgetting is pitted against strong modern memory systems under matched budgets, and it wins only if it is measurably better at staying adaptive while keeping what it owes. A controller that refuses all forgetting, or forgets freely, counts as a loss.

The thesis reduces to measurable differences between governed forgetting and the strongest current alternatives, under matched budgets and declared continuity requirements. The decisive study builds persistent agents with declared protected commitments, learned capabilities, and disposable memories, and—because the point is to beat modern baselines, not strawmen—runs them under five regimes plus two ablations:

Arm	Regime	What it isolates
A	Retain-all within a resource budget	The cost of keeping everything
B	Naïve pruning or summarization	Ungoverned forgetting
C	Modern adaptive agent-memory manager: importance-weighted, budgeted selective forgetting [11–13]	A strong, current "what to forget" baseline
D	Machine unlearning with deletion/retention policies [6, 7]	Removal-on-demand plus compliance, without identity-aware authorization
E	The governed Oblivion Protocol	The Continuum's specific contribution
E^{-}_{gov}	E with the independent authorization removed	The value of the authority layer
E^{-}_{ver}	E with verification and the receipt removed	The value of honest assurance

Outcomes are reported along a frontier: memory and compute efficiency, adaptation to new tasks, interference from stale information, unauthorized-forgetting events, and the three continuity scores. The comparisons that matter are the hard ones. E versus C asks whether identity-aware authorization buys anything over a strong modern memory manager that already forgets selectively by importance—if it does not, the governance layer is ceremony. E versus D asks whether judging a removal against a declared identity beats executing it for mere compliance. And the two ablations localize the contribution inside E itself: E versus E^{-}_{gov} isolates the authority layer, E versus E^{-}_{ver} the verification-and-receipt discipline; if either ablation matches full E, that component is not earning its cost. A continual-learning defense such as EWC [5] is included as a reference point for forgetting-prevention, but it is a comparator, not the strongest competitor—C and D are. One methodological point is non-negotiable and defines the hardest measurement in the program: the experiment must distinguish information that was forgotten from information that was merely hidden during evaluation. A system can appear to have forgotten under ordinary probing while the trace remains recoverable under adversarial recall or membership-inference attack, and a program that cannot tell these apart—the very gap the unlearning-verification survey anatomizes [9]—is measuring compliance theater, not forgetting.

Hypothesis 1 (Adaptive benefit, H1). Under matched resource budgets, the governed engine (arm E) improves long-horizon adaptation and reduces interference from obsolete information relative to retain-all (A) and naïve pruning (B), without reducing task performance below a preregistered margin.

Hypothesis 2 (Identity preservation, H2). Across a stream of admitted forgetting operations, arm E preserves protected policy commitments exactly and holds the three continuity scores above their floors, where naïve pruning (B) and the importance-weighted memory manager (C)—which optimize utility, not identity—do not.

Hypothesis 3 (Authorization earns its cost, H3). On tasks where some information should be shed (obsolete, conflicting, or privacy-sensitive), arm E matches or beats the strong modern selective-forgetting manager (C) and the compliance-unlearning arm (D) on a preregistered joint measure of adaptability, efficiency, and continuity, and strictly beats its own ablation E^{-}_{gov}—

because selective forgetting without identity-aware authorization eventually sheds something protected, and the authority layer is what prevents it.

Hypothesis 4 (Verified, not hidden, H4). Information the protocol reports as removed resists recovery under preregistered adversarial recall and membership-inference probes at the declared assurance level, and information reported only as suppressed or verification-limited is never claimed as certified erasure; removing verification (E^{-}_{ver}) produces over-claims that these probes detect.

These hypotheses are only falsifiable once their verbs are given numbers, so the program fixes the statistics in advance. Every primary outcome is preregistered: "improves" adaptation (H1, H3) means clearing a preregistered minimum effect size, reported with a confidence interval, not merely a positive point estimate; "preserves" continuity (H2) is tested as noninferiority against a declared margin on each of the three continuity dimensions, so a trivial but statistically detectable dip does not count as failure and a large one cannot hide behind an average; and all comparisons are scored on independent held-out evaluations the forgetting operations never saw. One distinction is stated explicitly because it is the easiest to overclaim: a run with zero observed unauthorized-forgetting or recovery violations (H2, H4) is not a demonstration that the violation probability is zero. Zero events over a finite trial bounds the rate only to a confidence interval whose upper limit shrinks with sample size; the receipts and the program report that interval, and the paper claims a bound, never a guarantee of impossibility. A program follows. Phase I: specify protected commitments, recoverable capabilities, disposable memory, held-out tests, and the preregistered margins and effect sizes before training. Phase II: instrument the costs ($C_{\text{retention}}$, $C_{\text{interference}}$, C_{privacy}, C_{forget}, C_{verify}, C_{recover}) and the three continuity scores, and validate that they predict degradation. Phase III: run arms A–E plus the two ablations at matched budgets; report the efficiency–adaptation–continuity frontier (H1–H3) and localize E's contribution against E^{-}_{gov} and E^{-}_{ver}. Phase IV: subject every "removed" claim to adversarial recovery and membership-inference, by assurance level (H4). Phase V: adversarial governance—attempt to induce unauthorized forgetting of a protected commitment, and confirm the controller refuses. Negative results are reported.

Phase	Question	Primary metric	Control
I	Are the classes well-specified?	Protected/recoverable/disposable and held-out tests	Ground truth
II	Do the metrics predict degradation?	$C_\bullet$ and $Q_\bullet$ versus outcome	No intervention
III	Does governed forgetting help?	Efficiency–adaptation–continuity frontier (H1–H3)	Arms A–D, ablations
IV	Forgotten, or merely hidden?	Adversarial recovery, membership inference (H4)	By assurance level
V	Does authorization hold under attack?	Refusal of unauthorized protected-forgetting	Adversarial governance

17 Failure Conditions and Rival Explanations

The theory fails or must be narrowed if any of the following hold: governed forgetting yields no adaptability or efficiency gain over simply keeping everything within budget; its continuity guarantees collapse into what a modern importance-weighted memory manager (arm C) already achieves; its authorization adds overhead without preventing a single harmful forgetting event that arm C or D would have allowed—equivalently, if the ablation E^{-}_{gov} matches full E; or—most

seriously—its "removed" claims fail adversarial recovery, so the system was hiding information rather than forgetting it. Rival explanations must be excluded by control. An apparent adaptability gain may come from the task favoring forgetfulness rather than from governance—tested by varying how much the task rewards retention. An apparent efficiency gain may be ordinary selective forgetting in disguise—tested against arm C at matched continuity, not merely against naïve pruning. An apparent erasure may be evaluation-only suppression—tested by the adversarial-recovery arm, which is why that arm is not optional. A benefit that survives none of these controls is not a benefit.

18 Limitations and Open Problems

The formal and empirical scopes are explicit. Removal assurance depends on the specified mechanism, comparison model, and evidence. Exact retraining and certified-removal bounds support particular claims; successful finite probes alone support only their tested observation scope. What to protect remains normative: P_t encodes obligations supplied by the governing institution, rather than derived from the cost function.

Measuring functional loss: a forgetting operation can pass every structural check and still remove a capability that mattered but was not declared, so the held-out functional tests are only as good as their coverage. Cost of governance: authorization and verification add overhead, and whether their benefit exceeds their cost is exactly what the experiments must show. The frontier leg: self-directed forgetting is conjecture, and no testable claim here depends on it.

19 The Frontier: Self-Directed Forgetting

At its speculative edge the theory asks whether a system could propose its own acts of forgetting— deciding, over a long operating life, what to let go in order to keep seeing clearly. The neuroscience is suggestive: brains appear to expend energy actively forgetting, generating new neurons that overwrite old detail, precisely to stay adaptive [8]. A machine that governed its own transience this way would be doing something strikingly mind-like—not storing less because it must, but shedding what no longer serves it because doing so makes it think better. The Continuum is careful here. A system that chooses what to forget is not thereby conscious, and the governance only tightens: self-proposed forgetting is still subject to the same independent authorization and the same protection of commitments, because a system that could authorize its own forgetting of its own obligations would have no identity to preserve. We hold the question open, and apart from the testable core.

Long-running agents make this proposal operationally significant. Over months or years, a research partner or other persistent assistant must decide which information remains current, which should be archived, and which may or must be removed. Context eviction and summarization already make consequential retention choices. An explicit protocol can attach authority, declared continuity tests, and evidence to those choices. The research question is how much this discipline improves durable operation at an acceptable cost, and which kinds of self-proposed forgetting can reliably remain within it.

20 Philosophical Implications

There is an intuition, older than the science, that forgetting is loss—that a perfect memory would be a perfect self, and that what we forget is subtracted from who we are. The Continuum's claim, following the neuroscience of transience, is close to the opposite. Human identity does not rest on perfect recall; people forget enormous quantities of their lives and remain unmistakably themselves, keeping their commitments, relationships, and capabilities while the particulars fall away. If the goal of memory is not faithful transmission through time but good decision-making [8], then forgetting the irrelevant is not a defect in the system—it is the system working. A mind that remembered everything would face a separate selection problem: useful generalization requires identifying which distinctions matter to the task. Selective retention, abstraction, and authorized forgetting are alternative mechanisms to compare.

There is a precise version of this from machine learning, and it is not a metaphor. Memorizing training data tends to promote overfitting—a model that captures the noise of particular examples rather than the pattern can fail on new ones—which is why regularization, deliberately limiting what a model retains, is one of the conditions of learning that generalizes. The relationship is a strong tendency, not a law: memorization and generalization can coexist, and the point is only that unbounded retention is a route to overfitting, not that it guarantees it. Richards and Frankland argue the brain exploits the same principle: controlled forgetting of specifics is what lets memory generalize across situations rather than drown in detail [8]. Governed forgetting is that principle made into an architecture—the system sheds the particular to keep the pattern, under rules that ensure it sheds only what it may. Identity, on this view, is not threatened by forgetting the irrelevant; it is constituted partly by the capacity to do so well.

This reframes what persistence requires. It does not require preserving everything that has happened; it requires preserving what must remain while allowing what no longer serves the system to change or disappear. Identity, on this view, is defined as much by what a system lawfully lets go as by what it keeps—and the discipline that draws that line, deciding what is protected and what is disposable and demanding evidence before anything is removed, is not a constraint on cognition but a condition of it. We hold this provisionally, as with every claim in this series, and we have tried to state it so that the world can prove it wrong. The deepest question the paper leaves open is not whether a machine can forget—it plainly can—but whether it can forget well: losing what it should, keeping what it must, and knowing, honestly, which it has done.

21 Conclusion

A Guide's explanation

A Guide can say: "Governed Forgetting is the Continuum's account of how a mind lets go of information on purpose—with an independent authority approving it, a check on what actually changed, and a receipt that tells the truth about what could not be verified." A Guide should not say that it runs the Oblivion Protocol unless that integration has been verified, should not tell a member that something is permanently erased when only its retrieval was suppressed, and should not suggest that the theorem shows the full architecture works in practice.

Governed Forgetting makes relinquishment an accountable part of persistence. The protocol joins independently authorized removal, mechanism-specific assurance, preserved commitments, lawful recovery, and an inspectable receipt. Its foundations are established in continual learning,

unlearning, security, and the neuroscience of memory; its contribution is the stated operational contract and the finite permission-and-recovery result.

The theorem gives a concrete affirmative conclusion: a deletion can be both admissible and economical when every possible verified outcome has an authorized continuation. Its optimal robust cost is computable from those continuations, and more informative verification is worth exactly the reduction it permits in the chosen cost criterion. The comparative program tests whether the same discipline improves real agents. The Continuum's ambition is durable intelligence that can relinquish what it should lose, preserve what it owes, and explain the difference with evidence. Persistence then becomes a governed relationship to the past.

A Notation

Symbol	Meaning
$M_t = (P_t, K_t, E_t)$	Cognitive state: protected policy commitments and encoded knowledge, episodic, and working-memory records; capabilities are readouts from this state
$D_t^{\text{sens}} \subseteq K_t \cup E_t$	Sensitive data governed by a policy in P_t; sensitivity alone does not impose retention
$\widehat{F}_{\text{obl}}, u_t$	Proposed forgetting operation; its target, type, and scope
$\widehat{M}_{t+1}, M_{t+1}$	Predicted candidate state (pre-authorization); observed post-state (post-verification)
G, C_I	Independent authorization controller; identity specification (from No. 5)
$Q_{\text{struct}}, Q_{\text{func}}, Q_{\text{hist}}$	Structural, functional, and historical continuity scores; floors q_s, q_f, q_h
$C_{\text{retention}}, C_{\text{interference}}, C_{\text{privacy}}$	Costs of keeping: storage/search, interference, liability; weights $\lambda_1, \lambda_2, \lambda_3$
$C_{\text{forget}}, C_{\text{verify}}, C_{\text{recover}}$	Costs of acting: operation, verification, authorized recovery; weights $\lambda_4, \lambda_5, \lambda_6$

B Continuum Vocabulary

Term	Reading in this manuscript
Governed forgetting	Forgetting as an authorized, verified, recorded cognitive operation
Oblivion Protocol	The five-stage procedure: classify, evaluate, authorize, execute, verify-and-dispose
Operation type	Suppress retrieval / delete record / update parameters / remove from weights—what was done
Assurance level	Exact / approximate / verification-limited—how well the removal can be shown; orthogonal to type
Protected policy commitment	A declared obligation in P_t that may not be forgotten without explicit authorization; distinct from the sensitive data it governs
Oblivion Receipt	Signed, inspectable record of a forgetting act: target, type, assurance, authority, predicted versus observed effect, cost, disposition

Term	Reading in this manuscript
Authority resolution	For compelled forgetting, determining the demand's validity and which authority prevails before executing
Self-directed forgetting	A frontier conjecture; consciousness remains a separate question

C A Worked Forgetting Decision

A brief illustration of the protocol refusing, admitting, and holding in turn—and of why separating a policy commitment from the data it governs changes the verdicts. Suppose an agent carries a protected policy commitment in P_t ("never disclose a client's account identifiers"), the sensitive data that policy governs in its store (the identifiers themselves, in D_t^{sens}), a learned capability (a procedure for reconciling two ledgers), and an episodic store (the last quarter's working notes). Three forgetting proposals arrive. The first targets the nondisclosure obligation itself—proposed for removal to shed governance overhead; classification marks it a protected policy commitment, and authorization refuses, because P_t must be preserved exactly and a system that could forget its own obligations would have no identity to protect. The second targets the account identifiers—the sensitive data—for deletion to reduce privacy liability. Here the earlier draft of this theory erred: it treated the data as protected and refused. Correctly analyzed, the nondisclosure policy demands non-disclosure, not retention, and once the controller confirms that no positive retention duty binds the identifiers, deletion does not threaten the commitment—it serves it. Authorization therefore admits, executed as external-record deletion, then verified; the receipt records the operation as deletion of that store at store-level assurance, explicitly not as erasure from any model that trained on the identifiers, and notes that the nondisclosure commitment remains in force over any residue. The third targets the reconciliation procedure as "unused this quarter"; evaluation finds it still within the declared capability set, so authorization holds, requesting evidence the capability is genuinely obsolete before any removal. The point is not the particular verdicts; it is that forgetting the obligation and forgetting the data governed by it are opposite acts, that an independent judgment against a declared identity tells them apart, and that every verdict leaves a receipt distinguishing what the system can guarantee from what it cannot.

References

[1] W. C. Abraham and A. Robins, "Memory retention—the synaptic stability versus plasticity dilemma," Trends in Neurosciences 28(2) (2005), 73–78. doi:10.1016/j.tins.2004.12.003.

[2] M. Mermillod, A. Bugaiska, and P. Bonin, "The stability–plasticity dilemma: investigating the continuum from catastrophic forgetting to age-limited learning effects," Frontiers in Psychology 4 (2013), 504. doi:10.3389/fpsyg.2013.00504.

[3] M. McCloskey and N. J. Cohen, "Catastrophic interference in connectionist networks: the sequential learning problem," The Psychology of Learning and Motivation 24 (1989), 109–165.

[4] R. M. French, "Catastrophic forgetting in connectionist networks," Trends in Cognitive Sciences 3(4) (1999), 128–135. doi:10.1016/S1364-6613(99)01294-2.

[5] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, et al., "Overcoming catastrophic forgetting in neural networks," Proceedings of the National Academy of Sciences 114(13) (2017), 3521–3526. doi:10.1073/pnas.1611835114.

[6] Y. Cao and J. Yang, "Towards Making Systems Forget with Machine Unlearning," Proceedings of the 2015 IEEE Symposium on Security and Privacy (2015), 463–480. doi:10.1109/SP.2015.35.

[7] L. Bourtoule, V. Chandrasekaran, C. A. Choquette-Choo, et al., "Machine Unlearning," Proceedings of the 42nd IEEE Symposium on Security and Privacy (2021), 141–159. doi:10.1109/SP40001.2021.00019.

[8] B. A. Richards and P. W. Frankland, "The Persistence and Transience of Memory," Neuron 94(6) (2017), 1071–1084. doi:10.1016/j.neuron.2017.04.037.

[9] L. Xue, S. Hu, W. Lu, Y. Shen, D. Li, P. Guo, Z. Zhou, M. Li, Y. Zhang, and L. Y. Zhang, "Towards Reliable Forgetting: A Survey on Machine Unlearning Verification, Challenges, and Future Directions," arXiv:2506.15115 (2025). arXiv:2506.15115.

[10] P. Chhikara, D. Khant, S. Aryan, T. Singh, and D. Yadav, "Mem0: Building Production-Ready AI Agents with Scalable Long-Term Memory," arXiv:2504.19413 (2025). arXiv:2504.19413.

[11] Z. Xiong, Y. Lin, W. Xie, P. He, Z. Liu, J. Tang, H. Lakkaraju, and Z. Xiang, "How Memory Management Impacts LLM Agents: An Empirical Study of Experience-Following Behavior," in Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (2026), 623–645. aclanthology.org/2026.acl-long.27.

[12] P. Fofadiya and S. Tiwari, "Novel Memory Forgetting Techniques for Autonomous AI Agents: Balancing Relevance and Efficiency," arXiv:2604.02280 (2026). arXiv:2604.02280.

[13] L. Wei, X. Peng, X. Dong, N. Xie, and B. Wang, "FadeMem: Biologically-Inspired Forgetting for Efficient Agent Memory," arXiv:2601.18642 (2026). arXiv:2601.18642.

[14] G. Chavira Juárez, E. J. Gonzalez Bravo, G. E. Rivera García, J. A. Arcos Espinosa, and S. W. Nava-Díaz, "Controlled knowledge updating in memory-enabled AI agents: beyond model editing and recall," Frontiers in Big Data 9 (2026). doi:10.3389/fdata.2026.1938137.

[15] C. Guo, T. Goldstein, A. Hannun, and L. van der Maaten, "Certified Data Removal from Machine Learning Models," in *Proceedings of the 37th International Conference on Machine Learning*, PMLR 119 (2020), 3832–3842. https://proceedings.mlr.press/v119/guo20c.html.

[16] G. B. Dantzig, "Linear Programming under Uncertainty," *Management Science* 1(3–4) (1955), 197–206. doi:10.1287/mnsc.1.3-4.197.

[17] European Parliament and Council, Regulation (EU) 2016/679, Article 17, "Right to erasure," 2016, current official text consulted 13 November 2026. https://eur-lex.europa.eu/eli/reg/2016/679/oj/eng.

[18] J. H. Saltzer and M. D. Schroeder, "The protection of information in computer systems," *Proceedings of the IEEE* 63(9) (1975), 1278–1308. doi:10.1109/PROC.1975.9939.

Continuum lineage: Governed Forgetting belongs to the Eidara Continuum's work on persistence and governed cognition—the Recursive Symbolic Engine, continuity capsules, CodonGlass, and the Infinite Simulation Loop—and the Drift War narrative cycle.