

To Read Is Not to Remember

Interpretation, Discontinuity, and the Making of a
Durable Reader

Abstract

A machine can receive an entire novel and still fail to interpret its relations; it can read a few passages and write as though it had read the whole. Between the delivery of text and the justification of a reading lies a problem that a larger context window cannot by itself resolve. This essay argues that the continuity of interpretation need not depend on the continuity of the interpreting model, provided that the history of encounters with a source is preserved as attributable, revisable evidence rather than as an accumulating substitute for the source. The argument develops through long-context research, literary hermeneutics, and the documented formation of Reframe's reading architecture. Its central case is instructive because it contains a reversal: Reframe deliberately removed a derived semantic index that had acquired too much authority, then had to discover how to remember without rebuilding that authority under another name. The resulting distinctions among source, reading operation, Reading Continuity, Compiled Knowledge, and two forms of Drift Calculus establish a testable programme, not a completed proof of machine comprehension. The essay follows that programme into executable reading, crash recovery, scholarly reference, disagreement, and the limits of evidence. It concludes that an accountable reading is neither a stored answer nor a simulation of continuous consciousness, but a history in which assertions can be traced, questioned, and changed.

The book and the claim

A reader opens a novel in an artificial-intelligence application and asks what happens to the relationship between two characters. The answer arrives almost immediately. It is fluent, persuasive, and furnished with references. Perhaps it even identifies a gesture from the first chapters whose significance becomes visible only near the end. The reader has one entirely reasonable question:

Did the model actually read the whole book?

An application can answer that question badly while telling no obvious lie. It can count the bytes it imported, the chapters it indexed, the tokens it transmitted, the fragments it retrieved, or the requests it completed. It can show a progress bar reaching one hundred percent. Every one of these measures may describe something that truly happened, yet none, by itself, establishes the proposition the reader meant to ask. A complete import establishes custody of a document. A complete sequence of requests establishes that a sequence of operations was attempted, or perhaps completed. Neither demonstrates that a later interpretation attends correctly to the work in its entirety. The attractive interface label *read* has concealed several different events. This is not simply a semantic objection to an unfortunate button. It determines the epistemic character of the whole product. A wrong answer about a minor character may become the premise of a synopsis; that synopsis may inform a screenplay; the screenplay may then be audited against the synopsis rather than against the original work. If each stage inherits the conclusions of the preceding stage as unexamined authority, an initial mistake can become more certain with every transformation. No spectacular hallucination is required. Ordinary continuity, the very quality a literary tool promises, can carry error forward.

The first problem is therefore to distinguish a model's *encounter* with text from its *understanding* of it. A passage may have been identified by deterministic software, selected for a prompt, transmitted to a provider, reflected in a model result, compared with another passage, and finally accepted by a human editor. Those events have different evidential weight. Nor should the passage a human last viewed be equated with the last passage offered to a model. The reader's bookmark and the

machine’s reading history describe different forms of attention. Once that distinction is made, the familiar request—make the model remember the book—begins to look mistaken. What should persist is not an unverifiable claim that something once occupied a machine’s mind. What should persist is a record of what happened to the source in the course of an inquiry, including the points at which the inquiry has not yet taken place.

There is a philosophical temptation to say that reading is more than such a record and to leave the matter there. Of course it is. Human reading involves habits, recollection, bodily time, language, expectation, and circumstances that an execution log does not reproduce. The point of an engineered reading history is not to manufacture human experience. It is to make a narrower kind of accountability possible: a later operation can discover what earlier operations claimed, where their evidence lies, which claims were challenged, and which passages remain untreated, without pretending that the earlier model’s active context still exists. My thesis is that the *continuity of interpretation can survive the discontinuity of the interpreter*. But the qualification is decisive: it can do so only when the recorded continuity remains subordinate to the work and open to correction. Otherwise the system has merely invented a permanent reader who is incapable of changing its mind.

Why a longer context does not settle the question

A language model’s context window is a finite workspace, usually measured in tokens. A large window may allow an application to supply a whole novel in one inference request, but it does not follow that the model will use each part reliably. Liu and colleagues’ study *Lost in the Middle* examined the positional sensitivity of models on multi-document question answering and retrieval tasks; in their tested configurations, relevant material was often harder to use when it appeared in the middle of a long context (Liu et al. 2024). The finding is not a universal theorem about every current or future model, and the evaluated tasks are not equivalent to reading literature. Its methodological lesson is nevertheless durable. Capacity—the number of tokens a provider will accept—and effective use of information are different properties. Benchmarks such as LongBench and RULER extend that

distinction by testing task diversity, aggregation, and multi-hop dependencies rather than equating large input limits with comprehension (Bai et al. 2024; Hsieh et al. 2024).

Now suppose the novel is processed in smaller windows. The model that receives chapter twelve does not thereby possess chapters one through eleven. An application may supply earlier transcripts, summaries, selected quotations, structured observations, or none of them. It may reuse a warm execution process while deliberately isolating each new semantic session. In every case the active prompt is a constructed object, not a historical guarantee. This suggests a strict and useful separation. *Source custody* says which edition and exact bytes the application holds; *reading history* says which operations addressed which ranges; *active context* says what was selected for the present request; *interpretive evidence* says what propositions the operation produced and how they can be checked. A system that collapses these distinctions cannot answer the opening question honestly. Retrieval-augmented generation offers an indispensable partial remedy. In the formulation by Lewis and colleagues, generation is assisted by material retrieved from an external knowledge store rather than depending solely on parametric knowledge encoded during training (Lewis et al. 2020). For a literary task this makes source passages available to an otherwise forgetful model. If a reader asks where a character mentions a key, retrieval can work extremely well. But the most interesting literary questions are often not search queries in disguise. When did an apparently affectionate relationship become one of obligation? Which unresolved question first appeared as a joke, disappeared for several chapters, and returned as a serious accusation? The relevant evidence may be distributed across silence, shifts of address, contradictions, and retrospective reinterpretation. One need not find the word *obligation* to discover a relation of obligation.

NarrativeQA was designed in response to just this limitation: conventional question-answering datasets could reward superficial matching when the real objective was the integration of events, entities, and relationships across narratives (Kočíský et al. 2018). Book-length summarization exposes a related problem. The FABLES study used human judgments of claims from summaries of recent novels and found failures especially around events and character states, often requiring indirect reasoning across the narrative to detect (Kim et al. 2024). These results should not

be read as a verdict that machine literary interpretation is impossible. They mean that the apparent coherence of a synopsis is not an adequate test of fidelity to the work. A persuasive compression can be systematically wrong. The obvious answer—summarize each window and summarize the summaries—also has a peculiar weakness. The meaning of what an early summary omits may not be apparent until much later. A barely mentioned object may become central; an apparently incidental remark may be recast by a final disclosure. Compression economizes context by discarding information before its later relevance is known. A reading system therefore needs the ability to return from a current interpretation to an exact earlier passage. It needs a source address, not merely a shortened account of that passage. It needs to preserve a distinction between what was observed and what was inferred from observation.

The same principle limits what the popular category *deep research* can provide. A research system may plan searches, consult external scholarship, compare witnesses, and produce an admirably documented report. That is helpful when an allusion depends on historical knowledge or an identity is settled by an edition's commentary. Yet a report assembled from search results about a novel is not evidence that the novel itself was read. Research on a work and reading of the work can correct one another; neither should masquerade as the other. The boundary is particularly important where a model recalls a plausible-looking bibliographic reference from its training. A scholarly citation asserts the existence and relevant content of an external document. It must be retrieved or otherwise verified, not merely spoken confidently. Reframe eventually encoded precisely that rule under the name *Referenced Knowledge* (Fountain Coach, "Referenced Knowledge," Chapter 32).

At this point the problem of forgetting becomes more interesting. Imagine that a model has found an identity relation in an earlier chapter, but its next prompt does not contain that result. The model may waste time rediscovering it; worse, it may replace it with a plausible false relation. The instinct to store the relation seems sound. But once stored, what stops the relation from becoming an unchallengeable assertion? How does the system remember *without transferring authority from the book to its own memory*? That is the question at which Reframe's particular development history begins to matter.

A system that had to forget in order to remember

Reframe's history is not a sequence of increasingly elaborate components assembled according to a master plan. Its primary documents preserve reversals, failed readings, corrections, and boundaries drawn after a system had crossed them. The Grounding-first refactoring programme separated the canonical source, the writer's declared interpretive posture, the Storify reading operation, the authored Cut Script, subsequent Continuity assessment, and the acts of composition and publication. In this architecture, *Continuity* was already a term of art for auditing downstream work against a declared basis; it did not mean an indefinitely persistent model conversation. More importantly, the refactor removed a semantic index that had begun to function as an independent authority over the manuscript. Its derived summaries and classifications could influence later decisions without sufficient return to the source (Fountain Coach, "Target Architecture," Chapter 04). The removal was deliberate, not a failure to appreciate the value of memory.

Then the actual reading experiments made the cost of forgetting impossible to ignore. On the opening episode of Joyce's *Ulysses*, *Telemachus*, the application repeatedly asked a small on-device model to perform work the text's own structure could settle. A documented run returned an internal schema type where a meaningful description had been requested, and later produced unsupported assertions concerning Stephen and Mulligan. Another failure made the two characters into one fictitious identity. A sequence of local passage readings yielded observations without carrying the same dramatic question forward. The application was asking a generative instrument to supply names, structures, and facts that either a parser already knew or the source did not authorize (Fountain Coach, "Parse Before You Ask," Chapter 27; "A Beat Is the Question It Raises," Chapter 28).

The correction began with a simple division of intellectual labor. A text declares some facts in its form: chapter headings, lines, speaker markers, and—in structured formats—scenes or stage directions. These should be parsed. Local non-generative language tooling can measure other properties: spans, mentions, candidate names, resemblance. Such measurements can be wrong, particularly across languages or literary idioms, so they remain candidates rather than conclusions. Only then should a generative reader attempt interpretation: what changed, who acted, what remains

open, what is implied rather than stated? Chapter 29’s audit of Apple’s NaturalLanguage framework illustrates why even this middle tier must be restrained. A named-entity recognizer can mistake a liturgical word for a character; an embedding similarity can make unrelated questions appear closer than related ones. The lesson is neither to distrust measurement nor to worship it. It is to decline its claim to answer a question of meaning (Fountain Coach, “NaturalLanguage Measures, Storify Interprets,” Chapter 29).

That separation exposed the instability of another convenient unit: the *beat*. At first a reading pipeline can be tempted to treat its own chunks as dramatic segments. But no law of drama says that a question begins and ends where a context window does. In *Telemachus* a relationship may remain troubled through several scenes; a gesture may alter the force of an earlier line; an unanswered accusation may stop being spoken of without being resolved. The Reframe account therefore shifted from automatically naming small stretches to following the question the story holds open. It distinguished the addressable material encountered in a reading from the interpretive thread that can span several encounters. The writer-facing terminology later favoured *Questions* and *Movements*: the former for what remains unsettled, the latter for grounded changes. Importantly, the disappearance of a question could be described only as silence, not as proof that the question had been answered. That is a subtle refusal of premature closure, and it comes from a specific implementation failure rather than from an abstract preference for ambiguity (Fountain Coach, Chapter 28).

The problem of the cast list followed the same pattern. How could a reader retain the fact that a person had been named under several forms without asking a model to re-infer identities on every passage? The proposed *Living Gazetteer* accumulated source-evidenced vocabulary about the world of the work: people, places, objects, names, and relationships. But it was explicitly forbidden to become another semantic index. The Gazetteer could say which candidate identity had been observed and confirmed under which source addresses; it could not, by its own authority, determine what the story meant or decide the reader’s next action (Fountain Coach, “The Living Gazetteer,” Chapter 30). It was memory under constraint. The constraint was not an inconvenience imposed by governance; it was what distinguished this memory from the index Reframe had removed.

There remained cases the manuscript itself could not settle. Joyce's nicknames provide an instructive example. An identity relation might be obvious to a scholarly reader familiar with the novel but not be stated as an explicit equivalence in the bounded passage a machine has examined. Asking the writer to correct every such omission would turn the reading environment into a tiresome editorial examination. Asking a model to supply the correction from general recollection would hide the difference between manuscript evidence and external knowledge. The project therefore made external reference a first-class evidence type. An edition, annotation, dictionary, or scholarly work could be consulted, recorded, and cited as a reference, but it could never be presented as though the novel itself had said the same thing. A useful reading sometimes depends on another book. That dependency should be visible.

A further step was named *Compiled Knowledge*. Difficult reasoning performed once could produce a durable claim, together with the passage or retrieved source that justified it, its scope, the responsible operation, and a way to revise it. Later local models could consult that knowledge without repeating a costly frontier-model call. In Chapter 31's formulation, the cloud may help *compile* understanding, while a device can reuse what has been established. Yet the compiled artifacts were not supposed to route the user's actions, decide the meaning of the source, or authorize spending; and they were not merely disposable caches. A cache can be replaced without explanation. A corrected interpretive decision has a history. It must be dated, inspectable, and removable without rendering the source unreadable (Fountain Coach, "Compiled Knowledge," Chapter 31). We can now see the contradiction that actually drove the architecture forward. The system had to rid itself of an authoritative derivative corpus, because such a corpus could drift away from what it was meant to illuminate. But a system that began each passage as a blank slate could not sustain an interpretation. The answer was not to restore the old index. It was to separate several responsibilities that the index had once confused: the source's custody and addressability, the operations of reading, the historical evidence of those operations, the bounded reuse of confirmed information, and the explicit comparison of competing or changing claims. The point of this history is not that every named subsystem must survive. It is that the failures forced Reframe to formulate a rule: *the more useful memory becomes, the more precisely its limits of authority must be described*.

This development also explains why the literature of interpretation belongs in the discussion. In Wolfgang Iser's account of the reading process, a text is realized through an activity in time; expectations form, are frustrated, and are revised as a reader proceeds (Iser 1972). Hans-Georg Gadamer's hermeneutics likewise refuses a simple picture in which the meaning of a whole is known before its parts are encountered; the understanding of parts and whole is mutually implicated (Gadamer, *Truth and Method*; see the Stanford Encyclopedia of Philosophy's account of his hermeneutics). Neither philosopher supplied a data structure for Reframe, and invoking them cannot establish that the application behaves like a human reader. Their relevance is more exacting: an architecture that preserves only the latest summary erases the history of revised expectations, while an architecture that preserves everything as equally authoritative cannot distinguish understanding from its mistakes. A reading must be allowed to develop, and its development must remain legible.

Continuity without a continuous interpreter

The name *Reading Continuity* was fixed in Scenario R25 in October 2026, but the underlying operation had an earlier name: *Storify Source Auto*. The distinction between them is architectural, not merely historical. Storify Source Auto is responsible for the governed work of reading a defined source through bounded, semantically relevant passages and reconciling what its operations discover. Reading Continuity is the durable account of those operations, beginning when a real source is admitted to the Bookshelf. One performs reading; the other remembers, with evidence, where and how reading has occurred. A source can therefore enter the Bookshelf with a complete structural map and no semantic reading at all. This is a truthful state, not an empty stage waiting to be decorated with progress. What does *real source* mean in this contract? Not a book cover, a catalogue title, or a URL that might still lead somewhere. The source is an edition and revision with identifiable origin, verified content bytes, and an address system robust enough to support later return. Chapter 85 required stable admission through a declared provider release, manifest and content digests, and verified range operations. Its use of repeated exact reads is a custody safeguard, not a substitute for literary criticism. A deterministic

structure pass can then locate chapters, headings, paragraphs, source lines, or TEI structures before a model is asked to interpret any of them. The model should not be asked how many chapters exist when those chapters are already explicitly encoded. Nor should an arbitrary token limit be allowed to become a theory of where the novel's dramatic units begin and end (Fountain Coach, "Storify Source Auto Reads a Defined Whole," Chapter 85).

This source discipline changes the meaning of the Bookshelf. A work may be original Fountain Coach authorship, a Project Gutenberg edition, a DraCor drama, a TEI text, or another admitted corpus. A Git repository can act as an owning, versioned recovery path for source material and publication artifacts. An OPDS 2.0 feed can describe discoverable publications. TEI can encode documentary and textual structure. None of those mechanisms should be conflated with evidence that a particular semantic operation has read a particular passage, and none automatically establishes redistribution or model-processing permission. Source identity, rights, discoverability, reading history, and publication state are related but separable contracts. This is essential in a library where works from very different origins share a quiet visual presence as book covers (OPDS Community, *OPDS 2.0*; TEI Consortium, *P5 Guidelines*).

A reading operation also needs an identity of its own. It names the source version, exact range, the interpretive instructions and their revision, the executing lane or model, and the operation's terminal outcome. If it produces a claim, the claim carries enough evidence to allow later inspection. If it fails, times out, or is cancelled, the history must not silently label its range as successfully interpreted. A warm process may be reused for efficiency while the semantic context of each range is kept explicit and isolated. Chapter 102 articulates the resulting distinction between an execution session, the bounded source window, and the typed result preserved by FountainStore. MIDI 2.0 serves as the governed command, correlation, and lifecycle seam; it is neither the source text nor an independent model of the text. FountainStore owns durable execution and reading state, while the existing source view remains the authority on the manuscript (Fountain Coach, Chapter 102).

These choices could seem austere until one examines what happened when R25 met a full-length novel. In the 16 October experiments, a verified Book Library edition of *Ulysses* supplied 33,142 physical source lines and eighteen chapter boundaries. Deterministic preparation constructed 1,044 contiguous bounded reading ranges without claiming that any had already been semantically read. An actual Apple Foundation Models session then addressed a small part of the novel and wrote identifiable result evidence. In the persistent native Mac test on 17 October, the recorded model-correlated reading stood at just 23 of 25,748 readable lines. Beside that history the human writer's bookmark could point to line 97. The two positions were visibly different. The system had not reached a grand interpretive success, but it had ceased making a grander claim than its evidence could sustain (Fountain Coach, R25, 16–17 October 2026). The test exposed two more mundane facts that turn out to be theoretically useful. Attempting to send all 33,142 lines through the book compositor at once stalled the rendering path. The correction was not a new meta-physics of books, but a bounded 96-line presentation section inside the existing Fountain Press compositor, retaining the original absolute addresses. Similarly, a genuine operating-system process termination left a Store lease behind; the first relaunch correctly refused to take over that authority. The recovery was repaired through a checked dead-owner lease procedure, not by forging or deleting the lease. A new process could then reconstruct the source edition, recorded semantic result, unresolved next range, and human bookmark. The significant result is not the 23 lines themselves. It is that the facts about those lines did not depend on the survival of the process that had read them.

Yet this demonstration has to be stated at its actual size. It did not prove full-book comprehension of *Ulysses*. It did not prove that a model attended to every token submitted to its provider, that later claims were literarily adequate, or that a cloud-authorized unattended worker had completed the remaining reading. The scenario's own record leaves those gates open. This refusal to promote partial evidence to a total claim is not a weakness in the architecture's presentation; it is one of its strongest achievements. A reading system that states "23 lines correlated, the remainder unresolved" has answered a narrower but more trustworthy question than one that displays "100% read" after mapping a source.

A useful continuity record therefore needs to distinguish source *mapped*, source *offered*, source *engaged with an inspectable result*, interpretations *reconciled*, claims *challenged*, and results *accepted for a declared purpose*. These are evidentiary categories, not a single percentage. The record should be able to tell a reader which version produced an answer and which passages were omitted from the answer's active context. Where a model claims knowledge from a previous reading, the application should know whether that knowledge was actually selected into the current prompt. The purpose is not to manufacture a pseudo-conscious reader whose memory becomes magically continuous. It is to make continuity a property of the governed reading environment.

This requirement also changes the economics of inference. If the source is parsed once and its structural facts are reused, the system need not repeatedly pay a language model to rediscover them. If a difficult identity is settled with a verified external reference, a local model may consult that result without repeating a paid research operation. But neither the desire to economize nor the ability to run a task in the background authorizes transmission of a private manuscript. Reading Continuity can begin automatically at admission as a structure-and-evidence lifecycle; semantic provider work must still obey its own permission, privacy, and spending boundary. *Automatic* is not a single capability. Deterministic preparation, free local reasoning, paid external inference, and outward publication require different kinds of authority.

The reader-facing consequence should be relief rather than administration. The Bookshelf does not need an ever-expanding control panel for reading budgets, context sizes, and run identifiers. The book opens as a book. The original text is visible through its established composition engine; editorial space can tell the reader what has actually been examined and what remains in dispute. A question about the interpretation can lead back to the passage and to the operation that generated the claim. An interruption can be resumed. When previous knowledge is stale, the interface should not quietly restyle it as new. The technical machinery matters only if it enables an intelligible conversation with the work.

A history that can disagree with itself

Persistence alone is a dangerously incomplete answer. An application may preserve every interpretive error faithfully. Indeed, the more efficiently it reuses its earlier mistakes, the more coherent those mistakes can become. The question now changes from *how can a reading continue?* to *how can a continuing reading discover that it was wrong?* This is the point at which Fountain Coach’s two formulations of Drift Calculus become necessary, and where their similarity of name is a genuine conceptual hazard.

The earlier *Drift–Pattern–Reflection* calculus, specified in FountainKit’s “Flow of Reason,” is longitudinal. At one moment the system has an explicit baseline of sentence-level claims, B_n . When the state changes, a typed comparison with the previous baseline gives $D_n = \text{Diff}(B_n, B_{n-1})$; recurring relationships across baselines and their differences give P_n ; reflection upon the changes in view of those patterns gives R_n . The original minimal specification proposed token-based normalization and overlap to identify additions, removals, and changes, followed by clustered patterns and a bounded set of reflective claims. The associated Semantic Daemon design envisaged asynchronous work over a FountainStore corpus, recording drift, patterns, and reflection as separate artifacts (Fountain Coach, *Drift–Pattern Reflection Calculus*; *Semantic Daemon*). The attraction of this formulation becomes clearer in an example. Early in a novel, a character may interpret a friend’s silence as indifference. A later passage makes the silence appear protective; another complicates both explanations. A history that retains only the latest interpretation cannot tell the reader how that interpretation came into being. A typed comparison can show that the basis of a claim changed and where the new evidence entered. But the formalism must be handled more carefully than its minimal implementation suggests. The disappearance of a sentence from a later summary may mean that it has been refuted, forgotten, deprioritized, or expressed in different words. Token overlap cannot decide which. Nor can a clustering algorithm confer thematic meaning on repeated vocabulary. The observable difference between two *records* is not yet a verified transformation in the *story*.

This makes the distinction between **world change** and **reader change** indispensable. In a novel, a character may change her mind. Separately, Reframe may change its mind about what that character believed all along. A new model version may phrase the same proposition differently; a revised grounding stance may alter what counts as relevant evidence. The calculus must not report all these movements as though the fictional world itself had undergone an event. It must retain the provenance of the interpreting apparatus as well as the provenance of the text. A corrected reading of an unchanged passage is a history of interpretation, not an edit to the manuscript. Even a rejected claim can remain useful evidence: it records a tempting error and the grounds on which the error was rejected, while allowing reconsideration if genuinely new evidence appears.

There is an especially interesting relationship here to the literary problem of questions. A reading that treats a question as open at one moment and no longer mentioned at another has registered a change in its observations. It has not thereby proven a narrative resolution. The longitudinal calculus becomes intellectually valuable only when it can preserve that difference. It can say that attention shifted, evidence was added, or an interpretation was challenged. It should not manufacture closure to satisfy the neatness of its output format. This is where the non-integrational principle adopted in Reframe's Grounding doctrine becomes a practical epistemic constraint: a system must not force coherence, causation, or final reconciliation where the work itself leaves the matter unsettled (Fountain Coach, "Grounding as a Given," Chapter 11).

Chapter 110 introduced a second, quite different, calculus. Instead of comparing states separated by time, it governs a cohort of *three* separately identified readings of the *same* source-addressed material. Their operations must be authorized, correlated, and receipted; only after all have reached the required terminal state may a comparative drift field be claimed. Agreement, a two-with-one pattern, and three-way divergence must remain visible rather than being collapsed into a single confidence score. The central epistemic assertion is modest: divergence is a phenomenon to inspect, not a failure to conceal. Two readings can of course be compared, but under this particular governed specification they do not constitute a three-run semantic-drift cohort (Fountain Coach, "Three-Run Semantic Drift Is a Governed Measurement Principle," Chapter 110). Three is a procedural design choice, not a

statistical proof of independence. Three differently numbered executions may share the same model weights, source selection, training biases, and prompt assumptions. Even different models may have overlapping training histories and common errors. A majority interpretation of a literary ambiguity has no automatic claim to truth; it may merely be the most culturally habitual account. Conversely, a minority reading may draw attention to a neglected passage. The proper artifact is therefore the whole comparison—each reading, its evidence and limitations, and the method by which their differences were described—not a numerical verdict detached from its witnesses. Independence must be defined operationally and, where stronger statistical claims are wanted, tested empirically rather than presumed.

There is also an essential implementation qualification. At the checked state, Reframe’s portable **SemanticDriftCalculator** validates exactly three successful child observations with distinct run identities and a shared source identity, and it calculates differences in quantities such as page counts and route-link counts. The corresponding adapter persists the cohort result. That is executable infrastructure for a comparison contract; it is not yet an implemented scholarly instrument capable of resolving disagreements about narrative causality, unreliable narration, or the ethical meaning of a character’s act. Chapter 110 describes a richer governing target than the current calculator demonstrates. Calling both *implemented* would obscure the very distinction between claim and evidence that this essay is defending (Fountain Coach, **SemanticDrift.swift**; **ReframeSemanticDriftCohortStoreAdapter.swift**).

The two calculi do not compete. The longitudinal calculus asks what changed across successive interpretive states and why. The three-run calculus asks what variations appear when separately governed readings address the same material. Both may consume evidence retained by Reading Continuity, and both may produce candidate Compiled Knowledge, but neither is entitled to replace the source. One can be applied incrementally as a reading progresses; the other is better invoked where ambiguity or consequence makes independent readings worth their cost. Requiring three full-length readings of every new book would be a costly ritual masquerading as methodological rigor. The question is where comparison changes what the reader can responsibly know.

What would count as understanding?

The architecture now has a plausible shape, but a plausible shape is not an experimental result. A scientifically serious account has to specify its alternatives and its failure conditions. It should ask whether the same reader-facing improvements might be achieved with a simpler search index, a conventional retrieval pipeline, a longer context, or a series of summaries. The crucial experiment is comparative: use the same source editions and evaluation questions; control as far as possible for model, budget, and permissions; compare unassisted long-context inference, retrieval-augmented generation, hierarchical summaries, source-addressed Reading Continuity, and Reading Continuity with selective drift analysis. Measure not only token costs and completion times but evidence fidelity, detection of contradictions, recovery after interruption, correction effort, and the quality of multi-passage reasoning.

The evaluation should include questions that do not yield to quotation search. Why does the final scene alter the force of an apparently minor earlier remark? Which apparent resolution depends on a narrator whose reliability has been questioned? Where did a recurring question cease to be raised, and does the work ever actually answer it? Such tests should distinguish objectively checkable source claims from genuinely plural interpretation. Kočiský and colleagues' NarrativeQA work establishes the importance of integrated narrative reasoning, but its dataset is not a complete theory of literary merit. FABLES usefully demonstrates that even apparently strong summaries can misrepresent character states and events, and that automated raters themselves may be unreliable (Kočiský et al. 2018; Kim et al. 2024). A Reframe evaluation needs human experts who can inspect source citations and disagree in documented ways, not a hidden oracle that declares one interpretation universally correct.

One should also test a very different property: provenance truthfulness. Does a report that calls a range *mapped* identify actual source bytes and an appropriate parser revision? Does a report that calls a range *model-engaged* have a recorded execution and result, rather than a spinner, a fabricated note, or a completed request of a different scope? Does a later claim identify the source passage actually offered to the model that produced it? Can the application distinguish evidence from an original

manuscript, a retrieved secondary source, and an explicit human editorial decision? The W3C PROV family supplies a general vocabulary of entities, activities, and agents for explaining derivation; Reframe need not implement PROV-O verbatim to benefit from that discipline (W3C 2013). These questions can be tested at the level of receipts and reconstructed state, independent of whether a particular interpretation pleases the reader.

Interpretive memory introduces a subtler class of tests. Suppose one source passage is edited while a thousand others remain unchanged. Does the system invalidate precisely those derived claims whose support depends on the changed passage, or does it blindly discard the whole reading? Does it accidentally continue presenting earlier results as current? If a human rejects an invented identity, is that judgment preserved with its evidence and authorship? If a later edition makes the rejected reading tenable, can the system reopen the question without pretending that the older correction never happened? A durable record becomes trustworthy only if it can describe its own obsolescence. That is why Compiled Knowledge is neither a second manuscript nor an automatically regenerated cache.

A full recovery experiment is equally necessary. Destroy a model process in the middle of an operation; then relaunch with the same authorized Store. A successful result should reconstruct exactly what was committed, refuse unsupported completions, preserve the human bookmark separately, and resume from a justified unresolved range. R25 now has evidence for part of this experiment on a real Mac and a real novel. The remaining conditions—whole-work semantic processing, independent delivery evidence, robust multi-passage evaluation, writer approval, iPad presentation, and permissioned unattended activity—cannot be inferred from that success. The proper Definition of Done is layered: custody, structure, execution, continuity, interpretive quality, and publication each need their own accepted evidence. The fact that they can be named independently is a methodological advantage.

We should also remain alert to the politics of an apparently neutral information system. An author's work may be private; a public-domain text may exist in an edition subject to separate conditions; a scholarly comment may have its own rights. Neither the ability to ingest nor the ability to reason about a source authorizes publi-

cation of its words. Git ownership makes recovery and authorship provenance tractable, but a repository is not automatically the legal or epistemic authority over all the materials it contains. Publication of an evidence cohort, as proposed in Chapter 111, should expose appropriately sanitized proofs of what a system actually established, not private source content or an implementation promise disguised as completed acceptance. The design of a public Fountain Coach essay and the acceptance of a live Reframe reading are different acts (Fountain Coach, “Evidence Cohorts Are the Publication Units,” Chapter 111).

A more difficult objection remains. Has the whole architecture done anything more than build an elaborate notebook around an imperfect reader? In one sense, yes: it has indeed built an external record. The word *notebook* becomes inadequate only if the record changes what operations can be performed and justified. A properly source-addressed question can be resumed across an interruption. A claim can be traced to the exact edition and passage that supported it. An error can be corrected without losing the history of why it was once thought plausible. The relevant evidence can be reintroduced into a bounded prompt, and competing readings can be compared without overwriting one another. These are executable affordances, not decorative annotations. But the objection remains healthy. If a simpler system delivers the same accountable results at lower cost, the larger architecture has not earned its complexity. Reframe must be willing to find that out.

The reading remains open

What the opening question sought was not really a count of completed model requests. It sought grounds for trusting an answer about a work that exceeds any one encounter with it. The temptation was to overcome finitude by adding context, transcripts, summaries, and persistent memory until the model could be spoken of as though it had lived continuously with the book. Reframe’s more promising path is almost the opposite. It accepts finitude, isolates the operation, and makes the history of those operations durable outside the model. The model does not have to remain the same reader tomorrow. The source, the evidence, the disputed questions, and the possibility of correction have to remain available. This does not make literary interpretation mechanical. If anything, it makes the difference between measure-

ment and understanding harder to ignore. The parser can tell where a chapter begins; it cannot tell whether the beginning is a confession. The model can propose a connection; its fluency cannot certify the connection. Three models can disagree; the arithmetic of their disagreement cannot decide which interpretation a writer should adopt. A stored claim can be consulted a hundred times without becoming a fact about the manuscript. Even the most thorough source ledger can show that an operation occurred without proving what an artificial intelligence *experienced*, a claim for which there is neither a necessary product requirement nor a suitable current test.

The gain is therefore not the fiction of a final reader who remembers everything. It is the possibility of a reading whose history is capable of bearing criticism.

Reframe’s actual development—its removal of a misleading index, its failed encounters with *Telemachus*, the gradual formation of Questions, Gazetteer, Compiled Knowledge and Referenced Knowledge, the disciplined source admission of *Storify Source Auto*, the bounded proof of R25, and the two nonidentical Drift Calculi—belongs to the argument because it shows how one problem repeatedly changed its shape. First the system needed to stop inventing. Then it needed to stop forgetting. Finally, it needed to learn that remembering without the ability to revise is merely another way to be wrong.

A book is not its answer. Nor is the history of its reading identical to the book. But between a transient inference and an immutable page there can be a responsible, ongoing practice of interpretation: capable of returning to its evidence, distinguishing a question from its resolution, and refusing to hide the places where no reading has yet occurred. Its continuity need not be the continuity of a single intelligence. It must be the continuity of an inquiry that can explain what it has done, what it has not done, and why the next passage might require it to begin again.

Documentary lineage and references

The project documents cited here are primary evidence for Fountain Coach’s own design decisions, experiments, and reported implementation checkpoints. They do not independently validate its uncompleted semantic claims. External publications provide separate conceptual or empirical

evidence. Dates describe known documentary or publication states, not necessarily the first occasion on which an idea was conceived. This essay is a substantive reconstruction and critique, not an accepted governance chapter or a claim of finished product acceptance.

Fountain Coach primary documents

Fountain Coach. *Target Architecture*, Reframe-Refactoring, Chapter 04. <https://github.com/Fountain-Coach/Reframe-Refactoring/blob/main/docs/04-target-architecture.md>

Fountain Coach. *Grounding as a Given — The Canonical Manifesto and Conversational Authoring*, Reframe-Refactoring, Chapter 11. <https://github.com/Fountain-Coach/Reframe-Refactoring/blob/main/docs/11-grounding-as-a-given.md>

Fountain Coach. *Parse Before You Ask*, Reframe-Refactoring, Chapter 27. <https://github.com/Fountain-Coach/Reframe-Refactoring/blob/main/docs/27-parse-before-you-ask.md>

Fountain Coach. *A Beat Is the Question It Raises*, Reframe-Refactoring, Chapter 28 and 29 July 2026 amendment. <https://github.com/Fountain-Coach/Reframe-Refactoring/blob/main/docs/28-a-beat-is-the-question-it-raises.md>

Fountain Coach. *NaturalLanguage Measures, Storify Interprets*, Reframe-Refactoring, Chapter 29. <https://github.com/Fountain-Coach/Reframe-Refactoring/blob/main/docs/29-natural-language-measures-storify-interprets.md>

Fountain Coach. *The Living Gazetteer*, Reframe-Refactoring, Chapter 30. <https://github.com/Fountain-Coach/Reframe-Refactoring/blob/main/docs/30-the-living-gazetteer.md>

Fountain Coach. *Compiled Knowledge — the Cloud Teaches, the Device Remembers*, Reframe-Refactoring, Chapter 31. <https://github.com/Fountain-Coach/Reframe-Refactoring/blob/main/docs/31-compiled-knowledge.md>

Fountain Coach. *Referenced Knowledge — What the World Knows Outside the Book*, Reframe-Refactoring, Chapter 32. <https://github.com/Fountain-Coach/Reframe-Refactoring/blob/main/docs/32-referenced-knowledge.md>

Fountain Coach. *Drift–Pattern Reflection Calculus (Minimal Spec)*, FountainKit. <https://github.com/Fountain-Coach/FountainKit/blob/main/Docs/Flow-of-Reason-Baseline-Drift-Patterns-Reflection.md>

Fountain Coach. *Drift–Pattern–Reflection Calculus*, MIDI2 integration documentation. <https://github.com/Fountain-Coach/midi2-gpu-fabric/blob/main/docs/calculus-drift-pattern-reflection.md>

Fountain Coach. *Semantic Daemon*, MIDI2 integration documentation. <https://github.com/Fountain-Coach/midi2-gpu-fabric/blob/main/docs/semantic-daemon.md>

Fountain Coach. *Storify Source Auto Reads a Defined Whole*, Reframe-Refactoring, Chapter 85. <https://github.com/Fountain-Coach/Reframe-Refactoring/blob/main/docs/85-storify-source-auto-reads-a-defined-whole.md>

Fountain Coach. *Semantic Inference Execution, Session, and Latency Governance*, Reframe-Refactoring, Chapter 102. <https://github.com/Fountain-Coach/Reframe-Refactoring/blob/main/docs/102-semantic-inference-execution-session-and-latency-governance.md>

Fountain Coach. *Three-Run Semantic Drift Is a Governed Measurement Principle*, Reframe-Refactoring, Chapter 110. <https://github.com/Fountain-Coach/Reframe-Refactoring/blob/main/docs/110-three-run-semantic-drift-calculus.md>

Fountain Coach. *Evidence Cohorts Are the Publication Units of the Fountain Coach Estate*, Reframe-Refactoring, Chapter 111. <https://github.com/Fountain-Coach/Reframe-Refactoring/blob/main/docs/111-evidence-cohorts-as-publication-units.md>

Fountain Coach. *R25 — Reading Continuity Begins at Bookshelf Source Admission*, scenario and dated implementation evidence, 16–17 October 2026. <https://github.com/Fountain-Coach/Reframe/blob/main/scenarios/025-bookshelf-reading-continuity.md>

Fountain Coach. **SemanticDrift.swift**, current portable calculator. <https://github.com/Fountain-Coach/midi2-gpu-fabric/blob/main/apps/modernization-studio/Packages/ReframePortable/Sources/ReframeCore/SemanticDrift.swift>

Fountain Coach. **ReframeSemanticDriftCohortStoreAdapter.swift**, Store persistence adapter. <https://github.com/Fountain-Coach/midi2-gpu-fabric/blob/main/apps/modernization-studio/Sources/ReframeCore/ReframeSemanticDriftCohortStoreAdapter.swift>

External scholarship and standards

Bai, Yushi, et al. 2024. “LongBench: A Bilingual, Multitask Benchmark for Long Context Understanding.” *Proceedings of ACL 2024*, 3119–3137. <https://doi.org/10.18653/v1/2024.acl-long.172>

Gadamer, Hans-Georg. 1960. *Wahrheit und Methode [Truth and Method]*. Tübingen: Mohr. See also Stanford Encyclopedia of Philosophy, “Hans-Georg Gadamer,” <https://plato.stanford.edu/entries/gadamer/>

Hsieh, Cheng-Ping, et al. 2024. “RULER: What’s the Real Context Size of Your Long-Context Language Models?” arXiv:2404.06654. <https://arxiv.org/abs/2404.06654>

Iser, Wolfgang. 1972. “The Reading Process: A Phenomenological Approach.” *New Literary History* 3 (2): 279–299. <https://doi.org/10.2307/468316>

Kim, Yekyung, et al. 2024. “FABLES: Evaluating Faithfulness and Content Selection in Book-Length Summarization.” arXiv:2404.01261. <https://arxiv.org/abs/2404.01261>

Kočiský, Tomáš, et al. 2018. “The NarrativeQA Reading Comprehension Challenge.” *Transactions of the Association for Computational Linguistics* 6: 317–328. https://doi.org/10.1162/tacl_a_00023

Lewis, Patrick, et al. 2020. “Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks.” *Advances in Neural Information Processing Systems* 33. <https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html>

Liu, Nelson F., et al. 2024. “Lost in the Middle: How Language Models Use Long Contexts.” *Transactions of the Association for Computational Linguistics* 12: 157–173. https://doi.org/10.1162/tacl_a_00638

OPDS Community. *OPDS Catalog 2.0*, living specification. <https://specs.opds.io/opds-2.0>

Text Encoding Initiative Consortium. *TEI P5: Guidelines for Electronic Text Encoding and Interchange*. <https://tei-c.org/guidelines/p5/>

W3C Provenance Working Group. 2013. *PROV-O: The PROV Ontology*. W3C Recommendation, 30 April. <https://www.w3.org/TR/prov-o/>