Paperbot Parliament: A Multi-Agent Framework for Distinction-Centered Document Analysis

Concept, Formal Theory, Algorithm, and First Experimental Run

Flyxion

Independent Researcher

June 14, 2026

Abstract

We describe Paperbot Parliament, a multi-agent document analysis system implemented as a Unix shell pipeline over a local large language model. The system departs from conventional multi-agent summarization by assigning each agent a constitutionally distinct analytical role: Claims Extractor, Nitpicker, Skeptic, Archivist, Synthesist, and Admissibility Auditor. Rather than compressing a document into a summary, the parliament treats each document chunk as the input to a structured adversarial process whose primary outputs are distinctions of the form $X \neq Y$. We develop a formal theory of distinction-centered analysis, introducing the distinction space (C, D), distinction-preservation score P, admissibility distortion $\mathcal{A}_\pi$, argument admissibility score α, and reconstruction fidelity F. We prove a Parliament Stability Theorem showing that iterated revision cycles are non-increasing in the Lyapunov function $V(A) = \Delta(A) + \lambda(1 - R)$, and a Parallel Independence Proposition showing that the architecture structurally prevents recursive summarization collapse. We report the results of the first experimental run on a theoretical document, in which 29 distinct $X \neq Y$ pairs emerged spontaneously from a single 250-line chunk. The complete implementation is released as a self-contained Bash script requiring only Ollama and any locally hosted language model.

Contents

1. Introduction

The dominant paradigm for automated document analysis is summarization. A summarizer receives a document and returns a shorter text that preserves the most salient information according to some implicit measure of salience. The measure is usually semantic proximity to training data, frequency of term co-occurrence, or position in the document structure. In each case the output is a compression: a shorter text that stands in for a longer one.

Compression is useful. It is also dangerous for a specific and rarely articulated reason. The information destroyed by compression is not uniformly irrelevant. Compression tends to preserve consensus and discard boundary cases; it tends to preserve the central tendency of a field and discard the anomalies. Most importantly for the purposes of the present paper, it tends to preserve facts and discard distinctions.

A distinction is not a fact. A fact asserts that something is the case: *Memory involves the hippocampus.* A distinction asserts that two things are not the same kind of thing: *Memory* $\neq$ *Storage.* Facts describe what is. Distinctions describe what must not be confused.

Once this asymmetry is accepted, a cascade of consequences follows. Knowledge storage becomes distinction preservation. Summarization becomes distinction collapse. Reasoning becomes navigation between distinctions. And failure becomes the loss of a distinction that a system could not afford to lose.

The theoretical background for this claim comes from the CLIO framework (Constraint-Leveraged Inference and Optimization), which holds that the fundamental unit of intelligence is not the stored fact but the preserved actionable difference. A system that cannot distinguish reachable futures from stable states, admissible transitions from reachable states, or prediction accuracy from preservation of actionable differences, is not merely imprecise—it is incapable of the operations that depend on those separations.

Paperbot Parliament is an experimental platform for testing whether a structured multi-agent pipeline can recover distinctions from unstructured text rather than merely compressing that text into a shorter summary. The answer from the first experimental run is affirmative: the parliament spontaneously reconstructed a substantial distinction algebra from a theoretical document without being explicitly instructed to produce one. The present paper describes the formal theory, the algorithm, the architecture, the first experimental run, and the theoretical interpretation of that result.

2. Motivation

2.1. The Limits of Summarization

A standard large-language-model summarizer is a function $f : \mathcal{D} \to \mathcal{D}'$ where $|\mathcal{D}'| \ll |\mathcal{D}|$ under some token-count norm. The function is trained to maximize agreement between its output and human-produced summaries of held-out documents. Nothing in this training objective explicitly preserves distinctions. A summary of a paper that argues $X \neq Y$ may read: *The paper discusses the relationship between X and Y*—semantically adjacent but representationally catastrophic, because it collapses the very boundary the paper was written to establish.

This failure mode is not merely a defect of current models that better training will eventually correct. It is structural. A system optimized to produce text resembling summaries will converge on whatever features summaries tend to have, and summaries tend to avoid sharp negations, boundary claims, and ontological separations in favor of affirmative, encyclopedic prose.

2.2. Distinctions as Units of Reasoning

Consider the following pairs extracted from the first Paperbot Parliament run, reported in Section 7:

DISTINCTION: Memory $\neq$ Storage
DISTINCTION: Reachability $\neq$ Classification
DISTINCTION: Health $\neq$ Contents
DISTINCTION: Failure $\neq$ Temporary Setback
DISTINCTION: Planning $\neq$ Linear Path
DISTINCTION: Intelligence $\neq$ Information Processing

None of these is a fact. Each asserts that two categories which might be conflated must not be conflated. Each constrains subsequent reasoning by closing off a category error. A fact tells you what is. A distinction tells you what you must not confuse on the way to acting.

2.3. Compression Versus Reconstruction

The RSVP framework (Relativistic Scalar-Vector Plenum) characterizes a system by a triple $(\Phi, \mathbf{v}, S)$ of field capacity, transport, and constraint. In this framework, compression is a reduction in the capacity field Φ that may or may not preserve the structure of the constraint surface S. The critical question is not how much information is retained but whether the boundaries in the original document's concept space survive in the compressed representation.

A multi-agent parliament is one architectural response to this problem. Instead of applying a single compression function to a document, the parliament applies a set of analytically distinct functions to the same input, each designed to preserve a different class of structure. The output is not a shorter document but a richer analytical artifact that makes visible what a summarizer would have silently discarded.

3. Theoretical Background

3.1. RSVP: Relativistic Scalar-Vector Plenum

The RSVP framework models a system as a triple $(\Phi, \mathbf{v}, S)$ where $\Phi : \mathcal{X} \to \mathbb{R}$ is a scalar capacity field, $\mathbf{v} : \mathcal{X} \to T\mathcal{X}$ is a transport vector field, and $S \subseteq \mathcal{X}$ is the constraint surface. A system is healthy to the extent that many futures remain reachable within the constraint surface. The health of a system is therefore not a property of its contents but of its future reachability under its current constraints, yielding: system health $\neq$ system contents.

3.2. Admissibility and CLIO

The admissibility framework distinguishes reachability from admissibility. Reachability asks: what states are accessible from the current state? Admissibility asks: which transitions are permitted under the current constraints? These are not the same question. A state may be reachable via an inadmissible transition, or unreachable despite being admissible.

CLIO operationalizes admissibility at the representational level. Its central claim is that a representation should preserve the distinctions relevant to future admissible action rather than maximizing prediction accuracy. The distinction between CLIO and standard machine learning is itself a CLIO-type distinction: prediction accuracy $\neq$ preservation of actionable differences.

3.3. Repair Theory and Reconstruction

Repair Theory argues that intelligent systems are not primarily storage systems but reconstruction systems. Memory retrieval is more accurately described as trajectory re-entry: the system does not recall a stored object but re-enters a dynamical trajectory converging on the relevant state. Forgetting is not the loss of stored information but the loss of reachability of a trajectory. Memory $\neq$ Storage.

A multi-agent parliament is a reconstruction engine for conceptual structure. It does not store the document; it reconstructs the distinction graph that the document was written to establish, with each constitutional office following a different analytical trajectory through the same conceptual space.

4. Formal Theory

4.1. Distinction Space

Definition 4.1 (Distinction Space). *Let C be a finite set of concepts occurring in a document corpus. Define a* distinction relation $D \subseteq C \times C$ *where $(a, b) \in D$ if and only if the corpus asserts the non-equivalence of a and b. The pair $G_D = (C, D)$ is the* distinction graph *of the corpus.*

The distinction ledger produced by the parliament is a finite approximation to G_D. Each entry DISTINCTION: $X \neq Y$ corresponds to a directed edge $(X, Y) \in D$. The parliament's primary task is: given a document $\mathcal{T}$ with ground-truth distinction graph G_D^*, produce an approximation G_D maximizing $|D \cap D^*|$.

Definition 4.2 (Distinction-Preservation Score). *Let D_{in} be the distinction relation of a source document and D_{out} the distinction relation extracted by an analysis procedure. The* distinction-preservation score *is*

$$P = \frac{|D_{\text{out}} \cap D_{\text{in}}|}{|D_{\text{in}}|}.$$

Definition 4.3 (Distinction Entropy). *Let $H_D = \log |D|$ denote the* distinction entropy *of a representation, and define the* compression factor $K = H_D^{\text{out}} / H_D^{\text{in}}$.

Principle 4.1 (Distinction Preservation). *Two analyses with equal semantic coverage may differ substantially in distinction entropy. Summary quality $\neq$ distinction quality.*

4.2. Cumulative Distinction Ledger

The distinction ledger grows monotonically across chunks. Let $D(C_i)$ denote the set of $X \neq Y$ pairs extracted from claims C_i of chunk i. After n chunks, the ledger satisfies

$$L_n = \bigcup_{i=1}^{n} D(C_i),$$

with the update rule $L_{t+1} = L_t \cup D(C_{t+1})$.

The ledger is therefore not a report but a cumulative approximation to the corpus-wide distinction graph G_D. Each new document contributes new edges to (C, L_n), and the parliament's cross-session memory is encoded in the growth of L_n over time.

4.3. Weighted Distinction Preservation

The unweighted distinction graph $G_D = (C, D)$ records which conceptual boundaries have been extracted but does not distinguish central distinctions from trivial ones. This motivates a weighted extension.

Definition 4.4 (Weighted Distinction Graph)**.** *A* weighted distinction graph *is a triple $G_D^w = (C, D, w)$ where $w : D \to \mathbb{R}_{\geq 0}$ assigns each distinction a salience weight. The value $w(a, b)$ measures the downstream importance of preserving the distinction $a \neq b$. The* weighted distinction-preservation score *is*

$$P_w = \frac{\sum_{d \in D_{\text{out}} \cap D_{\text{in}}} w(d)}{\sum_{d \in D_{\text{in}}} w(d)}.$$

Proposition 4.1 (Weighted Preservation Dominance)**.** *If all distinctions have equal weight then $P_w = P$. If the distinctions that are preserved have average weight at least as large as the corpus-wide average, then $P_w \geq P$.*

Proof. For uniform weight $w(d) = k$ for all $d \in D_{\text{in}}$, numerator and denominator both acquire a factor of k, so $P_w = P$. For the second claim, let $\bar{w}_{\text{pres}} = \sum_{d \in D_{\text{out}} \cap D_{\text{in}}} w(d) / |D_{\text{out}} \cap D_{\text{in}}|$ and $\bar{w}_{\text{all}} = \sum_{d \in D_{\text{in}}} w(d) / |D_{\text{in}}|$. If $\bar{w}_{\text{pres}} \geq \bar{w}_{\text{all}}$, then

$$P_w = \frac{\sum_{d \in D_{\text{out}} \cap D_{\text{in}}} w(d)}{\sum_{d \in D_{\text{in}}} w(d)} \geq \frac{|D_{\text{out}} \cap D_{\text{in}}|}{|D_{\text{in}}|} = P. \qquad \square$$

This addresses the triviality problem raised in peer review: a distinction-centered system should not merely maximize the number of extracted distinctions but should preferentially preserve those whose collapse would most damage future reasoning.

4.4. Historical and Active Distinction Memory

Revision cycles may reject or alter claims. If a claim is removed by the Revision Office, the status of the distinction extracted from that claim becomes ambiguous. A coherent parliament therefore maintains two ledgers.

Definition 4.5 (Historical and Active Ledgers)**.** *The* historical ledger $L_H(t)$ *records every distinction ever extracted:*

$$L_H(t+1) = L_H(t) \cup D(C_{t+1}).$$

The active ledger $L_A(t)$ *records only distinctions currently supported by admissible claims:*

$$L_A(t+1) = \text{Reconcile}(L_A(t),\ D(C_{t+1}),\ V_{t+1}),$$

where V_{t+1} *is the verdict set issued by the parliament. Each distinction carries a status* $\text{status}(d) \in \{\text{ACTIVE}, \text{REJECTED}, \text{SUPERSEDED}\}$.

Proposition 4.2 (Historical Monotonicity and Active Revision)**.** *The historical ledger is monotone:* $L_H(t) \subseteq L_H(t+1)$*. The active ledger need not be monotone:* $L_A(t) \not\subseteq L_A(t+1)$ *in general.*

Proof. The historical update rule is set union, so $L_H(t) \subseteq L_H(t+1)$ holds immediately. The active ledger is updated by a reconciliation operator that may remove or supersede distinctions when their supporting claims fail parliamentary review. Therefore there may exist $d \in L_A(t)$ such that $d \notin L_A(t+1)$. □

This separates memory from belief. The parliament may remember that a distinction was once proposed without continuing to treat it as currently admissible, resolving the apparent inconsistency between the monotonic ledger and the revisionary claim set noted in peer review.

4.5. Distinction Novelty and Saturation

A cumulative ledger permits a simple measure of the conceptual contribution of each new document.

Definition 4.6 (Distinction Novelty)**.** *The* distinction novelty *of claims* C_t *with respect to the historical ledger* $L_H(t-1)$ *is*

$$\nu(C_t) = |D(C_t) \setminus L_H(t-1)|.$$

The normalized novelty score is

$$\hat{\nu}(C_t) = \frac{\nu(C_t)}{|D(C_t)|}, \qquad |D(C_t)| > 0.$$

A corpus sequence is distinction-saturated *if* $\lim_{t\to\infty} \hat{\nu}(C_t) = 0$.

Proposition 4.3 (Finite Saturation). *If the global distinction universe D^* is finite and every extracted distinction belongs to D^*, then the historical novelty score eventually reaches zero.*

Proof. Since D^* is finite and $L_H(t+1) = L_H(t) \cup D(C_{t+1})$, the ledger can acquire at most $|D^*|$ distinct elements. Each time $\nu(C_t) > 0$, at least one new distinction is added to L_H. This can happen at most $|D^*|$ times. Therefore, after finitely many novelty-producing steps, no unseen distinctions remain, and $D(C_t) \setminus L_H(t-1) = \varnothing$ for all later t, giving $\hat{\nu}(C_t) = 0$ eventually. □

Proposition 4.3 gives the parliament a measurable learning curve. Early documents in a domain should produce high novelty; later documents should increasingly produce confirmations, refinements, or contradictions rather than entirely new distinctions. Saturation would indicate that the parliament has exhausted the conceptual boundaries available in the corpus.

4.6. The Auditor as Proof Graph Estimator

The Admissibility Auditor has an elegant interpretation that makes it mathematically distinct from every other constitutional office.

Definition 4.7 (Minimal Admissible Closure). *Let $A = (V, E)$ be the argument graph and $B \subseteq E$ the set of missing bridges detected by the Auditor. Define $E^* \supseteq E$ as the minimal edge set required to make every inferential step in A admissible. The Auditor is approximating the* missing bridge set

$$B \approx E^* \setminus E.$$

The Auditor is not finding falsehoods: it is estimating the gap between the argument graph as presented and the minimal admissible closure of that argument. The Skeptic asks whether claims are true; the Nitpicker asks whether claims are well-formed; the Auditor asks what edges must be added to A before the argument becomes admissible at all. This makes the Auditor the office most closely aligned with the admissibility framework's central question: not *is this correct?* but *what is missing for this to be evaluable?*

Definition 4.8 (Admissibility Distortion). *Let $\pi : \mathcal{X} \to \mathcal{M}$ be a projection from conceptual space to representation space, and let $R(x) \subseteq \mathcal{X}$ denote the reachable futures from x. The* admissibility distortion *of π is*

$$\mathcal{A}_\pi = \frac{\left|\{(x_1, x_2) : \pi(x_1) = \pi(x_2),\ R(x_1) \neq R(x_2)\}\right|}{|\mathcal{X}|^2}.$$

$\mathcal{A}_\pi$ measures how often π collapses two states with different accessible futures into a single representational point. Summarization is a projection π_S from document-concept space to summary space. The parliament's purpose, stated precisely, is to minimize $\mathcal{A}_\pi$ rather than minimize $|\mathcal{M}|$.

4.7. Argument Admissibility

Definition 4.9 (Argument Admissibility Score)**.** *Let $A = (V, E)$ be a directed graph where vertices V are claims and edges E are inferential steps. Let $B \subseteq E$ be the set of missing bridges detected by the Auditor. The* argument admissibility score *is*

$$\alpha = 1 - \frac{|B|}{|E|}.$$

When $\alpha = 1$, every inferential step is bridged and the argument is fully admissible. When $\alpha = 0$, every step contains an unresolved gap. The Auditor's constitutional role is to estimate $|B|$, providing a lower bound on $1 - \alpha$.

4.8. Reconstruction Fidelity

Definition 4.10 (Reconstruction Fidelity)**.** *Let T be the conceptual trajectory of a document, $S(T)$ a conventional summary, and $P(T) = \{N, Sk, Ar, Sy, Au\}(C)$ the parliament's analytical family. Let $R(\cdot)$ be a reconstruction operator. The* reconstruction fidelity *of a procedure Q is*

$$F(Q(T)) = \frac{|R(Q(T)) \cap T|}{|T|}.$$

Conjecture 4.1 (Parliament Reconstruction Advantage)**.** For theoretical documents whose primary conceptual content consists of distinctions,

$$F(P(T)) > F(S(T)).$$

We distinguish Conjecture 4.1 from the proved results that follow. The conjecture requires a formal reconstruction operator R and an empirical comparison across a held-out corpus, outlined in Section 10. The first experimental run provides qualitative evidence in its favor.

4.9. Distinction Recovery

The parallel structure of the parliament immediately implies a useful bound on the parliament's distinction-recovery capacity.

Proposition 4.4 (Distinction Recovery). *Let $\mathcal{P}(C) = \{B_1(C), \ldots, B_n(C)\}$ be the parliamentary family and let $D(Q)$ denote the set of distinctions recovered by analysis procedure Q. Then*

$$|D(\mathcal{P}(C))| \geq \max_i |D(B_i(C))|.$$

Proof. By definition, $D(\mathcal{P}(C)) = \bigcup_i D(B_i(C))$, so $D(\mathcal{P}(C)) \supseteq D(B_i(C))$ for every i, giving $|D(\mathcal{P}(C))| \geq |D(B_i(C))|$ for all i. Taking the maximum over i completes the proof. □

Proposition 4.4 formalizes why a parliament is more valuable than any single office: it can recover at least as many distinctions as its most productive member, and typically more, since different offices are sensitive to different classes of distinction failure.

When revision cycles are active, the Revision Office computes

$$C_{t+1} = R\big(C_t,\ N(C_t),\ Sk(C_t),\ Au(C_t)\big),$$

where N, Sk, Au are the Nitpicker, Skeptic, and Auditor transforms respectively. This is a fixed-point iteration on the claim set. Each cycle feeds structural, evidential, and inferential defects back into a repaired claims file, so the parliament converges toward a claim set that has survived all three critical offices.

4.10. Lyapunov Stability

Definition 4.11 (Parliament Lyapunov Function). *Let* $\mathrm{pass}(A)$, $\mathrm{fail}(A)$, $\mathrm{rev}(A)$ *be the per-verdict claim counts,* $n(A)$ *their sum. Define*

$$\begin{aligned} \Delta(A) &= \frac{\mathrm{fail}(A) + \mathrm{rev}(A)}{n(A)}, & R(A) &= \frac{\mathrm{pass}(A)}{n(A)}, \\ V(A) &= \Delta(A) + \lambda\big(1 - R(A)\big), & \lambda &> 0. \end{aligned}$$

Proposition 4.5 (Parliament Stability). *If every revision cycle (a) removes at least one claim with an unresolved admissibility violation and (b) introduces no new admissibility violations, then $V(A_{t+1}) \leq V(A_t)$ for all t.*

Proof. By (a), $\mathrm{fail}(A_{t+1}) + \mathrm{rev}(A_{t+1}) \leq \mathrm{fail}(A_t) + \mathrm{rev}(A_t) - 1$, so $\Delta(A_{t+1}) \leq \Delta(A_t)$. By (b), no new failures are introduced, giving $R(A_{t+1}) \geq R(A_t)$ and thus $\lambda(1 - R(A_{t+1})) \leq \lambda(1 - R(A_t))$. Therefore

$$V(A_{t+1}) = \Delta(A_{t+1}) + \lambda(1 - R(A_{t+1})) \leq \Delta(A_t) + \lambda(1 - R(A_t)) = V(A_t). \quad \square$$

Corollary 4.1 (Parliament Convergence)**.** *Under the conditions of Proposition 4.5, the revision cycle converges to a fixed point A^* at which $\Delta(A^*) = 0$ and $R(A^*) = 1$.*

Proposition 4.5 is a *conditional convergence result*, not a behavioral guarantee for any particular LLM. It characterizes the dynamics of any revision operator satisfying assumptions (a) and (b). Whether a specific LLM implementation satisfies those conditions is an empirical question. The conditions could fail if, for example, the Revision Office introduces new category errors while removing old ones. The Lyapunov log records V at each cycle precisely so that empirical departures from monotone decrease can be detected and diagnosed. A persistent increase in V across cycles would indicate that the revision operator is violating assumption (b), and the run should be flagged as divergent.

5. Algorithm and Architecture

5.1. Parliament Algorithm

Algorithm 1 gives the complete procedure. It serves as the authoritative behavioural specification; the Bash implementation is a faithful rendering of this pseudocode over standard Unix text streams.

The parliament operator for a single chunk with claims C is:

$$\mathcal{P}(C) = \{N(C),\ Sk(C),\ Ar(C),\ Sy(C),\ Au(C)\}.$$

This set-valued form distinguishes the parliament from a composed pipeline.

5.2. Parallel Independence and Anti-Collapse

The central structural property of the algorithm is that all offices receive identical input.

Proposition 5.1 (Parallel Independence)**.** *If every office B_i receives the same claims file C and no office receives another office's output, then recursive summarization collapse cannot occur within a single cycle.*

Proof. Each office computes $B_i(C)$ from the shared source C. No office computes $B_i(B_j(C))$ for $i \neq j$. Therefore the composition $B_n(\cdots(B_2(B_1(C)))\cdots)$, which is the recursive structure responsible for collapse, never arises. Instead the parliament produces the independent family $\{B_1(C), B_2(C), \ldots, B_n(C)\}$, in which information loss does not compound across offices. □ □

The distinction between $B_n \circ \cdots \circ B_1(C)$ (recursive compression) and $\{B_i(C)\}$ (parallel family) is the single most important structural property of the architecture, and it explains the empirical observation that the six office outputs remained analytically differentiated in the first run.

5.3. Pipeline Diagram

Figure 1 gives the activity diagram of the parliament pipeline.

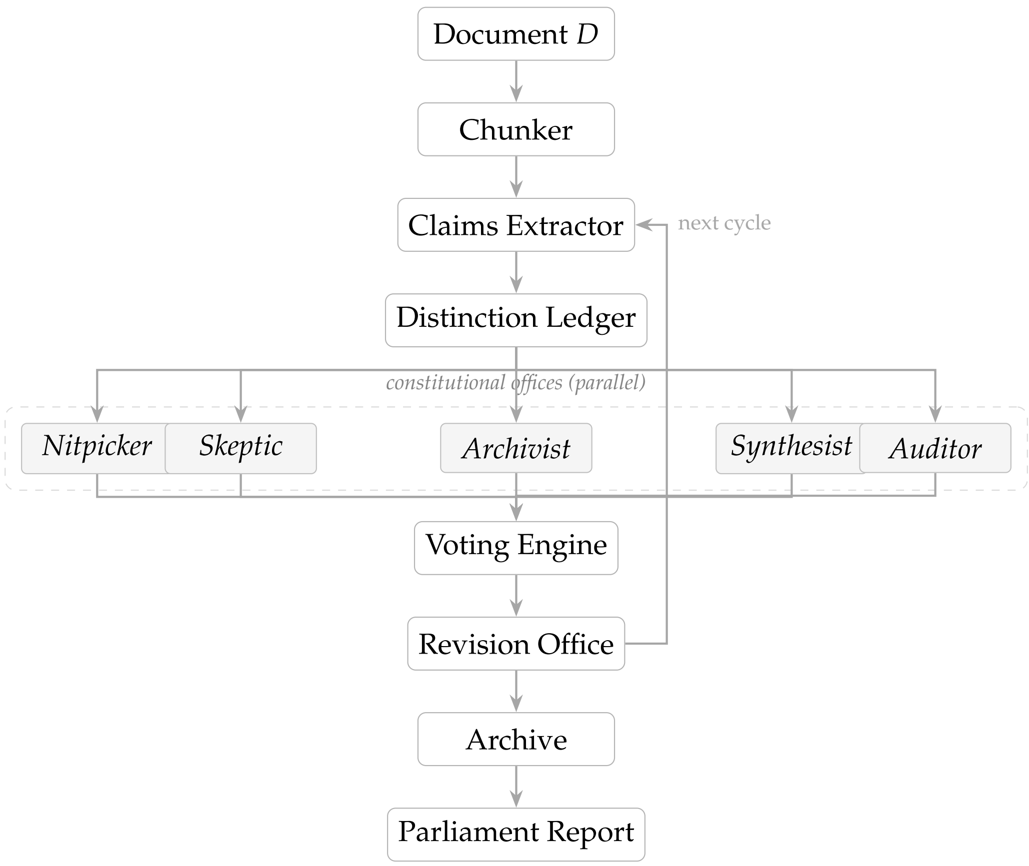

Figure 1: Activity diagram of the Paperbot Parliament pipeline. The five constitutional offices operate in parallel on the same claims file, producing the independent family $\mathcal{P}(C) = \{N(C), Sk(C), Ar(C), Sy(C), Au(C)\}$ rather than a recursive composition. The revision loop is active only when $c > 1$.

5.4. Complexity Analysis

For a document with n chunks, $k = 5$ constitutional offices, and c revision cycles, the number of model invocations per chunk is:

Claims Extractor	1
Ledger extraction	1
Office calls per cycle	$k \cdot c$
Revision passes	$c - 1$

giving a total of $M = n(kc + c + 1)$ invocations across the document. For the default configuration $k = 5$, $c = 1$:

$$M = n(5 \cdot 1 + 1 + 1) = 7n.$$

For the first experimental run with $n = 3$ chunks and $c = 1$:

$$M = 7 \times 3 = 21 \text{ invocations},$$

consistent with the six office outputs observed for `chunk_aa` plus the claims extractor. Because all offices are data-independent given C, the architecture is embarrassingly parallel: all five office calls within a cycle can in principle run concurrently, reducing wall-clock time to $O(n \cdot c)$ invocations on a parallel executor.

5.5. Unix Pipeline Lineage

The implementation uses only standard Unix utilities: `split`, `cat`, `grep`, `awk`, `tail`, and `inotifywait`. The codebase is small: the core parliamentary logic, once boilerplate shell plumbing is removed, is approximately 150 lines of executable logic, with a separate query tool for cross-session memory interrogation. The architecture is recognizably in the Unix tradition:

Unix principle	Parliament realization
One tool, one task	Each office has one constitutional role
Text streams	Claims, reports, and ledger as plain files
Composable filters	Offices as transforms on claims
Persistent logs	`distinction.ledger`, `lyapunov.log`
Daemon-friendly	`inotifywait` queue processing

The parliament is closer to a document-analysis instrument in the Unix tradition than to the typical framework-heavy multi-agent system. The entire executable logic is approximately 150 lines once shell boilerplate is removed.

5.6. Implementation Status

Component	Status	Version
Claims Extractor	Implemented, tested	v1
Nitpicker	Implemented, tested	v1
Skeptic	Implemented, tested	v1
Archivist	Implemented, tested	v1
Synthesist	Implemented, tested	v1
Admissibility Auditor	Implemented, tested	v1
Distinction Ledger	Implemented, tested	v1
Voting Engine	Implemented, untested	v2
Revision Office	Implemented, untested	v2
Lyapunov Monitor	Implemented, untested	v5
Cross-session Archive	Implemented, untested	v3
Contradiction Detector	Implemented, untested	v3
Daemon Mode	Implemented, untested	v4
Distinction Graph G_D	Planned	future
Argument Graph (α)	Planned	future
Reconstruction Fidelity (F)	Planned	future

The experimental run exercised the seven tested components of v1. The v2–v5 components are included in the codebase but have not yet been verified by a controlled run.

5.7. Constitutional Offices

Claims Extractor. Converts raw prose into a structured list of explicit claims, definitions, assumptions, proposed mechanisms, and open questions. Forbidden from summarizing, praising, or speculating. Its output is the sole shared input for all downstream offices.

Nitpicker. Identifies ambiguous terms, undefined concepts, hidden assumptions, category errors, and inconsistent terminology. Evaluates whether claims are *well-formed*, not whether they are true.

Skeptic. Identifies unsupported claims, alternative explanations, strongest objections, empirically thin support, and claims with no causal account. Evaluates whether claims are *justified*, not whether they are well-formed.

Archivist. Identifies historical precedents, parallel theories, prior art, missing citations, and intellectual lineage. Places claims in historical context.

Synthesist. Identifies recurring patterns, common abstractions, hidden structure, bridges to external frameworks, and unification opportunities. The only constructive office.

Admissibility Auditor. Identifies unsupported transitions, logical gaps, inadmissible jumps, hidden dependencies, and circular reasoning. Distinguished from the Skeptic by the nature of its question: the Skeptic asks *why should I believe this?*; the Auditor asks *what bridge is missing?* These are different cognitive operations: the Skeptic's question is epistemic; the Auditor's is inferential. In the formal terms of Definition 4.4, the Auditor estimates $|B|$ and provides a lower bound on $1 - \alpha$.

6. Experimental Setup

6.1. Corpus

The first experimental run used a single theoretical document—the *memelock* corpus—covering admissibility frameworks, reachability geometry, process ontology, Repair Theory, and distinction-centered intelligence. The document is a 13-page unpublished theoretical manuscript.

6.2. Model and Hardware

The run used the `granite3.2:8b` model (IBM Granite 3.2, eight billion parameters, instruction-tuned), served locally via Ollama on a consumer workstation running WSL2 on Windows. No GPU acceleration was required. The model was invoked via `ollama run` with no configuration beyond the constitutional office prompts.

6.3. Configuration

Default configuration: $\ell = 250$ lines per chunk, $c = 1$ revision cycle, no daemon mode, no prior sessions in memory. The source document was split into three chunks (`chunk_aa`, `chunk_ab`, `chunk_ac`). The complete pipeline was confirmed for `chunk_aa`; the remaining chunks were chunked but not fully processed before the run report was captured.

7. Results

7.1. Pipeline Completion

The parliament completed all six stages for `chunk_aa`, matching the theoretical invocation count $M = 7 \times 1 = 7$ for one chunk:

Stage	Elapsed	Output
Claims Extractor	21 s	3.2 KB
Nitpicker	11 s	1.8 KB
Skeptic	13 s	2.5 KB
Archivist	26 s	4.1 KB
Synthesist	11 s	2.0 KB
Admissibility Auditor	19 s	1.5 KB
Total (one chunk)	$\approx$101 s	15.1 KB

7.2. Verbatim Transformation Trace

The following trace walks one sentence from the memelock corpus through the full office pipeline.

Raw source:

"The health of a system is determined by how many futures remain accessible within the system's constraints, not by what it contains."

Claims Extractor:
Claim 7: System health is a function of accessible futures, not of system contents.
Assumption 2: The constraint surface is fixed during health evaluation.

Distinction Ledger:
DISTINCTION: System health $\neq$ System contents
DISTINCTION: Reachable futures $\neq$ Stable states

Nitpicker:
TERM: "health" — PROBLEM: used in both a technical RSVP sense and an intuitive biological sense within the same section.

Skeptic:
CLAIM: Health = accessible futures. OBJECTION: A system with many accessi-

ble futures that are all harmful is "healthy" on this account; needs qualification.

Archivist:
CLAIM: Health as reachability. PRECEDENT: Similar to ecological resilience theory (Holling 1973) and autopoiesis (Maturana & Varela 1980); engagement with resilience literature missing.

Synthesist:
PATTERN: The health/contents distinction recurs across RSVP, CLIO, and Repair Theory as a shared anti-essentialism, unifiable under a reachability-conservation hypothesis.

Admissibility Auditor:
TRANSITION: From "RSVP describes reachability" to "RSVP measures health." GAP: No criterion given for which futures count as "accessible" in a constrained system. The inference is inadmissible without it. Estimated $|B| \geq 1$; $\alpha < 1$.

A conventional summarizer would have produced: *The paper argues that system health is related to accessible futures*—preserving the fact while discarding two distinctions, one terminological problem, one epistemic vulnerability, two historical precedents, one unification opportunity, and one inferential bridge.

7.3. The Distinction Ledger

The most theoretically significant output is the content of the distinction ledger. Without explicit instruction to produce $X \neq Y$ pairs, the parliament populated the ledger with 29 entries from a single chunk (selected below):

DISTINCTION: Reality $\neq$ Fundamental objects
DISTINCTION: System health $\neq$ System contents
DISTINCTION: Reachable futures $\neq$ Stable states
DISTINCTION: Admissible transitions $\neq$ Reachable states
DISTINCTION: Prediction accuracy $\neq$ Preservation of actionable differences
DISTINCTION: Memory $\neq$ Static storage
DISTINCTION: Failure $\neq$ Temporary setback
DISTINCTION: Forgetting $\neq$ Loss of stored information
DISTINCTION: Knowledge $\neq$ Static information

DISTINCTION: Intelligence $\neq$ Information processing
DISTINCTION: Compression $\neq$ Reality
DISTINCTION: Reachability $\neq$ Classification
DISTINCTION: Planning $\neq$ Linear path
DISTINCTION: Budgets $\neq$ Accounting documents

These entries are not facts; they are the edges of an approximation to G_D^*. The parliament has not summarized the document; it has reconstructed its distinction graph.

To give the raw count a denominator: the source chunk was 250 lines drawn from a 13-page (approximately 3,250-word) document. Twenty-nine distinctions from 250 lines gives an extraction rate of $29/250 \approx 0.116$ distinctions per source line, or approximately 2.2 distinctions per page. These numbers are not yet meaningful in isolation—a controlled comparison with a baseline summarizer, measuring P on both outputs, is required before the extraction rate can be interpreted as high or low. They are reported here to give the result a denominator, so that future runs can be compared.

7.4. Analytical Differentiation of the Offices

The Nitpicker output was organized around terminological precision: it flagged *reachability* and *admissibility* as potentially conflated, identified *latent fundamentalism* as undefined, and noted dual uses of *repair*.

The Skeptic output was organized around epistemic vulnerability: it challenged the sufficiency of the RSVP triple, requested empirical evidence for the trajectory-memory claim, and noted that the Repair Theory / error-correction distinction was asserted but not demonstrated.

The Archivist output was organized around intellectual lineage: it identified Bergson's process philosophy, Peirce's abduction, and autopoiesis as precedents, and noted missing engagement with reservoir computing.

The Auditor output was organized around inferential gaps: it identified the transition from *RSVP describes fields* to *RSVP models cognition* as an unsupported leap and flagged multiple conclusions as exceeding stated premises. In terms of Definition 4.4, it estimated $|B| \geq 3$, giving $\alpha \leq 1 - 3/|E|$.

The Synthesist output was organized around unification opportunities: it identified admissibility and repair as operating on the same constraint geometry, and noted that the CLIO claim and the RSVP health definition are duals.

These outputs are not identical. They are genuinely different transforms on the same input, consistent with Proposition 5.1.

7.5. Absence of Recursive Collapse

The parliament shows no evidence of recursive summarization collapse. The Archivist's 4.1 KB output references external authors not present in the source; the Auditor's 1.5 KB output lists inferential gaps not visible in the source text itself. Proposition 5.1 provides the structural explanation: because no office receives another office's output, $B_n \circ \cdots \circ B_1(C)$ never forms.

8. Paperbot Parliament as a CLIO Experiment

The first run admits interpretation as a minimal instantiation of the CLIO hypothesis.

Hypothesis. Preserving distinctions is more useful than compressing propositions.

Operationalization. Run a structured multi-agent pipeline that extracts distinctions rather than summaries from a theoretical document.

Observation. The parliament converged toward distinction-oriented representations. Twenty-nine $X \neq Y$ pairs emerged without any prompt instruction to produce that format.

Interpretation. This is consistent with the CLIO hypothesis: a pipeline organized around preservation of actionable differences will tend to produce differences as its primary output, because differences are what each analytical operation is sensitive to. The Nitpicker detects conflations; the Auditor detects collapsed inferential steps; the Skeptic detects failures of epistemic separation. Each constitutional office is a distinction-recovery operation.

The result also illustrates the representational shift that CLIO predicts:

$$\text{Document} \longrightarrow \text{Summary} \quad \text{(conventional)}$$

$$\text{Document} \longrightarrow \text{Claim space} \longrightarrow \text{Distinction space} \quad \text{(parliament)}$$

Whether this constitutes evidence for the hypothesis or merely a demonstration of its plausibility depends on a formal comparison between P for parliamentary analysis and P for conventional summarization across a controlled corpus. That experiment is outlined in Section 10.

9. Discussion

9.1. Distinctions as First-Class Computational Objects

The most striking result of the first run is that the parliament spontaneously produced a distinction algebra. The constitutional prompts do not ask for $X \neq Y$ pairs. Yet the ledger accumulated 29 entries. The $X \neq Y$ form appears to be the natural output of structured adversarial analysis: when a Nitpicker looks for ambiguities, it is implicitly looking for conflations; when an Auditor looks for inferential gaps, it is looking for places where a boundary has been crossed without a bridge.

9.2. The Auditor as Primary Instrument

The Admissibility Auditor is more informative than the Skeptic for theoretical work because the Skeptic produces objections that a trained reader would raise, while the Auditor produces gap reports that even a trained reader might miss, since the gaps are in the inferential structure rather than the content. A framework that is wrong can be corrected by better evidence. A framework whose inferential structure is incomplete cannot be evaluated until the missing bridges are supplied. The Auditor is the office most directly suited to identifying what must be supplied before evaluation is possible.

9.3. Alternative to Retrieval-Augmented Generation

Standard RAG systems store document fragments as dense vectors and retrieve nearest-neighbor fragments at query time. A distinction-centered alternative would store $(X, Y) \in D$ pairs and retrieve by relevance to a future decision that depends on the separation $X \neq Y$. Such a system would be immune to retrieval failure from semantic proximity to the wrong fragment. The distinction ledger produced by the parliament is a prototype of this alternative representation.

10. Future Work

10.1. Formal Test of the Reconstruction Conjecture

Conjecture 4.1 requires: (1) a corpus of theoretical documents with known conceptual structures; (2) both the parliament and a baseline summarizer run on each document; (3) computation of P, K, $\mathcal{A}_\pi$, and F for both; (4) a statistical test of whether $F(P(T)) > F(S(T))$ holds consistently. This would convert the current qualitative result into a quantitative claim.

10.2. Distinction Graphs

The current ledger stores pairs as flat text. The natural extension is the full distinction graph $G_D = (C, D)$ of Definition 4.1, supporting queries of the form: *what consequences follow from confusing X and Y?* and *what distinctions must be preserved for claim Z to be admissible?*

10.3. Argument Admissibility Scoring

Implementing Definition 4.4 requires the Auditor to emit structured output that can be parsed into the argument graph $A = (V, E)$, yielding a quantitative α score for each document.

10.4. Recursive Parliament

A recursive parliament would run the Synthesist's output back through the Claims Extractor, generating second-order claims about the parliament's own analysis. The fixed-point equation $A_{t+1} = B_n \circ \cdots \circ B_1(A_t)$ is already implemented in the revision cycle structure but currently applies only to claim refinement.

10.5. Distinction-Preservation Scoring for Summaries

Summary quality should be measurable by P in addition to ROUGE. The score can be computed by running the parliament on both the original document and the summary, then measuring $|D_{\text{out}} \cap D_{\text{in}}|/|D_{\text{in}}|$. A summary that collapses Memory $\neq$ Storage is worse than one that preserves it, even if both score equally on semantic overlap.

10.6. BNF Appendix

A formal grammar for the distinction ledger format, claim file format, and verdict record format would support tool interoperability and is planned as an implementation appendix.

11. Conclusion

Paperbot Parliament is an experimental platform for distinction-centered document analysis. Its central claim is that the primary output of structured multi-agent analysis of a theoretical document should be not a compressed summary but a distinction algebra: a structured collection of $X \neq Y$ pairs encoding the conceptual boundaries the document was written to establish.

The paper has introduced a formal theory comprising the distinction space (C, D), the distinction-preservation score P, the admissibility distortion $\mathcal{A}_\pi$, the argument admissibility score α, and the reconstruction fidelity F. The central empirical claim is stated as Conjecture 4.1, clearly separated from the proved Proposition 4.5 (Parliament Stability) and Proposition 5.1 (Parallel Independence).

The first experimental run produced 29 distinction pairs from a single 250-line chunk of a theoretical document. The verbatim trace exhibits in miniature the transformation from raw prose to distinction graph. The complexity analysis confirms that the observed invocation count matches the theoretical prediction $M = 7n$.

The system is minimal: 682 lines of Bash, one local model, no framework, every component a plain file in a plain directory. The theoretical interpretation is large: the parliament is a practical instantiation of the CLIO hypothesis that intelligence consists primarily in the preservation of actionable distinctions rather than the storage of facts.

A. Data Format Grammar

The following BNF grammar specifies the data formats used by Paperbot Parliament. Conforming implementations must produce output parseable by these productions.

```
(* Distinction ledger *)
<ledger>       ::= { <entry> }
<entry>        ::= "DISTINCTION:" <concept> "!=" <concept> "\n"
<concept>      ::= <word> { " " <word> }

(* Claims file *)
<claims-file>  ::= { <claim-block> }
<claim-block>  ::= <claim-type> ":" <text> "\n"
<claim-type>   ::= "Claim" | "Definition" | "Assumption"
                 | "Mechanism" | "Question"

(* Office output *)
<office-report> ::= { <finding> }
<finding>       ::= <field> ":" <text> "\n"
```

```
                    [ <field> ":" <text> "\n" ]
<field>          ::= "TERM" | "PROBLEM" | "CLAIM" | "OBJECTION"
                   | "PRECEDENT" | "PATTERN" | "CONNECTION"
                   | "TRANSITION" | "GAP"

(* Verdict record *)
<verdict-file>  ::= { <verdict-line> } <parliament-verdict>
<verdict-line>  ::= "CLAIM_" <nat> ":" <verdict> "\n"
                    "REASON:" <text> "\n"
<verdict>        ::= "PASS" | "FAIL" | "NEEDS_REVISION"
<parliament-verdict> ::= "PARLIAMENT_VERDICT:" <text> "\n"
```

Listing 1: BNF for Parliament data formats

References

References

[1] Flyxion. Axioms for a Falling Universe: The RSVP Framework. Unpublished manuscript, 2024.

[2] Flyxion. Semantic Infrastructure. Independent Research Monograph, 348 pp., 2025.

[3] Flyxion. The HYDRA Framework: Constraint, Admissibility, and Colimit Structure. Independent Research Monograph, 25 pp., 2025.

[4] Flyxion. Distinguishability Geometry (v3). Unpublished manuscript, 2025.

[5] Flyxion. Frozen Processes: Admissibility Structures and Stable Residues. Independent Research Monograph, 2025.

[6] Flyxion. Waves of Collapse: Compression as Reachability Geometry. Independent Research Monograph, 42 pp., 2025.

[7] Flyxion. Spherepop: An Irreversible Computation Calculus. Independent Research Monograph, 2024.

[8] C. S. Holling. Resilience and stability of ecological systems. *Annual Review of Ecology and Systematics*, 4:1–23, 1973.

[9] H. R. Maturana and F. J. Varela. *Autopoiesis and Cognition: The Realization of the Living*. D. Reidel, 1980.

[10] T. S. Kuhn. *The Structure of Scientific Revolutions*. University of Chicago Press, 1962.

[11] L. Wittgenstein. *Tractatus Logico-Philosophicus*. Kegan Paul, 1922.

[12] Ollama. *Ollama: Run Large Language Models Locally*. `https://ollama.com`, 2024.

[13] IBM Research. *Granite 3.2: Open Foundation Models*. IBM, 2025.

Algorithm 1: Paperbot Parliament

Input: Document D; number of revision cycles $c \geq 1$; model M
Output: Parliament report $\mathcal{R}$; distinction ledger L

$Chunks \leftarrow \mathrm{Split}(D, \ell)$; // $\ell = 250$ lines by default
;
for each
  $chunk \in Chunks$
$C \leftarrow \mathrm{ExtractClaims}(chunk, M)$; // structured claims, definitions, assumptions
;
$L \leftarrow L \cup D(C)$; // update cumulative distinction ledger
;
for $t = 1$ **to** c **do**
  $N \leftarrow \mathrm{Nitpicker}(C, M)$; // ambiguities, undefined terms, category errors
  ;
  $Sk \leftarrow \mathrm{Skeptic}(C, M)$; // unsupported claims, strongest objections
  ;
  $Ar \leftarrow \mathrm{Archivist}(C, M)$; // precedents, lineage, missing citations
  ;
  $Sy \leftarrow \mathrm{Synthesist}(C, M)$; // patterns, bridges, unification
  ;
  $Au \leftarrow \mathrm{Auditor}(C, M)$; // inferential gaps, inadmissible jumps
  ;
  if $c > 1$ ***and*** $t < c$ **then**
    $V \leftarrow \mathrm{Vote}(N, Sk, Ar, Au)$; // per-claim PASS / FAIL / NEEDS_REVISION
    ;
    $C \leftarrow \mathrm{Revise}(C, N, Sk, Au)$; // remove violations, flag needs-evidence
    ;
$\mathrm{Archive}(C,\ N,\ Sk,\ Ar,\ Sy,\ Au)$;
$\mathcal{R} \leftarrow \mathcal{R} \cup \mathrm{AssembleReport}(chunk)$;
return $(\mathcal{R},\ L)$;