CONSTRAINT CLOSURE AND ITS FAILURES

Projection, Entropy, and the Degeneration of Form in AI Systems and Written Discourse

Flyxion

Independent Researcher

April 4, 2026

ABSTRACT

This essay develops a unified formal framework for analyzing two apparently distinct phenomena: the accidental release of proprietary AI system code and the degeneration of argumentative essay form under generative pressure. Both are shown to be instances of a general class of failures arising when complex systems operate under insufficient constraint. The formal machinery introduced encompasses projection-based inference over high-dimensional configuration spaces, prior-dominant attractor dynamics, entropy-driven diffusion following constraint boundary violations, and the condition of constraint closure as the determinant of whether a system produces a meaningful output or degenerates into an arbitrary sequence of locally admissible steps. The analysis demonstrates that such failures are not isolated accidents but structurally predictable consequences of partial constraint in rapidly iterating, high-dimensional systems. The essay concludes by deriving parallel design implications for software pipelines and argumentative practice, and by situating both within a general principle governing the reliability of observable artifacts as evidence of underlying generative structure.

CONTENTS

1. INTRODUCTION

This essay examines two related phenomena through a unified formal framework. The first is a recent incident in which proprietary AI system code was accidentally released into the public domain, an event widely interpreted as a failure of operational discipline but here analyzed as a structurally predictable outcome of high-dimensional system dynamics. The second is the degeneration of essay form under generative pressure, a phenomenon in which the surface properties of argumentative writing are reproduced without the underlying constraint structure that gives those properties meaning.

The central claim is that both phenomena are instances of a general class of failures arising when complex systems operate under insufficient constraint. The formal machinery developed to analyze the code leak—encompassing projection-based inference, prior-dominant attractor dynamics, entropy-driven diffusion, and the condition of constraint closure—applies without modification to the epistemological analysis of written discourse. This unification is not merely analogical. It reflects a deeper structural identity between the dynamics of software pipelines and the dynamics of argumentative trajectories.

In both domains, the observable surface exceeds its underlying necessity. A running system and a grammatically fluent essay are both low-dimensional projections of the processes that produced them. The reliability of either as evidence of the internal structure it represents depends on whether the generating process was sufficiently constrained to permit near-identification of that structure from the observable output. When constraint is partial, surfaces decouple from interiors. The result, in the case of software, is a pipeline that admits trajectory escape into unintended configurations; in the case of writing, a sequence of sentences that resembles argument without instantiating it.

The essay proceeds as follows. Sections 2 through 7 develop the formal analysis of the code leak, introducing the projection framework, the near-identifiability result, attractor dynamics, entropy diffusion, constraint failure in dynamical workflows, and the economics of computational enclosure. Section 8 establishes the bridging argument, formalizing essay writing as a constrained dynamical process and making explicit the structural identity that motivates the unified treatment. Sections 9 through 14 apply the formal machinery to the epistemological domain, analyzing the degeneration of essay form, the structural basis of clarity, the collapse of length as a reliability signal, and the shift in trust from artifact to process. Section 15 addresses the discourse on safety and the error of reasoning across incompatible abstraction levels. Section 16 derives design implications. Section 17 concludes.

2. THE INCIDENT AS A PROJECTION EVENT

Let $\mathcal{X}$ denote the full internal configuration space of an AI development system, understood as the product of all relevant state dimensions: source code, build artifacts, workflow specifications, deployment scripts, infrastructure definitions, organizational protocols, and the latent practices that govern their composition. A point $X \in \mathcal{X}$ represents a complete instantaneous specification of the system.

No external observer has access to X directly. What is observable is a projection $\Pi(X) \in \mathcal{O}$, where $\mathcal{O}$ is a lower-dimensional space and $\Pi : \mathcal{X} \to \mathcal{O}$ is the projection map induced by the particular leak event. In the incident under analysis, $\Pi(X)$ consists of the released code artifacts together with whatever structural inferences can be drawn from their organization, naming conventions, and dependency structure.

Definition 2.1 (Observation-consistent set)**.** Given an observation $o \in \mathcal{O}$, the observation-consistent set is

$$\mathcal{X}_o = \{X \in \mathcal{X} : \Pi(X) = o\}.$$

This is the fiber of Π over o, and represents all internal configurations that would have produced the observed projection.

The central interpretive question is whether $\mathcal{X}_o$ is sufficiently constrained to permit reconstruction of the underlying system. In general, Π fails to be injective: multiple configurations produce the same projection, and $\mathcal{X}_o$ may be high-dimensional. What the incident reveals, however, is not the impossibility of reconstruction but rather the regime of near-identifiability, in which the fiber is structured enough to admit efficient convergence to configurations that are functionally indistinguishable from X.

Proposition 2.2 (Near-identifiability under architectural priors)**.** *Let $\mathcal{P}$ denote the prior distribution over $\mathcal{X}$ induced by shared architectural conventions, published design patterns, and existing open-source systems. If the support of $\mathcal{P}$ restricted to $\mathcal{X}_o$ concentrates around a small number of equivalence classes under functional equivalence, then practical reconstructibility holds even when strict injectivity of Π fails.*

The rapid emergence of working replicas following the incident constitutes empirical evidence for this proposition. Reconstruction was not performed by exhaustive search over $\mathcal{X}_0$ but by gradient descent in the prior landscape, converging to the nearest attractor basin consistent with the observed projection.

3. CONSTRAINT CLOSURE, DEGENERACY, AND PRIOR COLLAPSE IN PROJECTION SYSTEMS

3.1. *Configuration, Projection, and Constraint Structure*

Let $\mathcal{X}$ be a configuration space, equipped with a topology sufficient to define continuity and minimization. Let $\mathcal{O}$ be an observation space, and let

$$\Pi : \mathcal{X} \to \mathcal{O}$$

be a (generally non-injective) projection operator.

Let $\mathcal{A} \subset \mathcal{X}$ denote the admissible set defined by structural constraints. These constraints may arise from physical laws, logical relations, architectural rules, or procedural requirements.

Definition 3.1 (Observation-Consistent Set)**.** Given an observation $o \in \mathcal{O}$, define the observation-consistent set

$$\mathcal{F}(o) := \{X \in \mathcal{A} : \Pi(X) = o\}.$$

When Π is non-injective, $\mathcal{F}(o)$ is generically non-singleton. This non-uniqueness is referred to as *projection degeneracy*.

3.2. *Variational Reconstruction Principle*

We define reconstruction as the solution to a variational problem balancing observational fidelity, constraint adherence, and prior preference.

Let $\mathcal{R} : \mathcal{X} \to \mathbb{R}_{\geq 0}$ be a constraint functional, vanishing exactly on $\mathcal{A}$, and let $\Phi : \mathcal{X} \to \mathbb{R}$ encode a prior over configurations.

Definition 3.2 (Reconstruction Functional)**.** For fixed weights $\lambda, \mu \geq 0$, define

$$\mathcal{E}(X; o) := \underbrace{\|\Pi(X) - o\|^2}_{\text{data fidelity}} + \lambda \underbrace{\mathcal{R}(X)}_{\text{constraint penalty}} + \mu \underbrace{\Phi(X)}_{\text{prior}}.$$

A reconstruction is any minimizer

$$X^* \in \arg\min_{X \in \mathcal{X}} \mathcal{E}(X; o).$$

3.3. Constraint Closure and Uniqueness

Definition 3.3 (Constraint Closure)**.** The system achieves constraint closure at observation o if $\mathcal{F}(o)$ is a singleton.

Theorem 3.4 (Uniqueness under Injectivity and Strict Constraint)**.** *Suppose:*

1. Π *is injective on* $\mathcal{A}$*,*
2. $\mathcal{R}$ *is strictly convex in a neighborhood of* $\mathcal{A}$*,*
3. $\lambda > 0$.

Then for any $o \in \Pi(\mathcal{A})$*, the reconstruction problem admits a unique minimizer* X^**.*

Proof. Injectivity implies $\mathcal{F}(o)$ contains at most one element. Strict convexity of $\mathcal{R}$ ensures that deviations from $\mathcal{A}$ are penalized uniquely. The data term enforces $\Pi(X) = o$, restricting minimizers to $\mathcal{F}(o)$. Hence uniqueness follows. □

This regime corresponds to systems in which every admissible trajectory is structurally determined. Outputs exhibit necessity: perturbations destroy validity.

3.4. Projection Degeneracy and Prior Selection

When Π is non-injective on $\mathcal{A}$, the feasible set $\mathcal{F}(o)$ contains multiple elements. In this case, the prior term Φ determines selection among them.

Definition 3.5 (Prior-Dominant Regime)**.** The system is prior-dominant if, for all $X \in \mathcal{F}(o)$,

$$\mathcal{E}(X; o) \approx \mu\Phi(X),$$

i.e., the data and constraint terms fail to distinguish elements of $\mathcal{F}(o)$.

Proposition 3.6 (Prior Collapse). *If Π is highly degenerate and $\mathcal{R}$ is flat on $\mathcal{F}(o)$, then*

$$X^* \approx \arg\min_{X \in \mathcal{F}(o)} \Phi(X).$$

Proof. On $\mathcal{F}(o)$, the data term vanishes and the constraint term is constant. Minimization reduces to the prior term. □

Thus, reconstruction collapses to the nearest prior-dominant attractor. The observed output does not determine the underlying configuration; instead, it selects among typical configurations encoded in Φ.

3.5. Dynamical Formulation and Trajectory Escape

Let $X(t)$ be a trajectory in $\mathcal{X}$ evolving under a dynamical process. Constraint closure requires

$$X(t) \in \mathcal{A} \quad \forall t \in [0, T].$$

Definition 3.7 (Trajectory Escape). A trajectory exhibits escape if there exists t_0 such that

$$X(t_0) \notin \mathcal{A}.$$

Suppose constraint enforcement occurs via k sequential verification steps, each with success probability $p_i(v)$ depending on a velocity parameter v.

Proposition 3.8 (Velocity-Induced Failure). *If $p_i(v) < 1$ for any i, then the probability of constraint closure over the trajectory is*

$$P_{success} = \prod_{i=1}^{k} p_i(v),$$

which decreases monotonically in v. In the limit of repeated execution, failure occurs with probability 1.

Proof. Immediate from independence of steps and $p_i(v) < 1$. Repeated trials yield eventual failure almost surely. □

Thus, trajectory escape is not anomalous but structurally inevitable under bounded enforcement capacity.

3.6. Informational Emptiness

Let O denote observable outputs and H denote hidden structural variables (e.g., constraints, arguments, internal states).

Definition 3.9 (Mutual Information)**.** The mutual information between O and H is

$$I(O; H) = H(O) - H(O \mid H).$$

Definition 3.10 (Informationally Empty Regime)**.** A system is informationally empty if

$$I(O; H) \approx 0$$

while $H(O)$ is large.

In this regime, outputs exhibit high surface complexity but encode negligible information about underlying structure. Such outputs are replaceable without loss of validity.

3.7. Synthesis

Two distinct but interacting failure modes emerge:

1. **Projection Degeneracy:** Non-injective Π produces large $\mathcal{F}(o)$.
2. **Constraint Failure:** Weak or violated $\mathcal{R}$ permits trajectories to exit $\mathcal{A}$.

When both occur simultaneously, reconstruction is governed by prior minimization rather than structural necessity. The system produces outputs that are locally coherent but globally arbitrary.

This regime is characterized by:

$$\text{degenerate projection} + \text{weak constraint} \quad \Rightarrow \quad \text{prior-dominant collapse.}$$

3.8. Interpretation

Systems exhibiting constraint closure produce outputs whose structure is necessary and fragile under perturbation. Systems in the prior-dominant regime produce outputs that are statistically plausible but structurally unconstrained.

This distinction is independent of domain. It applies equally to physical systems, computational pipelines, and symbolic constructions. In all cases, the determining factor is whether the admissible set collapses to a unique element under the combined action of projection and constraint.

□

4. PROJECTION DEGENERACY AND NEAR-IDENTIFIABILITY

The fiber $\mathcal{X}_o$ is in general a manifold of dimension $\dim \mathcal{X} - \dim \mathcal{O}$. When $\mathcal{X}$ is high-dimensional and Π collapses many dimensions, this fiber may itself be high-dimensional. The degeneracy of the projection is thus measured by the codimension of $\mathcal{O}$ in $\mathcal{X}$.

In the present case, the projection exposes a significant but still partial slice of X. The released artifacts include module boundaries, interface signatures, and dependency relationships sufficient to constrain the space of admissible implementations. However, many internal decisions remain hidden: the specific optimization choices, the training infrastructure, the evaluation pipeline, and the coordination protocols between subsystems. The fiber $\mathcal{X}_o$ is therefore high-dimensional, but not uniformly so.

Definition 4.1 (Effective degeneracy)**.** Let $\mathcal{X}_o^{\mathcal{P}}$ denote the restriction of the prior $\mathcal{P}$ to $\mathcal{X}_o$. The effective degeneracy of the projection is the entropy

$$H(\mathcal{X}_o^{\mathcal{P}}) = -\int_{\mathcal{X}_o} \mathcal{P}(X) \log \mathcal{P}(X)\, dX.$$

Low effective degeneracy means the prior concentrates the fiber onto a small number of typical configurations, enabling reconstruction even under high nominal degeneracy.

The distinction between nominal degeneracy—the dimension of the fiber—and effective degeneracy—the entropy of the prior-weighted fiber—is central to the analysis. A projection may be nominally degenerate while admitting near-identification because the prior structure of the domain strongly constrains which configurations are typical.

This result has a general implication: the security value of enclosure does not scale with the nominal dimension of the hidden configuration space, but with the effective degeneracy of the projection under the shared prior of the reconstruction domain. In a mature field with strong architectural conventions, this effective degeneracy may be surprisingly low even when the nominal configuration space is vast.

5. Prior-Dominant Attractors and Rapid Reconstruction

The reconstruction of a system from its projection is most naturally understood as a dynamical process on $\mathcal{X}$. External agents, operating within a shared domain characterized by learned architectural priors, initialize their reconstruction at some point X_0 and evolve according to an energy functional that balances fidelity to the observed projection against consistency with prior expectations.

Definition 5.1 (Reconstruction energy functional)**.** Let $\lambda > 0$ be a regularization parameter. The reconstruction energy functional is

$$\mathcal{E}(X) = \|\Pi(X) - o\|^2 + \lambda \cdot (-\log \mathcal{P}(X)),$$

where the first term penalizes deviation from the observed projection and the second term penalizes departure from the prior. Reconstruction trajectories evolve by gradient descent on $\mathcal{E}$.

The attractor structure of $\mathcal{E}$ is determined primarily by the prior term when λ is large relative to the constraint strength of the projection. In this prior-dominant regime, the landscape of $\mathcal{E}$ is essentially the landscape of $-\log \mathcal{P}$, and trajectories converge to the modes of the prior that are consistent with the projection.

Definition 5.2 (Prior-dominant attractor)**.** A configuration $X^* \in \mathcal{X}_o$ is a prior-dominant attractor if it is a local minimum of $\mathcal{E}$ in the prior-dominant regime, i.e., if X^* is a mode of $\mathcal{P}$ restricted to $\mathcal{X}_o$.

The rapid emergence of working replicas following the incident is explained by this structure. The reconstruction problem is not one of deriving the hidden system from first principles but of identifying the prior-dominant attractor in the neighborhood of the observed projection. When the prior is sufficiently concentrated—as it is in a mature field with strong shared conventions—this identification is computationally efficient.

Theorem 5.3 (Convergence under prior dominance). *Suppose $\mathcal{P}$ is log-concave and the projection Π is linear. Then the reconstruction energy $\mathcal{E}$ is convex on $\mathcal{X}_o$ in the prior-dominant regime, and gradient descent converges to the unique global minimum at a rate determined by the condition number of the Hessian of $\mathcal{E}$ restricted to $\mathcal{X}_o$.*

Proof. Under log-concavity, $-\log \mathcal{P}$ is convex. The quadratic fidelity term is convex by construction. The restriction of a convex function to a linear fiber is convex. Convergence of gradient descent on a convex function follows from standard results, with rate determined by the ratio of the largest to smallest eigenvalues of the Hessian. □

The practical import of this result is that near-identifiability and rapid reconstruction are not properties of the leaked artifact alone, but of the interaction between the artifact and the prior landscape of the reconstruction domain. Enclosure is therefore structurally fragile in any domain where shared priors are sufficiently expressive.

6. ENTROPY DIFFUSION AND STRUCTURAL RELAXATION

Once the constraint boundary of the enclosure has been breached and the projection $\Pi(X)$ has entered the observable domain, the subsequent dissemination of the leaked material follows the dynamics of entropy-driven diffusion on a networked information environment.

Let $\rho(n,t)$ denote the density of the leaked information at node n at time t in the network. In the continuous approximation, the evolution of ρ is governed by the diffusion equation

$$\frac{\partial \rho}{\partial t} = D\nabla^2 \rho - \mu \rho,$$

where D is the diffusion coefficient determined by network connectivity and sharing propensity, and μ is a decay term accounting for information becoming stale or superseded. In the absence of active counter-dissemination, the system evolves toward a diffuse equilibrium in which the leaked material has propagated throughout the reachable component of the network.

This process is irreversible in the thermodynamic sense. The total entropy of the information distribution,

$$S(t) = -\int \rho(n,t) \log \rho(n,t)\, dn,$$

is monotonically non-decreasing under the diffusion dynamics. Once the constraint boundary is breached, the system cannot return to the low-entropy state in which the information was concentrated within the enclosure. Retraction of the leaked material is not a reversal of the diffusion process but an attempt to impose a new boundary condition on an already relaxed distribution, an intervention that cannot undo the structural relaxation that has occurred.

Proposition 6.1 (Irreversibility of entropic relaxation)**.** *Let ρ_0 be the initial distribution concentrated within the enclosure, and let ρ_t be the distribution at time $t > 0$ under diffusion dynamics. Then $S(\rho_t) > S(\rho_0)$ for all $t > 0$, and there is no admissible dynamics on the network that returns the system to ρ_0 without external constraint reimposition.*

The general principle that emerges is one of entropic smoothing: localized constraint failures result in global redistribution of informational content until a new equilibrium is reached. The significance of the incident lies not in the quantum of information released but in the irreversibility of the relaxation it initiates.

7. CONSTRAINT FAILURE IN DYNAMICAL WORKFLOWS

The conditions under which the incident occurred can be formalized by treating the deployment pipeline as a constrained dynamical system. Let the state of the pipeline at time t be a point s_t in a state space S. The admissible states form a submanifold $S_{\text{adm}} \subset S$ defined by the constraints imposed by security protocols, access controls, and workflow specifications. The pipeline is designed to evolve along trajectories $\gamma : [0, T] \to S$ that remain within S_{adm}.

Definition 7.1 (Constraint closure)**.** A pipeline achieves constraint closure if for every admissible initial state $s_0 \in S_{\text{adm}}$ and every admissible input sequence, the resulting trajectory γ satisfies $\gamma(t) \in S_{\text{adm}}$ for all $t \in [0, T]$.

The presence of manual steps in the pipeline introduces unresolved degrees of freedom. A manual step is one whose output is not determined by the preceding state and a fixed computational rule, but depends on the judgment, attention, and situational awareness of a human operator. Such steps prevent full constraint closure because the operator's action cannot be guaranteed to preserve admissibility under all conditions.

Proposition 7.2 (Trajectory escape under partial closure)**.** *Let a pipeline contain k manual steps. The probability that the trajectory remains within S_{adm} over the full execution* *is at most* $\prod_{i=1}^{k} p_i$*, where* p_i *is the probability that the operator at step* i *takes an admissible action. If any* $p_i < 1$*, the pipeline does not achieve constraint closure, and trajectory escape occurs with positive probability.*

The incident is therefore a trajectory-level failure rather than a static error. The system had not achieved structural completion: the configuration was not the result of a constraint violation in a closed system but of an admissibility condition that had never been fully enforced. The distinction is important because it changes the appropriate remediation strategy. Static errors call for correction; trajectory-level failures call for redesign of the constraint structure.

8. THE ENCLOSURE OF COMPUTATION AND ITS BREAKDOWN

The economic significance of the incident is best understood through the concept of computational enclosure. Proprietary AI systems derive their commercial value not from the intrinsic scarcity of the computational processes they implement—since these processes can, in principle, be replicated by any agent with sufficient prior knowledge and computational resources—but from the enclosure of access to those processes within a controlled interface.

Computational enclosure functions analogously to other forms of enclosure in economic history: it converts a process that would otherwise be freely reproducible into a commodity whose access can be regulated and monetized. The value of the enclosure is a function of the cost of reconstruction from outside, which in turn depends on the effective degeneracy of the projection available to external agents.

The near-identifiability result from Section 3 implies that this enclosure is structurally fragile in any domain where the shared prior of potential reconstructors is sufficiently expressive. As the prior landscape of the AI development domain becomes richer—through the accumulation of published research, open-source systems, and trained practitioners—the marginal value of the constraint boundary decreases. The enclosure depends for its effectiveness on the reconstruction cost remaining high, but reconstruction cost is a decreasing function of prior expressiveness.

The leak temporarily collapses this enclosure, enabling widespread replication and recomposition of the leaked system's functional properties. The economic consequence is not primarily the loss of the specific information contained in the projection but the demonstration that the enclosure is weaker than its market valuation implied. This informational effect on beliefs about future reconstruction costs may be more significant than the direct informational content of the leak itself.

9. DEGENERACY, RECOVERABILITY, AND THE COLLAPSE OF ENCLOSURE

The preceding analysis introduces three concepts that, while developed in different sections, describe a single underlying phenomenon: effective degeneracy of projection, practical recoverability of the generating system, and the stability of computational enclosure. This section establishes their formal relationship.

Let $o = \Pi(X)$ be an observed projection of a configuration $X \in \mathcal{X}$, and let $\mathcal{X}_o^{\mathcal{P}}$ denote the prior-weighted fiber over o. Recall that the effective degeneracy of the projection is given by

$$H(\mathcal{X}_o^{\mathcal{P}}).$$

Definition 9.1 (Reconstruction cost)**.** Let $\mathcal{A}$ be a class of reconstruction algorithms operating under prior $\mathcal{P}$. The reconstruction cost $C(o)$ is the expected computational effort required for $\mathcal{A}$ to produce a configuration $\hat{X}$ such that

$$\Pi(\hat{X}) = o \quad \text{and} \quad \hat{X} \sim X \text{ under functional equivalence.}$$

Proposition 9.2 (Degeneracy–cost relation)**.** *Under a fixed prior $\mathcal{P}$ and reconstruction class $\mathcal{A}$, the reconstruction cost $C(o)$ is a monotone increasing function of the effective degeneracy $H(\mathcal{X}_o^{\mathcal{P}})$.*

Proof. Low effective degeneracy implies that $\mathcal{X}_o^{\mathcal{P}}$ is concentrated on a small number of high-probability configurations. Gradient-based or sampling-based procedures converge rapidly to these modes. As degeneracy increases, the mass of $\mathcal{X}_o^{\mathcal{P}}$ spreads over a larger set, increasing the search complexity required to locate a functionally equivalent configuration. □

This establishes that effective degeneracy is the fundamental quantity governing recoverability: systems with low degeneracy are practically reconstructible even when the projection is not injective.

Definition 9.3 (Enclosure strength)**.** The strength of a computational enclosure is defined as

$$E = C(o),$$

the reconstruction cost of the system given its observable projection.

This definition captures the economic function of enclosure: it derives its value from the difficulty of external reconstruction rather than from intrinsic properties of the system.

Theorem 9.4 (Degeneracy–enclosure equivalence)**.** *Under fixed prior $\mathcal{P}$, the following are equivalent up to monotone transformation:*

1. *Low effective degeneracy $H(\mathcal{X}_o^{\mathcal{P}})$*
2. *Low reconstruction cost $C(o)$*
3. *Weak enclosure strength E*

Proof. The equivalence of (1) and (2) follows from the degeneracy–cost relation. The equivalence of (2) and (3) is immediate from the definition of enclosure strength. Monotonicity preserves ordering across all three quantities. □

This result formalizes the collapse of enclosure observed in the incident. The leak did not merely expose information; it revealed that the effective degeneracy of the projection under the domain prior was already low. Reconstruction was therefore inevitable, and the enclosure was structurally weak prior to the leak.

9.1. Prior Dependence and Temporal Instability

The equivalence above holds relative to a fixed prior $\mathcal{P}$. In practice, $\mathcal{P}$ evolves over time as the domain accumulates knowledge.

Proposition 9.5 (Prior-induced collapse)**.** *Let $\mathcal{P}_t$ be the prior at time t, with increasing expressiveness over time. Then for a fixed projection o,*

$$H(\mathcal{X}_o^{\mathcal{P}_t}) \downarrow \quad \Rightarrow \quad C_t(o) \downarrow \quad \Rightarrow \quad E_t \downarrow .$$

Thus, even in the absence of new leaks, enclosure strength decays over time as the shared prior of the field becomes richer.

9.2. Interpretation

The collapse of computational enclosure is therefore not an event but a process. The leak acts as a discontinuity that reveals a pre-existing condition: that the system resides in a regime of low effective degeneracy under contemporary priors.

The same structure applies to argumentative texts. A text with low constraint density has high effective degeneracy, making its generating process unrecoverable and its content informationally weak. A text with high constraint density has low degeneracy, making its generating process identifiable and its structure resistant to arbitrary reconstruction.

In both cases, the governing principle is identical: the reliability and value of a projection are determined not by its surface complexity, but by the extent to which it reduces the space of admissible generating processes.

10. SPEED, AUTOMATION, AND THE LIMITS OF CONTROL

The incident can be further situated within the general dynamics of high-velocity development environments. Such environments are characterized by the combination of partial automation with human-in-the-loop processes, a combination that creates a regime in which constraint enforcement is probabilistic rather than absolute.

Let v denote the iteration velocity of the development pipeline, measured by the number of state transitions per unit time. As v increases, the time available for human operators to verify admissibility at each manual step decreases proportionally. If the cognitive cost of verification is approximately constant per step, and the time budget per step is $1/v$, then the probability of successful verification at each step is a decreasing function of v. Combining this with the result of Section 6:

Corollary 10.1 (Constraint enforcement probability under velocity pressure)**.** *In a pipeline with k manual steps operating at velocity v, the probability that the trajectory remains within S_{adm} decreases monotonically with v. For sufficiently high v, trajectory escape becomes effectively certain over a long enough operational period.*

This result formalizes the intuition that such failures are emergent properties of the system architecture rather than isolated lapses of individual operators. The operator at the relevant step did not fail to apply a rule; the system was operating in a regime in which reliable rule application was structurally impossible given the time constraints imposed by the development velocity. The appropriate unit of analysis is therefore not the individual action but the dynamical regime.

The tension between iteration speed and reliability is not resolvable within the existing architecture: increasing speed while maintaining manual steps produces a system in which constraint enforcement probability goes to zero. Resolution requires either reducing velocity, eliminating manual steps through automation, or restructuring the constraint system so that admissibility is enforced by structural invariants rather than by operator verification.

11. WRITING AS A CONSTRAINED DYNAMICAL PROCESS

The two domains examined in this essay—the structural analysis of AI system failures and the epistemological critique of essay form under generative pressure—are not merely analogous. They are instances of the same underlying phenomenon: the behavior of high-dimensional systems under partial constraint. Establishing this identity requires showing that the formal machinery developed in the preceding sections applies without modification to the dynamics of argumentative writing.

An essay and a software pipeline share a common formal structure. Both are trajectories through a space of possible states, where the admissibility of each transition depends on conditions imposed by prior commitments. In a software pipeline, these conditions are encoded in interfaces, contracts, and workflow dependencies. In an argumentative text, they are encoded in premises, definitions, and logical entailments. In both cases, constraint closure—the condition in which all degrees of freedom are resolved by the governing structure—determines whether the system produces a meaningful output or degenerates into an arbitrary sequence of locally plausible steps.

The failure modes are structurally parallel. A pipeline with unresolved manual steps admits transitions that violate admissibility, producing trajectories that escape the intended configuration space. An essay produced without governing premises admits continuations that are locally coherent but globally arbitrary, producing text that resembles argument without instantiating it. In both cases, the observable surface, whether running code or grammatical prose, provides insufficient evidence of the underlying constraint structure.

The projection framework introduced in Section 2 applies directly to textual artifacts. A reader observing a finished essay sees only a low-dimensional projection of the argumentative process that produced it. The reconstructibility of that process—and therefore the trustworthiness of the artifact—depends on whether the projection is sufficiently constrained to permit near-identification of the underlying trajectory. Transparent reasoning, explicit premises, and traceable inference steps are precisely the structural features that reduce projection degeneracy in written discourse, concentrating the observation-consistent set around a small number of admissible argumentative trajectories.

Similarly, the attractor dynamics of Section 4 apply to the formation of arguments under generative pressure. Shared conceptual priors, disciplinary conventions, and rhetorical patterns constitute an attractor landscape within which generated text settles. Without external constraint, generation converges to the nearest basin regardless of whether that basin corresponds to a true argument. The result is the prior-dominant essay: structurally familiar, disciplinarily conventional, intellectually inert.

Definition 11.1 (Argumentative constraint closure)**.** An essay achieves argumentative constraint closure if for every premise in the governing structure, all transitions between claims are determined by the logical and evidential relations specified by that structure, and no claim appears whose inclusion or position is underdetermined by the constraints.

Without constraint closure, the essay is formally indistinguishable from an unconstrained generation: locally plausible, globally arbitrary, and incapable of serving as reliable evidence of the reasoning process that produced it.

12. THE NECESSITY OF PROMPTED STRUCTURE

The analysis of writing as a constrained dynamical process has an immediate operational implication: the constraint structure of an essay cannot be inferred post hoc from an unconstrained generative request. It must be specified prior to or during the generation process. A request of the form "write an essay on topic T" defines only a projection target in $\mathcal{O}$ and leaves the corresponding fiber in $\mathcal{X}$ maximally degenerate.

Definition 12.1 (Unconstrained essay prompt)**.** A prompt is unconstrained if it specifies a target topic or domain without imposing conditions on the admissible argumentative trajectories, such as required premises, explicit tensions, or exclusion criteria.

Under an unconstrained prompt, the generating process operates entirely within the prior landscape described in Section 4. The resulting text is therefore a realization of a prior-dominant attractor rather than a trajectory determined by a specific constraint structure. The observable output may satisfy the formal properties of an essay, but the underlying trajectory is not identifiable from the projection, since the same output class is consistent with a large set of generating processes.

By contrast, a constrained prompt specifies a submanifold of admissible trajectories in $\mathcal{X}$. This reduces the effective degeneracy of the projection and forces the generator to produce a trajectory that is determined, at least partially, by the imposed constraints.

Proposition 12.2 (Constraint necessity for argumentative generation)**.** *Let $\mathcal{G}$ be a generative system operating over $\mathcal{X}$. Then $\mathcal{G}$ produces an argumentatively identifiable trajectory if and only if the prompt defines a constraint set whose projection reduces the effective degeneracy of the observation-consistent set.*

The implication is that the responsibility for producing an essay does not lie solely with the generator but with the specification of the constraint structure. Without such specification, the generator cannot produce necessity, because necessity is not a property of the output distribution but of the constraint space within which the output is selected.

13. THE DEGENERATION OF ESSAY FORM UNDER UNCONSTRAINED GENERATION

The contemporary availability of generative systems has introduced a fundamental asymmetry between form and substance in written discourse. It is now trivial to produce text that satisfies the superficial criteria of an essay: paragraph organization, grammatical continuity, disciplinary vocabulary, and transitions that simulate logical progression. These properties no longer reliably indicate the presence of an underlying argument, because they can be produced without one.

An essay, properly understood, is not defined by its length or its surface organization but by the existence of a constrained argumentative trajectory. Each claim is positioned relative to others in such a way that the structure is not arbitrary: the removal or reordering of components alters the meaning of the whole. This internal dependency is what distinguishes argument from aggregation.

Unconstrained generation severs this dependency. When text is produced without a governing structure of premises, definitions, or tensions, it becomes interchangeable in the sense relevant to the projection framework: the observation-consistent set over the space of generating processes is maximally degenerate, since the surface output is consistent with the entire range of unconstrained generative processes. The result is an essay-shaped object that lacks necessity. Its sentences follow one another, but not because they must.

This is experienced by readers as a form of cognitive emptiness, despite the appearance of completeness. The experience is not aesthetic but epistemic: the reader correctly perceives that the text fails to reduce uncertainty about the argumentative trajectory that produced it. No information about the generator is available from the output, because the output is consistent with all generators. The text is maximally uninformative in the information-theoretic sense while appearing maximally structured in the surface sense.

14. CONSTRAINT AS THE SOURCE OF CLARITY

The widespread claim that contemporary audiences prefer brevity over long-form writing is frequently misinterpreted as a rejection of essays as a form. A more precise characterization is that readers are rejecting unconstrained verbosity. What is selected for in the current environment is not shortness but clarity, and clarity is a structural rather than aesthetic property. It emerges when the degrees of freedom in an argument have been sufficiently reduced by the governing constraint structure.

Definition 14.1 (Structural clarity)**.** The structural clarity of an argumentative text is inversely proportional to the effective degeneracy of its projection over the space of generating processes. A text is structurally clear to the degree that its observable properties constrain the space of argumentative trajectories consistent with it.

Under this definition, a short text can be structurally opaque if its brevity is achieved by compression that eliminates the inferential structure that would permit reconstruction, and a long text can be structurally clear if each sentence functions as an admissibility constraint on the subsequent argument. The relevant quantity is not length but constraint density: the ratio of constraint-imposing content to total content.

Constraint reduces the space of admissible statements locally and globally. When an argument is governed by explicit premises and a defined problem, each sentence must contribute to resolving the tension established by that structure. Irrelevant or redundant expressions are excluded not by editorial judgment but by the logic of the constraint space: they are inadmissible transitions. The resulting text appears clearer because it is necessary, and it is trustworthy because its observable properties reduce the degeneracy of its projection over generating processes.

15. CONSTRAINT DENSITY AS A DESIGN VARIABLE

The preceding sections treat constraint density as an emergent property of a completed argumentative artifact. For practical purposes, however, constraint density must be understood as a design variable: a quantity that can be actively controlled during the generation process.

Let Y be a text generated under a constraint specification C. Then the resulting constraint density $D(Y)$ is a function of both the richness of C and the fidelity with which the generating process adheres to it. Formally,

$$D(Y) = D(C, \mathcal{G}),$$

where $\mathcal{G}$ denotes the generator.

A weak constraint specification—one that defines a topic but not a trajectory—induces a high-entropy trajectory space $\mathcal{T}(Y)$, resulting in low constraint density. A strong specification—one that defines premises, tensions, and admissibility conditions—reduces $\mathcal{T}(Y)$ and increases $D(Y)$.

This reframes writing as a problem of constraint design rather than content production. The relevant question is not what sentences should appear, but what degrees of freedom must be eliminated such that only a narrow class of trajectories remains admissible.

Proposition 15.1 (Constraint density monotonicity). *Let $C_1 \subset C_2$ be two constraint sets, where C_2 imposes strictly more restrictions on admissible trajectories than C_1. Then for any generator $\mathcal{G}$,*

$$D(Y_{C_2}) \geq D(Y_{C_1}),$$

where Y_{C_i} is the text generated under constraint set C_i.

This result implies that increasing clarity is not a matter of refining expression but of eliminating admissible alternatives. Clarity is therefore achieved not by adding content, but by reducing possibility.

In practical terms, the design of an essay becomes the design of a constraint surface within the trajectory space. The text that results is a trace of that surface, and its clarity is determined by how sharply the surface restricts movement.

16. ESSAY AS TRAJECTORY RATHER THAN ARTIFACT

The foregoing analysis supports a reconceptualization of the essay as a trajectory through a constrained space rather than as a static artifact. The value of an essay lies in the path it traces: how it moves from initial assumptions to conclusions, how it navigates and excludes alternative possibilities, and how each transition is determined by the constraint structure rather than by the generator's local preferences.

Proposition 16.1 (Necessity as the criterion of argumentative value)**.** *An argumentative text has positive necessity if and only if there exists at least one admissible alternative at each transition that is excluded by the governing constraint structure. A text of zero necessity is one in which all transitions are equally admissible, and the specific trajectory traced is determined only by the prior distribution of the generator.*

This proposition makes precise the intuition that a real argument has directionality: if a step is removed, something breaks; if the order is reversed, coherence degrades. This fragility is a signal of rigor rather than a defect, because it indicates that the trajectory is constrained. A text that could be rearranged without consequence is one in which the constraint structure is empty.

The implication for practice is that writing cannot be separated from the specification of constraints. To request an essay without providing an argument is to request a trajectory without defining a constraint space. The resulting text may exhibit local coherence—each sentence is locally plausible—but lacks global direction. The formal parallel with pipeline constraint failure is exact: in both cases, the absence of closure admits arbitrary continuation, and the resulting trajectory may escape the intended configuration space entirely without any single step being individually inadmissible.

17. Perceived Emptiness and the Failure of Informational Gain

The reader's experience of unconstrained generative text as "empty" can be formalized in information-theoretic terms. Let Y denote the observable text and G the generating process. The informational value of Y with respect to G is given by the mutual information $I(Y; G)$.

In the case of a constrained argumentative trajectory, Y reduces uncertainty about G, because the constraint structure leaves a detectable imprint on the observable output. Different generating processes would produce observably different trajectories under the same constraints. Thus $I(Y; G)$ is positive.

In the case of unconstrained generation, however, the observable output is consistent with a large class of generating processes. The same essay-shaped text could have been produced by many different trajectories within the prior landscape. The mutual information $I(Y; G)$ is therefore low, approaching zero in the limit of maximal degeneracy.

Definition 17.1 (Informational emptiness)**.** An argumentative text is informationally empty if $I(Y; G)$ is below a threshold required for reliable inference of the generating process.

This formalization explains why readers disengage from such text. The act of reading is an attempt to reconstruct the generating trajectory from the observable projection. When the projection is maximally degenerate, this reconstruction fails, and the reader receives no informational gain about the underlying reasoning process.

The preference for shorter, distilled content is therefore not a preference for reduced content but for increased information density. A shorter text that imposes strong constraints may yield higher $I(Y; G)$ than a longer unconstrained text. The relevant quantity is not length but the reduction in uncertainty about the generating trajectory achieved by observing the text.

18. The Collapse of Length as a Signal

Historically, the length and structural organization of a text functioned as low-dimensional proxies for a high-dimensional property: the intellectual effort, sustained engagement, and constraint-navigating work required to produce a genuine argument. The reliability of these proxies depended on the production costs of long structured text being sufficiently high to make simulation expensive. Under these conditions, form and substance were correlated in expectation, and readers could rationally treat surface features as evidence of underlying argumentative quality.

Generative systems have broken this correlation by enabling high-dimensional surface properties to be produced from a near-zero-constraint process. This is precisely the situation analyzed in Section 3 for the code leak: the projection has been decoupled from the configuration it was formerly reliable evidence of. Length and structural organization now belong to the class of low-cost observables that can be produced without instantiating the high-dimensional property they previously tracked.

The reader's response to this decoupling is structurally rational. When a proxy variable loses its correlation with the quantity it proxied, agents who relied on the proxy update their inference strategy. Readers are increasingly attending to properties that remain difficult to simulate under unconstrained generation: internal consistency over long dependencies, explicit acknowledgment of counter-arguments, traceable inferential structure, and the visibility of constraint. These properties are more resistant to unconstrained generation because they impose structural requirements on the generating process itself, not merely on the observable output.

The collapse of length as a signal is therefore not a cultural shift in attention spans but a rational Bayesian update in response to a change in the generating environment. The prior over generators, given an observation of long structured text, has shifted: whereas previously it was concentrated on processes involving sustained argumentative constraint, it is now distributed across a much larger class of processes, most of which involve no constraint at all.

19. ADVERSARIAL GENERATION AND THE SIMULATION OF CONSTRAINT

A potential objection to the preceding analysis is that generative systems are not limited to unconstrained outputs. They can simulate many of the surface properties associated with constraint: explicit premises, structured arguments, and even counter-argumentation. This raises the question of whether constraint density can itself be simulated without being instantiated.

Let Y be a text that exhibits surface markers of constraint. The question is whether these markers correspond to actual reductions in the admissible trajectory space or merely to patterns that resemble such reductions.

Definition 19.1 (Simulated constraint)**.** A text exhibits simulated constraint if it contains surface features associated with constraint (e.g., explicit premises or logical transitions) without those features reducing the effective degeneracy of the observation-consistent set.

Simulated constraint arises when the generator produces patterns that are correlated with constraint under the training distribution but does not enforce the underlying relations that give those patterns necessity. In such cases, the apparent structure does not restrict the trajectory space, because alternative continuations remain equally admissible.

Proposition 19.2 (Indistinguishability under shallow observation)**.** *If observation is restricted to local or short-range dependencies, then simulated constraint is observationally indistinguishable from genuine constraint.*

Proof. Local patterns associated with constraint can be reproduced independently of the global structure. If observation does not capture long-range dependencies or global consistency, then both genuine and simulated constraint produce identical local statistics. □

The distinction becomes observable only under deeper inspection: long-range dependency tracking, counterfactual perturbation (removing or altering a premise), or attempts to reconstruct the generating trajectory. Genuine constraint produces fragility under such perturbations, while simulated constraint remains robust because no underlying dependency exists.

This introduces an adversarial dimension to the evaluation of argumentative text. As generators improve, the detection of constraint requires increasingly global tests, shifting the burden from surface evaluation to structural analysis.

20. TRANSPARENCY, BIAS, AND TRUST IN HIGH-VOLUME TEXT ENVIRONMENTS

In an environment saturated with generated text, the trust function of a reader cannot be adequately defined over the artifact alone. The reader must attempt to infer the constraint structure of the generating process from the available evidence, and the reliability of this inference depends on the degree to which the artifact makes that structure legible.

Definition 20.1 (Process legibility)**.** An argumentative text has high process legibility if its observable properties substantially reduce the effective degeneracy of its projection over the space of generating processes. Equivalently, high legibility means that the observation-consistent set is concentrated around a small number of generating trajectories with similar constraint structures.

Transparency in this context is the act of increasing process legibility by making the constraint structure of the argument explicit. An argument that exposes its premises, acknowledges its exclusions, enumerates the alternative positions it considered, and traces its inferential steps reduces projection degeneracy by providing observational content that is inconsistent with most unconstrained generating processes. The underlying trajectory becomes more nearly identifiable.

Transparency does not eliminate bias in the sense of eliminating the generator's particular perspective or prior commitments. It makes bias legible by making the constraint structure explicit. A reader who can identify the constraints under which an argument was produced can evaluate its conclusions relative to those constraints, rather than treating the argument as if it were constraint-free. This is the appropriate epistemic stance in a high-volume generative environment: not the demand for unbiased text, which is formally incoherent, but the demand for text whose biases are identifiable from its observable properties.

The preference, widely observed in contemporary information practice, for distilled summaries from sources with known perspectives over comprehensive neutral accounts from anonymous generators, is therefore not a retreat from epistemic standards but an adaptation to a changed environment. When the volume of available text exceeds the capacity for detailed process reconstruction, readers rely on reputational signals as proxies for constraint structure. The source's known perspective functions as a prior over the constraint structures they typically employ, reducing the effective degeneracy of the projection without requiring the reader to verify the constraint structure from the text alone.

21. Reputation as a Prior Over Constraint Structures

In high-volume text environments, readers cannot reconstruct the generating process for each observed artifact. Instead, they rely on priors over classes of generators. A source's reputation functions as such a prior, encoding expectations about the constraint structures that the source typically employs.

Definition 21.1 (Reputational prior)**.** Let $\mathcal{G}$ be the space of generating processes. A reputational prior is a probability distribution π over $\mathcal{G}$ conditioned on the identity of the source, reflecting the expected constraint structures of that source's outputs.

When a reader encounters a text from a known source, the effective degeneracy of the projection is reduced by conditioning on π. The observation-consistent set becomes

$$\mathcal{X}_Y^{\pi} = \{G \in \mathcal{G} : G \text{ could have produced } Y \text{ and is consistent with } \pi\}.$$

If π is sufficiently concentrated, then even a relatively compressed or partial artifact can yield high process legibility. Conversely, text from an unknown or unconstrained generator corresponds to a diffuse prior, leaving the projection highly degenerate and reducing trust.

This explains the empirical preference for distilled summaries from sources with known biases over longer texts from anonymous or opaque generators. Bias, when made explicit, functions as a constraint on the generating process, thereby reducing degeneracy. The reader can condition on this constraint and evaluate the argument within its proper frame.

The absence of declared bias does not eliminate bias but increases degeneracy by removing a constraint that could otherwise be used to infer the generating trajectory. In this sense, transparency is not a moral property but an informational one: it reduces the uncertainty of the inverse problem faced by the reader.

22. Misplaced Abstraction and the Discourse on Safety

Public interpretations of both the code leak and the degeneration of essay form exhibit a characteristic error of reasoning: the conflation of observations at one abstraction level with conclusions that are only warranted at another. In the case of the leak, local operational failures are treated as evidence for global claims about the safety properties of AI systems. In the case of essay form, the local observation that generative systems can produce essay-shaped text is treated as evidence for global claims about the nature of reasoning and the death of argumentative culture.

Both errors have the same formal structure. A local event E is observed. E is consistent with a global hypothesis H at a higher abstraction level. The inference from E to H is made as if E were specific evidence for H, when in fact E is consistent with many incompatible global hypotheses and provides only weak discriminating power among them.

Definition 22.1 (Abstraction level error)**.** An abstraction level error occurs when a claim C at abstraction level ℓ_1 is inferred from evidence E at abstraction level $\ell_2 < \ell_1$ without accounting for the multiplicity of level-ℓ_1 states consistent with E.

The correct analytical response to the code leak is not a global claim about AI safety but a local claim about pipeline constraint failure and its structural predictability. The incident is evidence about the dynamics of high-velocity development environments under partial constraint, not about the alignment properties of the systems being developed. Similarly, the correct analytical response to the availability of essay-shaped generative text is not a global claim about the death of argument but a structural claim about the decoupling of surface from substance and the consequent shift in the reliability of length-based signals.

The discourse on safety, in particular, is prone to this error because safety is an inherently global property—it concerns the behavior of a system across all possible inputs and contexts—while the available evidence is always local, consisting of specific incidents, evaluations, and observed behaviors. The gap between local evidence and global safety claims is not closed by accumulating more local evidence; it requires formal arguments about the relationship between local and global properties, arguments of the kind developed in the formal sections of this essay.

23. CONSTRAINT CLOSURE AS NECESSARY BUT NOT SUFFICIENT

The analysis thus far establishes constraint closure as a necessary condition for the reliability of an output as evidence of its generating process. It does not, however, establish sufficiency.

A system may achieve constraint closure within an incorrectly specified constraint space. In such cases, the resulting trajectory is fully determined by the governing structure, but that structure does not correspond to the intended domain.

Definition 23.1 (Mis-specified constraint space)**.** A constraint space is mis-specified if it excludes trajectories that correspond to valid solutions of the underlying problem or includes trajectories that do not.

In argumentative terms, this corresponds to reasoning that is internally consistent but based on false premises or incomplete framing. In system terms, it corresponds to a pipeline that enforces all constraints correctly but implements the wrong specification.

Proposition 23.2 (Closure without correctness)**.** *Constraint closure guarantees identifiability of the generating process but does not guarantee correctness of the process with respect to an external objective.*

This distinction is essential for interpreting both technical failures and epistemological artifacts. The absence of constraint closure implies unreliability, but its presence implies only that the process is recoverable, not that it is correct.

The design problem is therefore twofold. First, to achieve constraint closure such that the system's outputs are identifiable. Second, to ensure that the constraint space itself is appropriately specified relative to the domain of interest.

Failure at the first level produces degeneracy. Failure at the second produces coherent but incorrect systems. The two are distinct and require different forms of analysis and remediation.

24. OPERATIONALIZATION AND EVALUATION OF CONSTRAINT STRUCTURE

The preceding analysis establishes constraint density, identifiability, and closure as formal properties of generative processes. To be practically useful, however, these quantities must admit operational approximations: procedures by which a reader or evaluator can estimate whether an observed text lies in the high-density, identifiable regime or in the degenerate regime.

Because the generating process is not directly observable, all such procedures must operate on the projection Y. The problem is therefore an inverse one: to infer properties of the constraint structure from the observable artifact.

24.1. Perturbation-Based Evaluation

Let Y be an argumentative text. Consider a perturbation operator Δ that modifies a component of Y, such as removing a premise, altering a definition, or reordering a segment. Let $\Delta(Y)$ denote the perturbed text.

Definition 24.1 (Perturbation sensitivity)**.** The perturbation sensitivity of Y with respect to Δ is

$$S_\Delta(Y) = d\big(\mathcal{T}(Y), \mathcal{T}(\Delta(Y))\big),$$

where $d(\cdot, \cdot)$ is a distance between trajectory sets.

Intuitively, $S_\Delta(Y)$ measures how much the space of admissible trajectories changes under a local modification of the text.

In high constraint density regimes, small perturbations induce large changes in the admissible trajectory space, because each component of the text participates in a network of dependencies. In degenerate regimes, perturbations have minimal effect, because the trajectory is not tightly constrained.

Proposition 24.2 (Sensitivity criterion)**.** *If $D(Y)$ is high, then $S_\Delta(Y)$ is large for a broad class of perturbations. If $D(Y)$ is low, then $S_\Delta(Y) \approx 0$ for most perturbations.*

This provides a practical test: texts that remain coherent under arbitrary local modification are likely to be low-density, while texts that degrade under targeted perturbation exhibit genuine constraint structure.

24.2. Counterfactual Trajectory Analysis

A second operational approach is to examine counterfactual continuations. Let Y be a partial text, and let $\mathcal{C}(Y)$ denote the set of admissible continuations.

Definition 24.3 (Continuation entropy)**.** The continuation entropy of Y is

$$H_{\text{cont}}(Y) = H(\mathcal{C}(Y)),$$

the entropy of the distribution over admissible continuations.

Low continuation entropy indicates that the trajectory is strongly constrained: only a small number of continuations are admissible. High continuation entropy indicates degeneracy: many continuations are equally plausible.

Proposition 24.4 (Continuation constraint relation). *Constraint density is inversely related to continuation entropy. In particular,*

$$D(Y) \uparrow \quad \Rightarrow \quad H_{cont}(Y) \downarrow .$$

Empirically, this can be approximated by attempting to generate or enumerate continuations and measuring their divergence. A text for which many divergent continuations remain plausible is structurally unconstrained.

24.3. Dependency Depth and Long-Range Consistency

Constraint structure manifests most clearly in long-range dependencies. Let $Y = (y_1, \dots, y_n)$ be a sequence of statements. Define a dependency relation $y_i \to y_j$ if the admissibility of y_j depends on y_i.

Definition 24.5 (dependency depth). The dependency depth of Y is the length of the longest chain

$$y_{i_1} \to y_{i_2} \to \cdots \to y_{i_k}.$$

High dependency depth indicates that early constraints propagate through the entire trajectory, enforcing global structure. Low dependency depth indicates that statements are locally but not globally constrained.

Proposition 24.6 (Depth as a proxy for constraint density). *Texts with high dependency depth exhibit higher constraint density than texts with shallow or fragmented dependency structures.*

This provides another operational signal: genuine arguments maintain consistency over long spans, while degenerate texts exhibit only local coherence.

24.4. Reconstruction Stability

A final operational measure is reconstruction stability. Let $\hat{G}(Y)$ denote an inferred generating process from the observed text.

Definition 24.7 (Reconstruction stability). A text Y has high reconstruction stability if small perturbations in Y produce small changes in $\hat{G}(Y)$.

In high-density regimes, the constraint structure tightly determines the generating process, so reconstruction is stable. In degenerate regimes, small changes in the text may correspond to entirely different generating processes, reflecting the diffuse posterior over $\mathcal{G}$.

24.5. Interpretation

These operational criteria do not provide exact measurements of constraint density or identifiability. They provide observable proxies that distinguish between regimes.

A text that is sensitive to perturbation, exhibits low continuation entropy, maintains long-range dependencies, and yields stable reconstructions is likely to have high constraint density and to function as reliable evidence of its generating process.

A text that is robust under arbitrary modification, admits many divergent continuations, exhibits only local coherence, and yields unstable reconstructions is likely to be degenerate.

The task of evaluation in high-volume generative environments is therefore not to assess surface quality, but to estimate which regime a given artifact occupies. Constraint structure, though not directly observable, leaves detectable signatures in the behavior of the text under these operations.

25. IRREVERSIBILITY OF CONSTRAINT FAILURE AND THE NO-FREE-RECONSTRUCTION PRINCIPLE

25.1. Statement of the Principle

We now unify the preceding results into a single structural theorem governing projection systems, dynamical constraints, and reconstruction processes.

Theorem 25.1 (Irreversibility of Constraint Failure)**.** *Let $\mathcal{X}$ be a configuration space, $\Pi : \mathcal{X} \to \mathcal{O}$ a non-injective projection, and $\mathcal{A} \subset \mathcal{X}$ an admissible set defined by constraints.*

Suppose:

1. *(**Constraint Failure**) There exists a trajectory $X(t)$ such that $X(t_0) \notin \mathcal{A}$ for some t_0,*

2. *(**Diffusion**) The resulting information distribution $\rho(x,t)$ evolves under a diffusion process,*

3. *(**Degeneracy**) The projection Π is non-injective on $\mathcal{A}$.*

Then:

1. *The entropy $S[\rho]$ increases monotonically,*

2. *The effective feasible set $\mathcal{F}(o)$ expands,*

3. *Reconstruction becomes prior-dominated,*

4. *The original configuration X cannot be uniquely recovered by any local or bounded process.*

Proof. Constraint failure introduces information outside $\mathcal{A}$. Diffusion spreads this information over Ω, increasing entropy. Non-injectivity of Π implies that multiple configurations remain consistent with observations. As $\mathcal{F}(o)$ expands and constraints weaken, minimization of the reconstruction functional is dominated by the prior term. Irreversibility follows from the entropy increase and the impossibility of reconstructing a unique pre-image under non-injective projection. □

25.2. No-Free-Reconstruction Corollary

Corollary 25.2 (No-Free-Reconstruction)**.** *Let $o = \Pi(X)$ be an observation obtained after constraint failure and diffusion. Then any reconstruction $\hat{X}$ satisfying $\Pi(\hat{X}) = o$ must incur one of the following:*

1. *Non-uniqueness: $\hat{X}$ is not uniquely determined,*

2. *Prior dependence: $\hat{X}$ depends on an external prior Φ,*

3. *Additional information: reconstruction requires data not contained in o.*

In particular, no procedure exists that reconstructs X uniquely from o alone.

Proof. Follows directly from projection degeneracy and expansion of $\mathcal{F}(o)$. Any selection requires either external bias (prior) or additional constraints. □

25.3. Necessity versus Replaceability

Definition 25.3 (Necessary Output)**.** An output O is necessary if it corresponds to a unique admissible configuration:

$$|\mathcal{F}(O)| = 1.$$

Definition 25.4 (Replaceable Output)**.** An output O is replaceable if

$$|\mathcal{F}(O)| \gg 1.$$

Proposition 25.5 (Structural Criterion)**.** *Constraint-closed systems produce necessary outputs. Systems subject to constraint failure produce replaceable outputs.*

Proof. Constraint closure implies injectivity of Π on $\mathcal{A}$. Failure introduces degeneracy, increasing $|\mathcal{F}(O)|$. □

25.4. Global Interpretation

The preceding results establish a universal structural law:

$$\text{constraint failure} \;\Rightarrow\; \text{diffusion} \;\Rightarrow\; \text{degeneracy} \;\Rightarrow\; \text{prior-dominant reconstruction.}$$

Here is the revised version in continuous prose:

This law applies independently of domain and governs physical systems under entropy increase, computational systems under pipeline escape, informational systems under data leakage, and symbolic systems under unconstrained generation.

In all cases, the loss of constraint closure induces an irreversible transition from necessity to replaceability.

25.5. Final Consequence

The fundamental invariant is not complexity, size, or surface structure, but constraint.

Once constraint is lost, no increase in observational detail or post hoc intervention can restore uniqueness. Reconstruction becomes a selection problem over priors rather than a recovery of underlying reality.

□

26. CATEGORICAL FORMULATION: OBSTRUCTION TO UNIQUE LIFTING UNDER NON-MONOMORPHIC PROJECTION

26.1. *The Projection as a Functor*

Let $\mathcal{C}$ denote a category of configurations (e.g., field states, programs, semantic structures), and let $\mathcal{D}$ denote a category of observables.

Let

$$\Pi : \mathcal{C} \to \mathcal{D}$$

be a functor representing projection from configurations to observations.

Definition 26.1 (Monomorphism)**.** A morphism $f : A \to B$ in $\mathcal{C}$ is a monomorphism if for all objects X and morphisms $g_1, g_2 : X \to A$,

$$f \circ g_1 = f \circ g_2 \;\Rightarrow\; g_1 = g_2.$$

We say Π is *monomorphic on a subcategory* $\mathcal{A} \subset \mathcal{C}$ if it reflects distinct morphisms in $\mathcal{A}$.

Failure of monomorphism corresponds to projection degeneracy.

26.2. *Lifting Problem*

Given an observable object $o \in \mathcal{D}$, the reconstruction problem is equivalent to finding a lift:

$$X[d, "\Pi"] * [r, "o"][ru, dashed, "\exists?"]o$$

That is, we seek an object $X \in \mathcal{C}$ such that $\Pi(X) = o$.

Definition 26.2 (Unique Lifting)**.** The lifting problem admits a unique solution if any two lifts X_1, X_2 satisfying $\Pi(X_1) = \Pi(X_2) = o$ are isomorphic in $\mathcal{C}$.

26.3. Obstruction via Non-Monomorphism

Theorem 26.3 (Non-Uniqueness of Lifts)**.** *If Π fails to be monomorphic on $\mathcal{A} \subset \mathcal{C}$, then there exist objects $X_1, X_2 \in \mathcal{A}$ such that*

$$\Pi(X_1) \cong \Pi(X_2) \quad \textit{but} \quad X_1 \ncong X_2.$$

Proof. Failure of monomorphism implies existence of distinct morphisms or objects that are identified under Π. Hence multiple non-isomorphic lifts exist. □

Thus, projection degeneracy corresponds categorically to the failure of Π to be faithful or monomorphic.

26.4. Constraint Closure as a Subcategory

Let $\mathcal{A} \subset \mathcal{C}$ be the full subcategory of admissible configurations satisfying constraints.

Constraint closure corresponds to the condition that $\Pi|_{\mathcal{A}}$ is monomorphic.

Proposition 26.4 (Constraint Closure and Monomorphism)**.** *If $\Pi|_{\mathcal{A}}$ is monomorphic, then lifts of o within $\mathcal{A}$ are unique up to isomorphism.*

26.5. Prior as a Selection Functor

In the presence of non-unique lifts, selection requires an additional structure.

Let

$$\Phi : \mathcal{C} \to \mathbf{Set}$$

be a functor assigning to each object a set of preference weights (or an ordered structure).

Definition 26.5 (Prior Selection)**.** A prior induces a choice of lift via a selection rule

$$X^* \in \arg \min_{X \in \Pi^{-1}(o)} \Phi(X).$$

Thus, priors act as external selection mechanisms resolving non-uniqueness.

26.6. Sheaf-Theoretic Interpretation

Let $\{U_i\}$ be a cover of a domain, and let $\mathcal{S}$ be a presheaf assigning local configurations:

$$\mathcal{S}(U_i) = \text{local admissible states.}$$

Observations correspond to local sections $s_i \in \mathcal{S}(U_i)$.

Consistency requires agreement on overlaps:

$$\rho_{ij}(s_i) = \rho_{ji}(s_j).$$

Definition 26.6 (Global Section)**.** A global section $s \in \mathcal{S}(\Omega)$ is a configuration whose restrictions match all local observations.

Theorem 26.7 (Obstruction via Čech Cohomology)**.** *A unique global reconstruction exists if and only if the first Čech cohomology group vanishes:*

$$H^1(\{U_i\}, \mathcal{S}) = 0.$$

Non-vanishing cohomology corresponds to incompatible or underdetermined local data, yielding multiple inequivalent global sections.

26.7. Dynamical Interpretation as Functorial Failure

Let $X(t)$ define a trajectory in $\mathcal{C}$. Constraint closure requires that $X(t)$ remains in $\mathcal{A}$.

Trajectory escape corresponds to leaving the subcategory:

$$X(t_0) \notin \mathcal{A}.$$

After escape, Π is no longer monomorphic on the reachable region, and lifting becomes non-unique.

26.8. Final Synthesis

The categorical structure of the system is:

$$\text{non-monomorphic } \Pi \;\Rightarrow\; \text{non-unique lifts} \;\Rightarrow\; \text{selection by prior}$$

and, dynamically,

$$\text{loss of subcategory } \mathcal{A} \;\Rightarrow\; \text{loss of monomorphism} \;\Rightarrow\; \text{obstruction to reconstruction.}$$

26.9. Conceptual Consequence

Reconstruction is not fundamentally an inverse problem but a lifting problem in a category where the projection functor is not monomorphic.

Constraint closure corresponds to a regime in which Π behaves like a faithful embedding on $\mathcal{A}$. Once this property is lost, uniqueness cannot be restored by any internal operation.

Thus, the irreversibility of constraint failure is equivalent to the impossibility of defining a canonical inverse functor to Π on its image.

□

27. RECURSIVE KNOWLEDGE SYSTEMS AND CONSTRAINT ACCUMULATION

27.1. Stateful Knowledge Dynamics

We extend the static reconstruction framework to a dynamical setting in which the system maintains and updates an internal knowledge state.

Let $\mathcal{X}$ denote a configuration space of structured knowledge representations (e.g., documents, graphs, indexed corpora). A recursive knowledge system is defined by an iterative process

$$X_{t+1} = \mathcal{C}(X_t, q_t),$$

where:

- $X_t \in \mathcal{X}$ is the knowledge state at iteration t,
- q_t is a query or task,
- $\mathcal{C}$ is an update operator that produces new knowledge from X_t and q_t.

The output O_t of the system is a projection

$$O_t = \Pi(X_t, q_t),$$

which may be reintegrated into the knowledge state.

Thus, the system evolves as

$$X_{t+1} = X_t \cup \Delta(O_t),$$

where $\Delta(O_t)$ denotes newly generated or refined structure.

27.2. Constraint Accumulation

Each update introduces additional structural constraints on admissible configurations.

Let $\mathcal{A}_t \subset \mathcal{X}$ denote the admissible set at time t. Then

$$\mathcal{A}_{t+1} = \mathcal{A}_t \cap \mathcal{C}(O_t),$$

where $\mathcal{C}(O_t)$ represents constraints induced by newly incorporated knowledge.

Definition 27.1 (Constraint Accumulation)**.** A recursive system exhibits constraint accumulation if

$$\mathcal{A}_{t+1} \subseteq \mathcal{A}_t \quad \forall t,$$

with strict inclusion for infinitely many t.

Thus, the admissible set shrinks over time, reducing ambiguity.

27.3. Degeneracy Reduction

Let $\mathcal{F}_t(o)$ denote the feasible set consistent with observation o at time t:

$$\mathcal{F}_t(o) := \{X \in \mathcal{A}_t : \Pi(X) = o\}.$$

Theorem 27.2 (Degeneracy Reduction under Constraint Accumulation)**.** *If the system exhibits constraint accumulation and each update introduces non-redundant constraints, then*

$$\dim(\mathcal{F}_{t+1}(o)) \leq \dim(\mathcal{F}_t(o)),$$

with strict inequality for infinitely many t.

Proof. Each new constraint reduces the admissible set. Non-redundancy ensures that at least one degree of freedom is eliminated, reducing the dimension of the feasible set. □

Thus, recursive updates progressively reduce projection degeneracy.

27.4. Transition from Prior-Dominant to Constraint-Dominant Regime

Recall that prior-dominant reconstruction occurs when constraints fail to distinguish elements of $\mathcal{F}(o)$.

Proposition 27.3 (Regime Transition)**.** *Under constraint accumulation, a system transitions from prior-dominant to constraint-dominant if*

$$|\mathcal{F}_t(o)| \to 1 \quad \text{as } t \to \infty.$$

Proof. Follows from monotonic reduction of $\mathcal{F}_t(o)$ and eventual elimination of degeneracy. □

In the limit, reconstruction becomes unique and independent of prior.

27.5. Fixed Points and Consistency

The update operator $\mathcal{C}$ defines a dynamical system on $\mathcal{X}$.

Definition 27.4 (Knowledge Fixed Point)**.** A state X^* is a fixed point if

$$\mathcal{C}(X^*, q) = X^* \quad \text{for all admissible queries } q.$$

Proposition 27.5 (Self-Consistent Knowledge State)**.** *If constraint accumulation converges, then the system approaches a fixed point X^* satisfying global consistency.*

This corresponds to a state in which all admissible queries yield outputs already encoded in X^*.

27.6. Relation to Identifiability

Let $\widetilde{\Pi}$ denote the induced map on equivalence classes. Degeneracy corresponds to non-injectivity of $\widetilde{\Pi}$.

Theorem 27.6 (Recovery of Identifiability)**.** *If constraint accumulation reduces $\mathcal{F}_t(o)$ to a singleton, then $\widetilde{\Pi}$ becomes injective on the limit admissible set.*

Proof. A singleton feasible set implies uniqueness of preimage, hence injectivity. □

Thus, identifiability can be achieved dynamically without modifying the projection operator.

27.7. Structural Interpretation

Recursive knowledge systems replace brute-force context accumulation with structured constraint accumulation.

Instead of approximating the configuration space via large context windows, the system constructs an indexed, persistent representation of constraints. Queries operate by navigating this structure rather than expanding the observation.

27.8. 8. Economic and Computational Consequences

Constraint accumulation yields reduced dependence on large context windows by externalizing and organizing information into persistent structures. It enables improved scalability through this persistence, as the system no longer requires repeated recomputation over the same informational substrate. It also increases interpretability by maintaining explicit representations of knowledge, allowing the structure of reasoning to be inspected and modified. Over time, it leads to decreasing marginal cost of inference, since each additional query benefits from an increasingly constrained and well-organized knowledge state.

Thus, recursive knowledge systems provide a more efficient alternative to stateless inference architectures.

27.9. Synthesis

Recursive knowledge systems implement the inverse of degeneracy-inducing processes:

$$\text{iteration} \;\Rightarrow\; \text{constraint accumulation} \;\Rightarrow\; \text{degeneracy reduction} \;\Rightarrow\; \text{identifiability.}$$

This complements the earlier result:

$$\text{constraint failure} \;\Rightarrow\; \text{degeneracy} \;\Rightarrow\; \text{prior collapse.}$$

Together, these describe two opposing dynamical regimes governing reconstruction systems.

□

28. ASSEMBLY INDEX AND CONSTRAINT-BASED COMPLEXITY

A related attempt to quantify structural complexity is provided by assembly theory, which assigns to an object an *assembly index* defined as the minimal number of steps required to construct the object from primitive components.

Formally, for an object O composed of K subunits, the assembly index a_O satisfies

$$\log_2(K) \leq a_O \leq K - 1,$$

with lower values corresponding to structures that can be constructed through reusable subcomponents and higher values corresponding to structures requiring sequential construction.

Assembly theory interprets this quantity as a measure of the amount of selection required to produce an object, and proposes that objects with high assembly index are unlikely to arise in the absence of structured generative processes.

28.1. Relation to Constraint Density

The assembly index can be interpreted within the present framework as a lower bound on the constraint structure required to generate an object. A high assembly index implies that the generative process must traverse a constrained sequence of states, reducing the admissible trajectory space.

In this sense, assembly index is correlated with constraint density: objects that require many non-redundant construction steps impose stronger constraints on the generating trajectory.

29. SIMULATION EXPERIMENT: DEGENERACY REDUCTION UNDER RECURSIVE CONSTRAINT ACCUMULATION

29.1. Experimental Objective

We now give a toy simulation illustrating the central claim of the recursive knowledge framework: repeated updates to a persistent knowledge state reduce the degeneracy of the feasible set and move the system from a prior-dominant regime toward a constraint-dominant one.

The purpose of the experiment is not to evaluate a specific commercial model or benchmark a production retrieval system. Rather, it is to exhibit in controlled form the structural phenomenon established in the preceding sections. The simulation is therefore deliberately minimal. It isolates the effect of recursive constraint accumulation from the many confounding variables present in large-scale deployed systems.

29.2. Synthetic Knowledge Space

Let $\mathcal{U} = \{c_1, c_2, \ldots, c_N\}$ be a finite universe of atomic concepts. A latent ground-truth domain is specified by a hidden relational structure

$$G^\star = (V^\star, E^\star),$$

where $V^\star \subseteq \mathcal{U}$ is a set of active concepts and $E^\star \subseteq V^\star \times V^\star$ is a set of semantic relations.

A knowledge state at iteration t is represented by a triple

$$X_t = (V_t, E_t, M_t),$$

where V_t is the set of concepts currently represented in the knowledge base, E_t is the set of explicit links between them, and M_t is a metadata layer consisting of summaries, tags, and index entries.

The system does not observe $G^\star$ directly. Instead, it receives partial projections in the form of raw source fragments, query results, and generated summaries.

29.3. *Observation Model*

At each iteration t, the system receives an observation

$$o_t = \Pi(G^\star, q_t, \xi_t),$$

where q_t is a query and ξ_t is a noise or incompleteness variable.

The projection operator Π is non-injective. Distinct hidden structures may yield identical local observations. In particular, the system may observe only a subset of the relevant vertices and edges, together with partial textual evidence for their existence.

The feasible set at time t is

$$\mathcal{F}_t := \{X \in \mathcal{A}_t : \Pi(X, q_t) = o_t\},$$

where $\mathcal{A}_t$ is the admissible set induced by the current knowledge base and its consistency conditions.

29.4. *Recursive Update Rule*

The simulation implements the recursive update operator

$$X_{t+1} = \mathcal{C}(X_t, q_t, o_t),$$

through the following steps.

First, the system retrieves a subset of the current state relevant to q_t. Second, it generates a provisional answer or summary from the retrieved material and the new observation o_t. Third, it extracts newly inferred concepts, relations, and summaries. Fourth, it writes these back into the persistent state, updating V_t, E_t, and M_t.

Formally, we write

$$V_{t+1} = V_t \cup \Delta V_t,$$

$$E_{t+1} = E_t \cup \Delta E_t,$$

$$M_{t+1} = M_t \cup \Delta M_t.$$

The increments ΔV_t, ΔE_t, and ΔM_t are accepted only if they satisfy a consistency filter. This filter rejects updates that contradict already stabilized relations unless sufficient supporting evidence is present.

29.5. Degeneracy Metric

To measure degeneracy, we define an effective ambiguity functional. Let

$$\mathcal{F}_t^{(K)} = \{X^{(1)}, \ldots, X^{(K)}\}$$

be a Monte Carlo approximation of the feasible set obtained by sampling K candidate reconstructions consistent with the current observations and constraints.

We then define the empirical degeneracy score

$$D_t := \frac{1}{K(K-1)} \sum_{i \neq j} d\big(X^{(i)}, X^{(j)}\big),$$

where $d(\cdot, \cdot)$ is a structural distance, such as graph edit distance or a weighted discrepancy over vertices, edges, and metadata.

Large values of D_t indicate that many structurally distinct reconstructions remain admissible. Small values indicate that the admissible region has collapsed toward a unique or nearly unique configuration.

A complementary metric is the feasible-set cardinality estimate

$$C_t := |\mathcal{F}_t^{(K)}(\epsilon)|,$$

where $\mathcal{F}_t^{(K)}(\epsilon)$ denotes the subset of sampled reconstructions lying within a tolerance ϵ of the observation. In practice, C_t is an approximate count, but its monotonic trend is the main object of interest.

29.6. Prior Dependence Metric

To track the transition from prior-dominant to constraint-dominant reconstruction, we introduce a prior sensitivity score. Let Φ_1 and Φ_2 be two distinct priors over configurations. Using the same observation history, we compute two reconstructions:

$$\hat{X}_t^{(1)} = \arg\min_{X \in \mathcal{F}_t} \Phi_1(X), \qquad \hat{X}_t^{(2)} = \arg\min_{X \in \mathcal{F}_t} \Phi_2(X).$$

We then define

$$P_t := d\big(\hat{X}_t^{(1)}, \hat{X}_t^{(2)}\big).$$

If P_t is large, reconstruction depends strongly on prior choice. If P_t tends to zero over time, the system has become increasingly constrained by accumulated structure rather than prior bias.

29.7. Experimental Conditions

We compare two regimes.

In the first regime, the system is stateless. At each iteration, it answers the query using only the current observation o_t and a fixed prior, without writing anything back to persistent storage. Formally, this means

$$X_{t+1} = X_t,$$

so no constraint accumulation occurs.

In the second regime, the system is recursive and persistent. Each iteration updates the knowledge state using the operator $\mathcal{C}$, and the resulting structures are retained for future queries.

Both regimes receive the same sequence of queries and observations. The only difference is whether outputs are reintegrated into the state.

29.8. Expected Dynamics

The theory developed in earlier sections predicts the following.

In the stateless regime, the degeneracy score D_t should remain high or fluctuate without sustained downward trend. The prior sensitivity P_t should remain bounded away from zero, indicating persistent dependence on external priors.

In the recursive regime, both D_t and P_t should decrease over time. The feasible set should shrink as links, summaries, and structural relations accumulate. After sufficiently many iterations, the system should enter a regime in which different priors produce nearly identical reconstructions because the admissible structure has become highly constrained.

This predicts a qualitative phase transition:

$$\text{stateless inference} \;\Rightarrow\; \text{persistent ambiguity},$$

whereas

$$\text{recursive accumulation} \;\Rightarrow\; \text{progressive identifiability}.$$

29.9. A Minimal Generative Model

For concreteness, we may instantiate the simulation with a random hidden graph $G^\star$ on N vertices. Each query q_t selects a local neighborhood in $G^\star$, and the observation o_t consists of a noisy summary of this neighborhood together with a subset of incident edges.

The system proposes candidate reconstructions by sampling graphs compatible with the observed local neighborhoods and with the current knowledge state. The update operator adds any newly stabilized vertices or edges that appear with sufficiently high posterior support across candidates.

A simple realization is:

$$\Delta E_t = \{e : \Pr(e \mid X_t, o_t) > \tau_E\}, \qquad \Delta V_t = \{v : \Pr(v \mid X_t, o_t) > \tau_V\},$$

for thresholds $\tau_E, \tau_V \in (0, 1)$.

This yields a monotone growth process in explicit structure, accompanied by a reduction in admissible alternatives.

29.10. *Simulation Hypothesis*

The core hypothesis of the experiment is that there exists a decreasing function f such that, in expectation,

$$\mathbb{E}[D_t] \leq f(t), \qquad \mathbb{E}[P_t] \leq f(t),$$

for the recursive regime, while no comparable monotone decrease holds in the stateless regime.

A stronger form of the hypothesis is that, under non-redundant updates and sufficiently informative queries,

$$\lim_{t \to \infty} D_t = 0, \qquad \lim_{t \to \infty} P_t = 0.$$

In that case, recursive knowledge accumulation asymptotically restores identifiability.

29.11. *Interpretation of Results*

If the simulated recursive system exhibits decreasing degeneracy and decreasing prior sensitivity, then the experiment supports the thesis that persistent knowledge organization functions as a mechanism of constraint accumulation. The system does not become more accurate merely by remembering more tokens. It becomes more identifiable because it constructs a better-organized admissible subspace.

If, by contrast, the stateless system fails to reduce ambiguity under repeated queries, then this supports the claim that large context windows alone are not sufficient for structural convergence. What matters is not raw throughput but persistent organization.

The simulation therefore makes visible the central distinction of the paper: a memoryless generator repeatedly revisits a wide feasible set, whereas a recursive knowledge system progressively narrows it.

29.12. *Relation to the General Theory*

This experiment instantiates the positive branch of the general framework. Earlier sections established that constraint failure produces diffusion, degeneracy, and prior-dominant reconstruction. The present simulation explores the inverse direction:

$$\text{recursive update} \Rightarrow \text{constraint accumulation} \Rightarrow \text{degeneracy reduction} \Rightarrow \text{identifiability}.$$

Thus, the experiment serves as a constructive complement to the no-free-reconstruction result. It shows that while uniqueness cannot be recovered from a single degenerate projection, it may be approached through iterative enrichment of the constraint set.

□

29.13. Limitation: Absence of Projection Analysis

However, assembly theory operates entirely at the level of generation and does not address the inverse problem of reconstruction from observation.

In particular, the assembly index a_O is a property of the minimal construction path, not of the degeneracy of the observation-consistent set under projection. Two objects with identical assembly indices may differ substantially in the extent to which their generating processes are identifiable from their observable representations.

The present framework introduces effective degeneracy $H(\mathcal{X}_o^{\mathcal{P}})$ as the relevant quantity for this inverse problem. While assembly index constrains the forward generative complexity of an object, effective degeneracy determines the recoverability of that object from its projection.

29.14. Forward vs Inverse Complexity

This distinction can be formalized as follows:

- Assembly index a_O measures *forward complexity*: the minimal effort required to construct an object.
- Effective degeneracy $H(\mathcal{X}_o^{\mathcal{P}})$ measures *inverse complexity*: the uncertainty remaining about the generating process given an observation.

High forward complexity does not imply low inverse complexity. An object may be difficult to construct yet easy to reconstruct if the prior strongly constrains the space of admissible generators.

29.15. Interpretation

Assembly theory provides a useful characterization of generative difficulty, but does not by itself determine whether an observed artifact serves as reliable evidence of its origin. That determination requires analysis of projection, prior structure, and degeneracy, as developed in the preceding sections.

The distinction mirrors that between effort and inference: the cost of producing an object is not, in general, equal to the information required to identify the process that produced it.

30. IMPLICATIONS FOR SYSTEM DESIGN AND ARGUMENTATIVE PRACTICE

The analysis yields parallel design implications for software systems and written discourse, unified by the principle that the reliability of observable outputs as evidence of underlying structure depends on the degree to which the generating process has achieved constraint closure.

For AI system pipelines, the primary implication is that security cannot be achieved by increasing the vigilance of human operators in high-velocity environments, because the corollary of Section 7 establishes that constraint enforcement probability decreases monotonically with velocity under human-in-the-loop verification. The appropriate remediation is structural: replacing probabilistic human verification with architectural invariants that are enforced by the system itself. Admissibility conditions must be embedded in the pipeline structure rather than delegated to operator judgment. Automation, in this context, functions as a mechanism for achieving constraint closure rather than merely for increasing throughput.

The near-identifiability result of Section 3 has an additional implication for system design. Because reconstruction cost is a decreasing function of prior expressiveness in the reconstruction domain, enclosure strategies that rely on the opacity of the configuration space become less reliable as the shared prior of the field matures. This suggests that the long-term competitive advantage of proprietary systems lies not in the opacity of their implementations but in properties that are genuinely difficult to reconstruct: accumulated data, evaluation infrastructure, deployment scale, and organizational coordination. These are high-dimensional properties whose projections remain genuinely degenerate even under mature priors.

For argumentative practice, the implication is that the specification of constraints is constitutive of the essay rather than preparatory to it. The premises, definitions, tensions, and exclusions that govern an argument are not scaffolding to be removed before the essay begins; they are the constraint structure that gives the essay its necessity. Making this structure explicit in the text increases process legibility and thereby increases the trust that a reader can rationally extend to the argument's conclusions.

The role of generative systems in this context is appropriately understood as a tool for exploring constrained spaces rather than as a source of unconstrained text. When the constraint structure of an argument is specified in advance—through explicit premises, a defined problem, or a set of tensions to be resolved—generative systems can function as efficient navigators of the constrained trajectory space, producing text whose necessity is determined by the specified constraints. Absent such specification, the system produces text that is formally correct but substantively indeterminate, and the burden of constraint remains entirely with the reader, who cannot reconstruct a constraint structure that was never imposed.

31. CONCLUSION

The two phenomena examined in this essay—the accidental release of proprietary AI system code and the degeneration of argumentative essay form under generative pressure—are unified by a common formal structure. Both arise when high-dimensional systems operate under partial constraint, producing outputs whose surface properties substantially exceed their underlying necessity. Both are best understood not as isolated failures but as structurally predictable consequences of systems that have not achieved constraint closure.

The analysis supports a general principle governing the reliability of observable artifacts as evidence of underlying generative structure: in any domain where complex processes produce observable outputs, the reliability of those outputs as evidence depends on the degree to which the generative process was constrained. As the cost of producing surface-plausible outputs decreases—whether through the automation of software deployment or the automation of text generation—the burden of signaling constraint shifts from the artifact to the process. Artifacts that make their constraint structure legible survive as reliable evidence. Artifacts that present surface plausibility without constraint structure become uninformative, regardless of their surface quality.

For software systems, this principle implies that security must be architectural rather than operational, that constraint closure must be structurally enforced rather than behaviorally maintained, and that the effective security of enclosure is bounded above by the reconstruction cost in the prior landscape of the reconstruction domain.

For written discourse, it implies that the essay survives as a form not by resisting the availability of unconstrained generation but by making its constraint structure explicit enough to remain identifiable under projection. The essay that achieves this is one in which every transition is determined by the governing structure, every claim reduces the space of admissible continuations, and the resulting trajectory is sufficiently constrained that a reader can reconstruct the argumentative process from the observable text. This is not a new standard for essays; it is the original one, newly visible because the prior landscape of text production has changed enough to make its absence obvious.

Modern AI systems are best understood as dynamical fields subject to projection, constraint, and diffusion rather than as static artifacts amenable to simple notions of security or ownership. Modern argumentative texts are best understood as trajectories through constrained spaces rather than as documents containing information. In both cases, the surface is a projection of an underlying process, and the value of the surface as evidence depends entirely on what can be inferred about that process from the projection.

APPENDICES

A. CONSTRAINT DENSITY AND INFORMATIONAL IDENTIFIABILITY

This appendix formalizes the notion of constraint density introduced informally in Sections 10 through 12 and establishes its relationship to informational gain and the identifiability of generating processes under projection.

A.1. Constraint Density

Let Y denote an observable text and let $\mathcal{T}(Y)$ denote the set of argumentative trajectories consistent with Y. Let $\mathcal{T}_0$ denote the space of all admissible trajectories in the absence of constraints.

Definition A.1 (Constraint density)**.** The constraint density of Y is defined as

$$D(Y) = 1 - \frac{H(\mathcal{T}(Y))}{H(\mathcal{T}_0)},$$

where $H(\cdot)$ denotes the entropy of the corresponding trajectory distribution under a reference prior over $\mathcal{T}_0$.

Intuitively, $D(Y)$ measures the fraction of the original trajectory space eliminated by the constraints encoded in Y. A text with $D(Y) \approx 0$ imposes few constraints and leaves most trajectories admissible. A text with $D(Y) \approx 1$ sharply restricts the space of admissible trajectories.

Remark A.2. Constraint density is invariant under reparameterizations of $\mathcal{T}_0$ that preserve entropy, and thus depends only on the relative reduction of the trajectory space rather than its absolute representation.

A.2. Constraint Density and Mutual Information

Let G denote the generating process, modeled as a random variable taking values in a space $\mathcal{G}$ of admissible generators. The observable text Y is a projection of G through the generation map.

Theorem A.3 (Constraint density bounds mutual information)**.** *Let $I(Y; G)$ denote the mutual information between the observable text and the generating process. Then, under* *mild regularity conditions,*

$$I(Y;G) \geq H(G) \cdot D(Y),$$

up to a constant factor determined by the alignment between the trajectory space $\mathcal{T}$ and the generator space $\mathcal{G}$.

Proof. The observable text Y restricts the space of admissible trajectories from $\mathcal{T}_0$ to $\mathcal{T}(Y)$. This induces a restriction on the space of generators consistent with Y. The entropy reduction in the trajectory space provides a lower bound on the entropy reduction in the generator space, since each generator induces a trajectory. The result follows from the data processing inequality applied to the mapping $G \mapsto \mathcal{T} \mapsto Y$. □

This result formalizes the intuition that constraint density is a proxy for informational content: texts that eliminate more admissible trajectories provide more information about the generating process.

A.3. Identifiability Threshold

Definition A.4 (Identifiability threshold)**.** Let $\epsilon > 0$. A text Y is ϵ-identifiable if the posterior distribution over generators satisfies

$$H(G \mid Y) \leq \epsilon.$$

Proposition A.5 (Threshold condition)**.** *There exists a threshold $D^* \in (0,1)$ such that if $D(Y) > D^*$, then Y is ϵ-identifiable for sufficiently small ϵ.*

Proof. As $D(Y) \to 1$, the entropy of $\mathcal{T}(Y)$ approaches zero, implying that the trajectory is nearly unique. Since generators that produce distinct trajectories must differ in their internal structure, the posterior over G concentrates. Continuity of entropy implies the existence of a threshold D^* beyond which $H(G \mid Y)$ falls below ϵ. □

This establishes a formal criterion for when an essay functions as reliable evidence of its generating process: it must exceed a critical constraint density.

A.4. Degenerate Regime

Definition A.6 (Degenerate text)**.** A text Y is degenerate if $D(Y) \approx 0$.

In this regime, the observation-consistent set over trajectories is nearly identical to the unconstrained space. Consequently, the posterior over generators remains diffuse, and the text provides negligible information about the process that produced it.

Corollary A.7. *In the degenerate regime, $I(Y;G) \approx 0$, and the text is informationally indistinguishable from a sample drawn directly from the prior over generators.*

This formalizes the notion of essay-shaped text that lacks argumentative content: it is a projection that fails to reduce uncertainty about its origin.

A.5. Constraint Density and Length

Finally, we relate constraint density to length.

Proposition A.8 (Length is neither necessary nor sufficient)**.** *There exist texts Y_1, Y_2 such that $len(Y_1) > len(Y_2)$ but $D(Y_1) < D(Y_2)$.*

Proof. Construct Y_1 as a long sequence of locally plausible but unconstrained statements, and Y_2 as a short sequence of tightly constrained logical steps. The entropy reduction induced by Y_2 exceeds that of Y_1, despite its shorter length. □

This result formalizes the collapse of length as a reliability signal discussed in Section 12. Constraint density, not length, determines the informational value of a text.

A.6. Interpretation

The formal results of this appendix support the central thesis of the essay. Essays that achieve high constraint density function as identifiable projections of their generating processes and therefore serve as reliable evidence of reasoning. Essays that fail to impose constraints remain in the degenerate regime, where their surface structure provides no information about their origin.

In the presence of generative systems capable of producing low-density text at scale, the distribution of observed texts shifts toward the degenerate regime. The task of both writer and reader becomes one of restoring or detecting constraint density sufficient to cross the identifiability threshold.

B. CONSTRAINT DENSITY, KOLMOGOROV COMPLEXITY, AND MDL

This appendix connects the notion of constraint density introduced in Appendix A to Kolmogorov complexity and the Minimum Description Length (MDL) principle. The goal is to relate the identifiability of argumentative trajectories to the compressibility of the generating process.

B.1. Kolmogorov Complexity of Generating Processes

Let G denote a generating process and Y the observable text it produces. Let $K(G)$ denote the Kolmogorov complexity of G, defined as the length of the shortest program that produces G on a fixed universal Turing machine.

Let $K(Y)$ denote the Kolmogorov complexity of the text. In general, $K(Y) \leq K(G) + c$, since Y can be generated by simulating G and recording its output.

However, the inverse problem is of primary interest: given Y, what is the minimal description length of a generator consistent with Y?

Definition B.1 (Minimal generating description)**.** The minimal generating description length of Y is

$$K^*(Y) = \min_{G: G \mapsto Y} K(G).$$

This quantity measures the complexity of the simplest process capable of producing the observed text.

B.2. Constraint Density and Compression

Constraint density can be reinterpreted as a measure of how strongly Y restricts the class of generators consistent with it. When $D(Y)$ is high, the set

$$\mathcal{G}_Y = \{G : G \mapsto Y\}$$

is small and concentrated, implying that $K^*(Y)$ is close to the true complexity of the generating process.

When $D(Y)$ is low, $\mathcal{G}_Y$ is large and diffuse, and there exist many simple generators that can produce Y. In this case, $K^*(Y)$ may be much smaller than the complexity of the original generating process.

Proposition B.2 (Constraint density and minimal description length)**.** *Let Y be an observable text. Then $D(Y)$ is monotonically increasing in the expected value of $K^*(Y)$ over the prior on generating processes.*

Proof. As constraint density increases, the admissible set $\mathcal{G}_Y$ shrinks, eliminating low-complexity generators that would otherwise be consistent with Y. Thus the minimum of $K(G)$ over $\mathcal{G}_Y$ increases. Monotonicity follows from the fact that adding constraints can only reduce the admissible set. □

This result provides a compression-theoretic interpretation of constraint density: texts with high constraint density require more complex generators to produce them, while low-density texts can be generated by simple, generic processes.

B.3. Minimum Description Length Principle

The MDL principle states that the best explanation for data Y is the one that minimizes the total description length

$$L(G, Y) = K(G) + K(Y \mid G),$$

where $K(Y \mid G)$ is the conditional description length of Y given G.

In the present context, $K(Y \mid G)$ is small when G deterministically generates Y, so the MDL objective reduces approximately to minimizing $K(G)$ over generators consistent with Y.

Definition B.3 (MDL-optimal generator)**.** An MDL-optimal generator for Y is

$$G^* = \arg\min_{G: G \mapsto Y} K(G).$$

The MDL principle therefore selects the simplest generator consistent with the observed text. This has different consequences depending on the constraint density of Y.

B.4. Degenerate Regime Under MDL

When $D(Y)$ is low, there exist simple generators that produce Y without encoding any meaningful constraint structure. The MDL-optimal generator in this case is a low-complexity process that reproduces the surface form of the text without instantiating an underlying argument.

Corollary B.4 (MDL collapse in the degenerate regime)**.** *If $D(Y) \approx 0$, then the MDL-optimal generator G^* is drawn from the class of generic prior-based generators, and does not recover the original generating process.*

This formalizes the intuition that essay-shaped text produced without constraints can be explained by a simple generative model, and therefore carries little information about any deeper reasoning process.

B.5. High-Density Regime and Generator Recovery

When $D(Y)$ is high, the admissible generator set $\mathcal{G}_Y$ excludes simple generators. Any generator capable of producing Y must encode the constraint structure that determines the trajectory.

Theorem B.5 (Recovery under high constraint density)**.** *If $D(Y) > D^*$ for some threshold D^*, then the MDL-optimal generator G^* approximates the true generating process up to equivalence under the constraint structure.*

Proof. For sufficiently high constraint density, all generators consistent with Y must encode the same constraint structure. Differences between generators are confined to representations that are equivalent under this structure. Since MDL selects the simplest such generator, it recovers a representative of the equivalence class of true generators. □

This establishes a compression-theoretic analogue of the identifiability threshold from Appendix A.

B.6. Interpretation

The connection to Kolmogorov complexity and MDL reinforces the central thesis of the essay. The value of an argumentative text lies in its ability to constrain the space of generators sufficiently that the simplest explanation for the text must encode the structure of the argument itself.

In the absence of such constraints, the simplest explanation is a generic generator that produces essay-shaped text without underlying necessity. The text is then compressible in a way that discards any interpretation of it as a structured argument.

Thus, constraint density serves as a bridge between information-theoretic, computational, and epistemological perspectives. It determines not only whether a text is informative about its generating process, but whether that process can be recovered through compression-based inference.

C. ASSEMBLY INDEX OF CONSTRAINT STRUCTURES

This appendix provides a structured mapping between the formal objects introduced in the main text and their operational, geometric, and statistical interpretations. Its purpose is to unify the theoretical machinery with observable diagnostics, thereby reducing the gap between abstract analysis and empirical evaluation.

C.1. Configuration Space Decomposition

Let the configuration space $\mathcal{X}$ be decomposed as a product

$$\mathcal{X} = \mathcal{X}_{\text{code}} \times \mathcal{X}_{\text{infra}} \times \mathcal{X}_{\text{workflow}} \times \mathcal{X}_{\text{latent}}.$$

Each component corresponds to a distinct class of constraints:

- $\mathcal{X}_{\text{code}}$: syntactic and semantic program structure.
- $\mathcal{X}_{\text{infra}}$: deployment environment, access controls, and system topology.
- $\mathcal{X}_{\text{workflow}}$: sequencing constraints and transition rules.
- $\mathcal{X}_{\text{latent}}$: informal practices, conventions, and tacit knowledge.

The latent component is not directly observable but induces a prior $\mathcal{P}$ over $\mathcal{X}$, shaping effective degeneracy.

C.2. Hessian Structure of Reconstruction Energy

Recall the reconstruction energy

$$\mathcal{E}(X) = \|\Pi(X) - o\|^2 + \lambda(-\log \mathcal{P}(X)).$$

The Hessian of $\mathcal{E}$ is

$$\nabla^2 \mathcal{E}(X) = \nabla^2 \|\Pi(X) - o\|^2 + \lambda \nabla^2(-\log \mathcal{P}(X)).$$

The first term encodes projection curvature, while the second encodes prior curvature.

Definition C.1 (Condition number)**.** Let κ denote the ratio of the largest to smallest eigenvalues of $\nabla^2 \mathcal{E}$.

Proposition C.2 (Convergence rate)**.** *The convergence rate of reconstruction is inversely proportional to κ. Low κ implies rapid convergence to a prior-dominant attractor.*

This formalizes the empirical observation that reconstruction is fast when the prior strongly constrains admissible configurations.

C.3. Diffusion on Network Topologies

Let the information network be represented as a graph $G = (V, E)$ with Laplacian L. The diffusion equation becomes

$$\frac{\partial \rho}{\partial t} = -DL\rho.$$

The spectrum of L determines the rate and pattern of diffusion.

Proposition C.3 (Spectral diffusion rate)**.** *The second-smallest eigenvalue λ_2 of L (the algebraic connectivity) controls the speed of global information spread.*

Highly connected networks (large λ_2) accelerate entropy diffusion, reducing the feasibility of containment after leakage.

C.4. Stochastic Modeling of Workflow Constraints

Let the pipeline be modeled as a Markov chain with states $s \in S$ and transition matrix P.

Manual steps introduce stochastic transitions:

$$P(s_{t+1} \mid s_t) = p_i \cdot \delta_{\text{adm}} + (1 - p_i) \cdot \delta_{\text{escape}},$$

where p_i depends on cognitive load and iteration velocity v.

Proposition C.4 (Velocity-dependent failure)**.** *If $p_i = p_i(v)$ is decreasing in v, then the probability of remaining within S_{adm} decays exponentially in the number of manual steps.*

This provides a stochastic foundation for trajectory escape under partial closure.

C.5. Metric for Argumentative Necessity

Let Y be a text and let Δ_i denote a perturbation of component i.

Define the necessity metric

$$N(Y) = \mathbb{E}_i\big[d(Y, \Delta_i(Y))\big],$$

where d measures structural degradation.

Proposition C.5 (Necessity-density relation)**.** *$N(Y)$ is monotonically increasing in constraint density $D(Y)$.*

High necessity implies that perturbations disrupt coherence, indicating strong constraint structure. Low necessity implies robustness under modification, indicating degeneracy.

C.6. Unified Mapping

The following correspondences summarize the structural equivalences developed in the essay:

Formal Object	Operational Interpretation
Projection Π	Observable artifact (code or text)
Prior $\mathcal{P}$	Shared conventions / training distribution
Fiber $\mathcal{X}_o$	Space of admissible reconstructions
Entropy H	Effective degeneracy
Energy $\mathcal{E}$	Reconstruction objective
Hessian $\nabla^2\mathcal{E}$	Reconstruction stability / convergence
Laplacian L	Network diffusion structure
Markov transitions P	Workflow reliability
Constraint density D	Informational clarity
Necessity N	Fragility under perturbation

C.7. Interpretation

The assembly index makes explicit that the same mathematical structures govern both technical systems and argumentative texts. Each object admits both a formal definition and an observable proxy, allowing inference to move between levels.

The central implication is that failures of security, clarity, and trust are not domain-specific phenomena. They arise from the same underlying condition: insufficient constraint relative to the dimensionality of the system.

The role of the appendix is therefore not merely to collect technical details, but to demonstrate that the theoretical framework is closed under interpretation. Every formal object corresponds to a measurable or observable property, and every observable failure can be traced back to a breakdown in constraint structure.

A. APPENDIX D: INSTANTIATIONS IN GENERATIVE MODELS AND RSVP SYSTEMS

A.1. D.1 Autoregressive Generation as Variational Trajectory

Let $\mathcal{V}$ be a token vocabulary and let a sequence be denoted

$$x_{1:T} = (x_1, x_2, \ldots, x_T), \quad x_t \in \mathcal{V}.$$

An autoregressive model defines a probability distribution

$$P(x_{1:T}) = \prod_{t=1}^{T} P(x_t \mid x_{<t}),$$

which induces a trajectory through the sequence space $\mathcal{X} = \mathcal{V}^T$.

This process can be interpreted as minimizing a local energy functional at each step:

$$x_t = \arg\min_{x \in \mathcal{V}} \left[-\log P(x \mid x_{<t}) \right].$$

Thus, generation is a discrete-time trajectory:

$$X(0) \to X(1) \to \cdots \to X(T),$$

where $X(t) = x_{1:t}$.

A.2. D.2 Absence of Global Constraint Closure

In the absence of external constraints, the model optimizes only local likelihood. There is no global functional enforcing structural consistency across the entire trajectory.

Let $\mathcal{C}$ denote a set of global constraints (e.g., logical consistency, argumentative necessity). Standard autoregressive decoding does not explicitly enforce

$$X(t) \in \mathcal{A}_{\mathcal{C}} \quad \forall t,$$

where $\mathcal{A}_{\mathcal{C}}$ is the constraint-admissible subset.

Instead, each step is conditionally optimal but globally unconstrained. Hence:

Proposition A.1 (Local Coherence without Global Necessity)**.** *Autoregressive decoding produces sequences that are locally coherent but need not satisfy any global constraint closure condition.*

Proof. Each token is selected to maximize $P(x_t \mid x_{<t})$, independent of future global structure. No global constraint functional appears in the optimization. □

A.3. D.3 Prior-Dominant Attractors in Language Models

The training distribution induces an implicit prior Φ over sequences. In the absence of strong conditioning signals, generation collapses toward modes of this distribution.

Let o denote a prompt. The conditional generation problem can be written as

$$x^*_{1:T} = \arg\min_{x_{1:T}} \left[-\log P(x_{1:T} \mid o)\right].$$

When o is weak or underspecified, we have

$$P(x_{1:T} \mid o) \approx P(x_{1:T}),$$

and thus

$$x^*_{1:T} \approx \arg\min \Phi(x_{1:T}),$$

where $\Phi(x) := -\log P(x)$.

This is precisely the prior-dominant regime defined in the main text.

A.4. D.4 Informational Emptiness in Generated Text

Let H denote latent structural variables (e.g., argument graph, proof structure), and O the generated text.

In unconstrained generation, O is sampled from a distribution shaped by surface statistics rather than structural necessity. Hence

$$I(O; H) \approx 0,$$

even when the entropy $H(O)$ is large.

Proposition A.2 (Replaceability). *If $I(O; H) \approx 0$, then there exist many sequences O' such that*

$$O' \approx O \quad \textit{(in distribution)}$$

while differing arbitrarily in latent structure.

This explains the empirical phenomenon of high-surface-quality but low-substance outputs.

A.5. D.5 RSVP Consistency Operator and Identifiability

We now map the general framework to the RSVP formulation.

Let $X = (\Phi, \mathbf{v}, S)$ denote a field configuration, and let $\Pi_i : X \to \mathcal{Y}_i$ be sensor projections. Observations y_i define the feasible set

$$\Omega_{\text{obs}} = \{X : \|\Pi_i(X) - y_i\| \le \epsilon_i \ \ \forall i\}.$$

Let Γ denote dynamical admissibility constraints, and define

$$\mathcal{F} = \Omega_{\text{obs}} \cap \Gamma.$$

The consistency operator

$$\mathcal{C} : \mathcal{A} \to \mathcal{A}$$

iteratively refines configurations toward admissibility.

Definition A.3 (Identifiability)**.** The system is identifiable if the induced map

$$\widetilde{\Pi} : X/\sim \to \prod_i \mathcal{Y}_i$$

is injective on $\mathcal{F}$.

A.6. D.6 Correspondence with Projection Degeneracy

The failure modes align exactly:

General Framework	RSVP Formulation
Projection degeneracy	$\widetilde{\Pi}$ non-injective
Constraint failure	Γ weak or violated
Feasible set $\mathcal{F}(o)$ large	$\mathcal{F}$ high-dimensional
Prior collapse	Regularizer $\mathcal{R}$ flat on $\mathcal{F}$

Theorem A.4 (RSVP Prior Collapse)**.** *If $\widetilde{\Pi}$ is non-injective on $\mathcal{F}$ and the regularizer $\mathcal{R}$ is not strictly convex on $\mathcal{F}$, then the consistency operator $\mathcal{C}$ admits multiple fixed points. Selection among them is determined by implicit priors.*

Proof. Non-injectivity implies multiple equivalence classes consistent with observations. Lack of strict convexity implies no unique minimizer of $\mathcal{R}$ on $\mathcal{F}$. Hence multiple fixed points exist. □

A.7. D.7 Unified Interpretation

Both generative language models and RSVP reconstruction systems obey the same structural law:

$$\text{Non-injective projection} + \text{weak constraint} \;\Rightarrow\; \text{prior-dominant selection.}$$

In RSVP, this manifests as non-identifiability of field configurations. In language models, it manifests as generation of statistically typical but structurally unconstrained sequences.

A.8. D.8 Consequence: Necessity as a Diagnostic

Let O be an output and consider perturbations δO.

Definition A.5 (Necessity)**.** An output exhibits necessity if small perturbations destroy admissibility:

$$O + \delta O \notin \mathcal{A} \quad \text{for generic } \delta O.$$

Proposition A.6 (Diagnostic Criterion)**.** *Constraint-closed systems produce outputs with high necessity. Prior-dominant systems produce outputs robust under perturbation.*

Thus, fragility under modification is a structural indicator of genuine constraint closure.

□

B. APPENDIX E: THERMODYNAMIC IRREVERSIBILITY AND INFORMATION DIFFUSION

B.1. E.1 Information as a Distributed State

Let $\Omega \subset \mathbb{R}^n$ denote a network domain (e.g., computational nodes, storage locations, agents). Let

$$\rho(x, t) \geq 0$$

represent the density of information (e.g., code fragments, model weights, knowledge states) at position $x \in \Omega$ and time $t \geq 0$.

We model the dissemination of information as a diffusion process:

$$\frac{\partial \rho}{\partial t} = D\nabla^2 \rho,$$

where $D > 0$ is an effective diffusion coefficient.

B.2. E.2 Entropy and Irreversibility

Define the entropy functional

$$S[\rho] = -\int_\Omega \rho(x,t) \log \rho(x,t)\, dx.$$

Proposition B.1 (Entropy Increase)**.** *For solutions of the diffusion equation with suitable boundary conditions,*

$$\frac{dS}{dt} \geq 0.$$

Proof. Standard result following from integration by parts and positivity of $D \int |\nabla \log \rho|^2 \rho\, dx$. □

Thus, information diffusion is intrinsically irreversible: the system evolves toward maximally dispersed configurations.

B.3. E.3 Boundary Rupture as Initial Condition

Let an initially confined system satisfy

$$\rho(x,0) = \rho_0(x), \quad \operatorname{supp}(\rho_0) \subset U \subset \Omega,$$

where U is a restricted domain (e.g., private repository, secured system).

A leak corresponds to removal or failure of the boundary ∂U, allowing diffusion into $\Omega \setminus U$.

Definition B.2 (Leak Event)**.** A leak is the transition from a constrained domain U to an unconstrained domain Ω, initiating diffusion of ρ.

After the leak, the support of $\rho(x,t)$ expands monotonically.

B.4. E.4 Impossibility of Reversal

Theorem B.3 (Irreversibility of Dissemination). *Let $\rho(x,t)$ evolve under diffusion from an initial confined state. There does not exist a physically realizable process that restores $\rho(x,t)$ to its initial support U without external work exceeding the system's entropy increase.*

Proof. Reversal would require decreasing entropy, contradicting the monotonicity of $S[\rho]$ unless compensated by external work. Such work scales with the entropy increase and is infeasible in distributed systems. □

Thus, once information has diffused, it cannot be recollected by local operations.

B.5. E.5 Ineffectiveness of Local Control Actions

Let $\mathcal{L}$ denote a set of local interventions (e.g., deletion requests, access restrictions).

These act on subsets $V \subset \Omega$:

$$\rho(x,t) \mapsto \rho'(x,t) \quad \text{for } x \in V.$$

Proposition B.4 (Local Control Insufficiency). *If ρ has nonzero support outside V, then local interventions cannot eliminate global presence:*

$$\int_{\Omega \setminus V} \rho(x,t)\, dx > 0.$$

Thus, local enforcement cannot reverse global diffusion.

B.6. E.6 Effective Degeneracy under Diffusion

As ρ spreads, copies of the information exist in multiple locations and forms. Let $\mathcal{X}$ denote the space of system configurations, and define equivalence classes under replication:

$$X \sim X' \quad \text{if they encode equivalent information.}$$

Definition B.5 (Effective Degeneracy). Effective degeneracy is the cardinality of equivalence classes consistent with observable states:

$$|\mathcal{F}(o)| \uparrow \quad \text{as } \rho \text{ diffuses.}$$

Diffusion increases degeneracy by proliferating indistinguishable instances of information.

B.7. E.7 Interaction with Prior-Dominant Reconstruction

As information spreads, reconstruction becomes easier due to the increased availability of partial observations across the domain, which provide multiple overlapping constraints on the underlying structure. At the same time, redundancy emerges through distributed representations across Ω, allowing fragments of information to be recombined in different ways. This process is further reinforced by alignment with shared priors encoded in distributed agents, which bias reconstruction toward commonly reinforced patterns and interpretations.

Proposition B.6 (Diffusion-Accelerated Reconstruction)**.** *Let ρ be widely distributed. Then the effective cost of reconstructing a configuration X consistent with o decreases as a function of the support of ρ.*

Thus, diffusion not only prevents reversal but actively facilitates reconstruction.

B.8. E.8 Entropic Interpretation of Computational Enclosure

Let U represent a computational enclosure (e.g., proprietary system). Its integrity corresponds to low entropy:

$$S[\rho] \text{ small}, \quad \text{supp}(\rho) \subset U.$$

Over time, due to diffusion and dissemination of knowledge, we have:

$$S[\rho] \to S_{\max}, \quad \text{supp}(\rho) \to \Omega.$$

Proposition B.7 (Enclosure Decay)**.** *The effective strength of a computational enclosure decreases monotonically with the entropy of ρ.*

Thus, enclosure is not a static property but a metastable low-entropy configuration.

B.9. E.9 Synthesis

Information dissemination obeys thermodynamic principles in that diffusion increases entropy and spreads information across the accessible space, progressively reducing its localization. Reversal of this process would require infeasible global coordination, as it would necessitate reconstructing the precise microstate from a highly dispersed configuration. Consequently, local interventions are insufficient to restore confinement once diffusion has occurred. As a result, degeneracy increases and reconstructability becomes more ambiguous as the spread of information grows.

Combined with the results of Appendix D, we obtain:

$$\text{diffusion} \;\Rightarrow\; \text{increased degeneracy} \;\Rightarrow\; \text{prior-dominant reconstruction.}$$

Thus, once constraint boundaries are breached, both physical containment and structural uniqueness are irreversibly lost.

□

C. APPENDIX F: DERIVED AND ∞-CATEGORICAL FORMULATION OF RECONSTRUCTION AND PRIOR SELECTION

C.1. F.1 Moduli of Configurations

Let $\mathcal{C}$ be a higher category (e.g., an ∞-category) of configurations. Rather than treating configurations as isolated objects, we consider their moduli space:

$$\mathfrak{M} := \mathrm{Obj}(\mathcal{C}) / \sim,$$

where $\sim$ denotes equivalence up to higher morphisms (gauge transformations, reparameterizations, semantic equivalences).

Thus, $\mathfrak{M}$ is a derived moduli space encoding not only configurations but their deformation structure.

C.2. F.2 Projection as a Map of Stacks

Let $\mathcal{D}$ be an ∞-category of observables. The projection functor lifts to a morphism of stacks:

$$\Pi : \mathfrak{M} \to \mathfrak{O},$$

where $\mathfrak{O}$ is the moduli stack of observations.

For a fixed observation $o \in \mathfrak{O}$, the space of compatible configurations is the homotopy fiber:

$$\mathfrak{F}_o := \mathfrak{M} \times_{\mathfrak{O}} \{o\}.$$

Definition C.1 (Derived Feasible Space)**.** The derived feasible space is the homotopy fiber $\mathfrak{F}_o$, encoding all configurations consistent with o, including higher equivalences.

C.3. F.3 Identifiability as Contractibility

Definition C.2 (Derived Identifiability)**.** The system is identifiable at o if the homotopy fiber $\mathfrak{F}_o$ is contractible:

$$\mathfrak{F}_o \simeq *.$$

Contractibility implies existence of a unique configuration up to higher equivalence.

Failure of identifiability corresponds to nontrivial homotopy:

$$\pi_k(\mathfrak{F}_o) \neq 0 \quad \text{for some } k \geq 0.$$

C.4. F.4 Constraint as a Derived Substack

Let $\mathfrak{A} \subset \mathfrak{M}$ be a substack of admissible configurations defined by constraints.

Constraint closure corresponds to restricting the projection:

$$\Pi|_{\mathfrak{A}} : \mathfrak{A} \to \mathfrak{O}$$

such that the fiber

$$\mathfrak{F}_o^{\mathfrak{A}} := \mathfrak{A} \times_{\mathfrak{O}} \{o\}$$

is contractible.

Proposition C.3 (Constraint Closure in Derived Form)**.** *Constraint closure holds if and only if* $\mathfrak{F}_o^{\mathfrak{A}} \simeq *$.

C.5. F.5 Obstruction Theory

Non-contractibility of $\mathfrak{F}_o$ is governed by obstruction classes.

Definition C.4 (Obstruction Class)**.** An obstruction to lifting o uniquely is an element

$$\omega \in H^k(\mathfrak{F}_o; \mathcal{G})$$

for some coefficient system $\mathcal{G}$.

Nonvanishing obstruction classes imply multiple inequivalent lifts.

This generalizes the Čech cohomology obstruction in Appendix E.

C.6. F.6 Prior as a Potential on Moduli

In the derived setting, a prior is naturally interpreted as a functional

$$\Phi : \mathfrak{M} \to \mathbb{R},$$

or more precisely, a potential on the moduli space.

Restriction to the fiber yields

$$\Phi|_{\mathfrak{F}_o} : \mathfrak{F}_o \to \mathbb{R}.$$

Definition C.5 (Prior-Selected Point)**.** A prior selects a configuration

$$X^* \in \arg\min_{X \in \mathfrak{F}_o} \Phi(X).$$

Thus, prior selection corresponds to choosing a point in a non-contractible homotopy type.

C.7. F.7 Homotopy Landscape and Attractors

When $\mathfrak{F}_o$ has nontrivial topology, it can be viewed as a landscape with multiple connected components or higher-dimensional cycles.

Definition C.6 (Prior-Dominant Attractor)**.** A prior-dominant attractor is a local minimum of Φ on $\mathfrak{F}_o$.

Reconstruction becomes gradient flow on the moduli space:

$$\frac{dX}{dt} = -\nabla\Phi(X).$$

Thus, solutions correspond to critical points of Φ rather than unique lifts.

C.8. F.8 Dynamical Escape and Topological Expansion

Trajectory escape corresponds to enlarging the accessible moduli space:

$$\mathfrak{A} \subset \mathfrak{M} \quad \longrightarrow \quad \mathfrak{M}.$$

This induces a change in the fiber:

$$\mathfrak{F}_o^{\mathfrak{A}} \simeq * \quad \longrightarrow \quad \mathfrak{F}_o \text{ non-contractible.}$$

Thus, escape corresponds to a topological phase transition from a trivial to a nontrivial homotopy type.

C.9. F.9 Derived Form of No-Free-Reconstruction

Theorem C.7 (Derived No-Free-Reconstruction)**.** *If the homotopy fiber $\mathfrak{F}_o$ is not contractible, then no canonical section*

$$s : \mathfrak{O} \to \mathfrak{M}$$

exists selecting a unique lift of o.

Any selection depends on additional structure (e.g., a prior Φ).

Proof. A canonical section would define a continuous choice of representative in each fiber, implying contractibility. Non-contractibility obstructs such a selection. □

C.10. F.10 Final Interpretation

At the highest level, reconstruction is the problem of selecting a point in a derived fiber:

$$X^* \in \mathfrak{F}_o.$$

Constraint closure corresponds to trivial topology:

$$\mathfrak{F}_o \simeq *.$$

Failure introduces higher structure:

$$\mathfrak{F}_o \not\simeq *,$$

and reconstruction becomes inherently non-canonical.

Thus, the transition from necessity to replaceability corresponds to a shift from a contractible to a non-contractible moduli space.

□

D. APPENDIX G: ALGORITHMIC REALIZATION OF RECURSIVE CONSTRAINT ACCUMULATION

D.1. G.1 Overview

We provide a constructive realization of the update operator

$$X_{t+1} = \mathcal{C}(X_t, q_t)$$

as a finite sequence of transformations over a persistent knowledge state X_t.

The state X_t is assumed to consist of:

$$X_t = (\mathcal{D}_t, \mathcal{I}_t, \mathcal{G}_t),$$

where $\mathcal{D}_t$ is a document store, $\mathcal{I}_t$ is an index (summaries, backlinks, embeddings), and $\mathcal{G}_t$ is a structural graph over concepts.

D.2. G.2 Operator Decomposition

The update operator $\mathcal{C}$ is decomposed into four stages:

$$\mathcal{C} = \mathcal{U} \circ \mathcal{A} \circ \mathcal{Q} \circ \mathcal{R},$$

corresponding to retrieval, query synthesis, analysis, and update.

D.3. G.3 Pseudocode

[colback=lightgray, colframe=black, title=Recursive Knowledge Update Operator $\mathcal{C}$]

```
Input:
    X_t = (D_t, I_t, G_t)       # knowledge state
    q_t                         # query

Procedure:

1. Retrieval (R):
    S_t âĘŘ retrieve_relevant(D_t, I_t, q_t)

2. Query Expansion (Q):
    Q_t âĘŘ expand_query(q_t, S_t, I_t)
    # includes subquestions, hypotheses, missing links

3. Analysis / Synthesis (A):
    O_t âĘŘ generate_outputs(Q_t, S_t)
    # outputs include:
    # - summaries
    # - new concept pages
    # - links and relations
    # - visualizations / derivations

4. Update (U):
    D_{t+1} âĘŘ D_t âĹł extract_documents(O_t)
    G_{t+1} âĘŘ update_graph(G_t, O_t)
    I_{t+1} âĘŘ update_index(I_t, D_{t+1}, G_{t+1})
```

```
Return:
    X_{t+1} = (D_{t+1}, I_{t+1}, G_{t+1})
```

D.4. G.4 Constraint-Inducing Transformations

Each stage contributes to constraint accumulation:

- Retrieval restricts attention to a subspace of $\mathcal{X}$,
- Query expansion introduces latent constraints,
- Analysis produces structured relations,
- Update enforces persistence and cross-consistency.

Proposition D.1 (Monotonic Constraint Enrichment)**.** *Under non-redundant updates, the transformation $\mathcal{C}$ induces a monotonic refinement of the admissible set:*

$$\mathcal{A}_{t+1} \subseteq \mathcal{A}_t.$$

D.5. G.5 Index as a Projection Control Mechanism

The index $\mathcal{I}_t$ functions as a control layer over the projection Π.

Definition D.2 (Index-Guided Projection)**.** The effective projection at time t is

$$\Pi_t(X, q) := \Pi(X, q \mid \mathcal{I}_t),$$

which restricts evaluation to indexed substructures.

Thus, the system replaces brute-force projection with structured navigation.

D.6. G.6 Fixed Point Approximation

In practice, the system approaches a fixed point X^* when updates become self-consistent:

```
if delta(X_t, X_{t+1}) < epsilon:
    terminate
```

where δ measures structural change (e.g., graph edit distance, document divergence).

D.7. G.7 Complexity Considerations

Let $N_t = |\mathcal{D}_t|$ denote the size of the knowledge base.

- Stateless inference cost: $\mathcal{O}(N_t)$ (full context scanning),
- Indexed retrieval cost: $\mathcal{O}(\log N_t)$ or sublinear,
- Update cost: amortized over iterations.

Thus, recursive systems achieve improved scaling by trading compute for structure.

D.8. G.8 Interpretation

The operator $\mathcal{C}$ implements a closed-loop system:

$$\text{query} \rightarrow \text{structure} \rightarrow \text{constraint} \rightarrow \text{refined state.}$$

Each iteration reduces degeneracy and increases the necessity of subsequent outputs.

□

E. APPENDIX H: MINIMAL IMPLEMENTATION OF RECURSIVE KNOWLEDGE SYSTEMS

E.1. H.1 System Architecture

We describe a minimal implementation of the recursive knowledge system using a filesystem-based architecture.

Let the knowledge state X_t be represented as a directory:

$$X_t \equiv \texttt{wiki/}$$

with the following structure:

```
wiki/
  raw/        # ingested source documents
  pages/      # compiled concept articles (.md)
  index/      # summaries, embeddings, metadata
  graph/      # link structure (edges, backlinks)
  outputs/    # generated answers and artifacts
```

This representation instantiates:

$$X_t = (\mathcal{D}_t, \mathcal{I}_t, \mathcal{G}_t)$$

as concrete files and directories.

E.2. H.2 Core Loop

The update operator $\mathcal{C}$ is implemented as a CLI loop:

[colback=lightgray, colframe=black, title=Main Loop]

```
while True:
    q_t = get_query()

    # 1. Retrieve relevant files
    S_t = search_index("wiki/index/", q_t)

    # 2. Expand query
    Q_t = generate_subqueries(q_t, S_t)

    # 3. Generate outputs
    O_t = run_llm(Q_t, S_t)

    # 4. Write outputs to filesystem
    write_markdown("wiki/outputs/", O_t)

    # 5. Extract structured updates
    new_pages = extract_pages(O_t)
```

```
    update_pages("wiki/pages/", new_pages)

    # 6. Update graph and index
    update_graph("wiki/graph/", new_pages)
    update_index("wiki/index/", new_pages)

end
```

E.3. H.3 Data Ingestion

Raw data is incorporated via:

```
ingest(file):
    save(file, "wiki/raw/")
    summary = summarize(file)
    write(summary, "wiki/pages/")
    update_index(summary)
```

This step initializes constraints by introducing structured representations of external data.

E.4. H.4 Index Construction

The index maintains short summaries for each page, enabling rapid semantic access to document content. It further encodes keyword mappings that support efficient lookup and retrieval across the knowledge base. In addition, it may include optional vector embeddings to provide a continuous similarity structure over documents, as well as cross-reference tables that capture explicit relationships between pages. Together, these components form a layered indexing system that supports both symbolic and statistical navigation of the knowledge state.

```
update_index(new_pages):
    for page in new_pages:
        summary = summarize(page)
        keywords = extract_keywords(page)
        store(summary, keywords)
```

The index enables sublinear retrieval, approximating

$$\Pi_t(X,q) = \Pi(X,q \mid \mathcal{I}_t).$$

E.5. H.5 Graph Construction

The graph encodes relationships:

```
update_graph(new_pages):
    for page in new_pages:
        links = extract_links(page)
        add_edges(page, links)
```

This produces a semantic network approximating $\mathcal{G}_t$.

E.6. H.6 Reintegration of Outputs

Generated outputs are fed back into the system:

```
extract_pages(O_t):
    return parse_markdown(O_t)
```

Thus:

$$X_{t+1} = X_t \cup \Delta(O_t),$$

ensuring accumulation of constraints.

E.7. H.7 Consistency Checks

Periodic validation enforces structural integrity:

```
lint():
    find_inconsistencies()
    suggest_missing_links()
    detect_duplicate concepts()
```

This approximates enforcement of $\mathcal{A}_t$.

E.8. H.8 Termination and Fixed Points

Define a distance metric:

$$\delta(X_t, X_{t+1}) = \text{graph difference} + \text{document divergence}.$$

Terminate when:

```
if delta(X_t, X_{t+1}) < epsilon:
    stop
```

This approximates convergence to a fixed point X^*.

E.9. H.9 Comparison with Context-Based Systems

Property	Context Window Systems	Recursive Knowledge Systems
State	Ephemeral	Persistent
Scaling	Linear in tokens	Sublinear via indexing
Memory	External (prompt)	Internal (filesystem)
Interpretability	Low	High
Cost growth	Increasing	Amortized

E.10. H.10 Interpretation

This implementation demonstrates that recursive constraint accumulation is realizable using simple and accessible tools, without requiring complex or specialized infrastructure. It shows that persistent structure can effectively replace large context windows by externalizing and organizing information in a way that supports efficient retrieval and reasoning. It further indicates that knowledge systems constructed in this manner can converge toward self-consistent states through iterative refinement and reinforcement of constraints.

Thus, the theoretical operator $\mathcal{C}$ corresponds directly to a practical architecture.

□

F. APPENDIX I: EXECUTABLE SIMULATION PROCEDURE

F.1. I.1 Overview

We provide a minimal executable realization of the simulation described in the main text. The goal is to construct a controlled environment in which degeneracy reduction can be observed as a function of recursive constraint accumulation.

The implementation operates on finite graphs and uses stochastic sampling to approximate feasible sets.

F.2. I.2 Data Structures

We represent a knowledge state X_t as a Python object:

```
class KnowledgeState:
    def __init__(self):
        self.V = set()          # vertices (concepts)
        self.E = set()          # edges (relations)
        self.M = dict()         # metadata (summaries, scores)
```

The hidden ground truth is:

```
class GroundTruth:
    def __init__(self, N, p):
        self.V = set(range(N))
        self.E = random_graph_edges(N, p)
```

F.3. I.3 Observation Generator

Each query reveals a partial, noisy subgraph:

```
def observe(G_star, query_size=5, noise=0.1):
    nodes = random.sample(G_star.V, query_size)
    edges = [(u,v) for (u,v) in G_star.E if u in nodes and v in nodes]

    # randomly drop edges (incompleteness)
```

```
    edges = [e for e in edges if random.random() > noise]

    return nodes, edges
```

This implements the non-injective projection operator Π.

F.4. I.4 Candidate Sampling

We approximate the feasible set by sampling candidate reconstructions:

```
def sample_candidate(X_t, observation, prior_bias=0.5):
    nodes_obs, edges_obs = observation

    V = set(X_t.V)
    E = set(X_t.E)

    # incorporate observation
    V.update(nodes_obs)
    E.update(edges_obs)

    # hallucinate edges (prior-driven completion)
    for u in V:
        for v in V:
            if u != v and (u,v) not in E:
                if random.random() < prior_bias:
                    E.add((u,v))

    return (V, E)
```

Different values of `prior_bias` simulate different priors Φ.

F.5. I.5 Degeneracy Estimation

We estimate degeneracy via pairwise distance:

```
def graph_distance(G1, G2):
    V1, E1 = G1
```

```
    V2, E2 = G2

    return len(V1.symmetric_difference(V2)) + \
           len(E1.symmetric_difference(E2))

def degeneracy_score(samples):
    K = len(samples)
    total = 0

    for i in range(K):
        for j in range(i+1, K):
            total += graph_distance(samples[i], samples[j])

    return total / (K * (K-1) / 2)
```

F.6. I.6 Prior Sensitivity

We compute prior dependence by comparing reconstructions under two priors:

```
def prior_sensitivity(X_t, observation):
    G1 = sample_candidate(X_t, observation, prior_bias=0.3)
    G2 = sample_candidate(X_t, observation, prior_bias=0.7)

    return graph_distance(G1, G2)
```

F.7. I.7 Update Operator

The recursive update operator extracts stable structure:

```
def update_state(X_t, samples, threshold=0.7):
    edge_counts = {}

    for V,E in samples:
        for e in E:
            edge_counts[e] = edge_counts.get(e, 0) + 1

    new_edges = set()
```

```
    K = len(samples)

    for e, count in edge_counts.items():
        if count / K > threshold:
            new_edges.add(e)

    X_t.E.update(new_edges)

    for e in new_edges:
        X_t.V.update(e)

    return X_t
```

This enforces:

$$\Delta E_t = \{e : \Pr(e \mid X_t, o_t) > \tau\}.$$

F.8. I.8 Experimental Loop

The full simulation is:

```
def run_simulation(T=50, K=20):
    G_star = GroundTruth(N=30, p=0.1)
    X = KnowledgeState()

    degeneracy = []
    prior_dep = []

    for t in range(T):
        obs = observe(G_star)

        # sample feasible reconstructions
        samples = [sample_candidate(X, obs) for _ in range(K)]

        # metrics
        D_t = degeneracy_score(samples)
        P_t = prior_sensitivity(X, obs)
```

```
        degeneracy.append(D_t)
        prior_dep.append(P_t)

        # recursive update
        X = update_state(X, samples)

    return degeneracy, prior_dep
```

A stateless baseline is obtained by removing the update step:

```
# comment out:
# X = update_state(X, samples)
```

F.9. I.9 Expected Output

The simulation produces two sequences:

$$\{D_t\}_{t=1}^{T}, \quad \{P_t\}_{t=1}^{T}.$$

In the recursive regime, both sequences should exhibit a decreasing trend. In the stateless regime, they should remain approximately stationary.

F.10. I.10 Visualization

A minimal plotting routine:

```
import matplotlib.pyplot as plt

deg, prior = run_simulation()

plt.plot(deg, label="Degeneracy D_t")
plt.plot(prior, label="Prior Sensitivity P_t")
plt.legend()
plt.xlabel("Iteration")
plt.show()
```

This visualizes the transition toward identifiability.

F.11. I.11 Interpretation

This executable construction demonstrates that degeneracy can be empirically approximated through sampling procedures, allowing the feasible set to be probed without requiring explicit enumeration. It further shows that recursive updates induce a measurable contraction of this feasible set, as accumulated structure progressively eliminates admissible alternatives. At the same time, prior dependence is observed to decrease as constraints accumulate, indicating a transition away from prior-dominant reconstruction toward constraint-dominant behavior.

Thus, the abstract dynamics developed in Sections 5–7 admit a concrete computational realization.

□

REFERENCES

[1] C. E. Shannon, "A Mathematical Theory of Communication," *Bell System Technical Journal*, 27(3):379–423, 1948.

[2] E. T. Jaynes, "Information Theory and Statistical Mechanics," *Physical Review*, 106(4):620–630, 1957.

[3] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd ed. Wiley-Interscience, 2006.

[4] A. N. Kolmogorov, "Three Approaches to the Quantitative Definition of Information," *Problems of Information Transmission*, 1(1):1–7, 1965.

[5] P. Grünwald, *The Minimum Description Length Principle*. MIT Press, 2007.

[6] D. J. C. MacKay, *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.

[7] N. Tishby, F. C. Pereira, and W. Bialek, "The Information Bottleneck Method," Allerton Conference, 1999.

[8] S. Amari and H. Nagaoka, *Methods of Information Geometry*. AMS, 2000.

[9] M. J. Wainwright and M. I. Jordan, "Graphical Models, Exponential Families, and Variational Inference," *FnT ML*, 2008.

[10] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[11] J. Pearl, *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, 1988.

[12] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning*. Cambridge University Press, 2014.

[13] M. Raginsky and I. Sason, "Concentration of Measure Inequalities in Information Theory," *FnT Communications*, 2014.

[14] R. T. Rockafellar, *Convex Analysis*. Princeton University Press, 1970.

[15] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[16] D. P. Bertsekas, *Nonlinear Programming*, 2nd ed. Athena Scientific, 1999.

[17] C. Villani, *Optimal Transport: Old and New*. Springer, 2009.

[18] L. C. Evans, *Partial Differential Equations*, 2nd ed. AMS, 2010.

[19] H. Risken, *The Fokker-Planck Equation*. Springer, 1989.

[20] R. M. Neal, "Probabilistic Inference Using MCMC," Technical Report, 1993.

[21] W. K. Hastings, "Monte Carlo Sampling Methods Using Markov Chains," *Biometrika*, 1970.

[22] S. Asmussen and P. W. Glynn, *Stochastic Simulation*. Springer, 2007.

[23] L. Lovász, "Random Walks on Graphs: A Survey," 1993.

[24] M. E. J. Newman, *Networks: An Introduction*. Oxford University Press, 2010.

[25] C. R. Shalizi, "Advanced Data Analysis from an Elementary Point of View," 2013 (draft).

[26] L. Cronin, S. I. Walker, et al., "Assembly Theory Explains and Quantifies Selection and Evolution," *Nature*, 2021.

[27] L. Cronin, "Assembly Theory," University of Glasgow, 2017.

[28] S. A. Benner, "On the Limitations of Assembly Theory," 2023.

[29] C. Marshall et al., "Assembly Theory and Algorithmic Information: A Critical Analysis," *Journal of Molecular Evolution*, 2024.

[30] A. Group et al., "Abiotic Generation of High Assembly Index Structures," *J. Royal Society Interface*, 2024.