Constraint Before Completion

A Trajectory-First Theory of Work

Flyxion

March 31, 2026

Abstract

Completion is a phase transition in the constraint space of a system, characterized by convergence to a minimum of a Lyapunov functional over its admissible configurations. What is commonly perceived as "finishing"—the accumulation of polish, detail, and surface resolution—is merely the projection of an already-solved structure into an observer-accessible interface. This essay argues that work ends when entropy over structurally relevant degrees of freedom is exhausted, and that what follows is mechanical projection: deterministic, delegatable, and epistemically redundant for the agent who holds the constraint structure. Across three domains—surface painting, generative scripting, and live performance—the locus of completion migrates from artifact to generator to trajectory, revealing a unified principle. The central distinction is between internal completion, in which the constraint set has collapsed the solution manifold to a single equivalence class, and external completion, in which that manifold has been projected into perceptual space for an observer. These two notions of completion do not coincide, and much apparent disagreement about whether a work is "done" reduces to a failure to specify which is at stake. A formal treatment of constraint closure, entropy reduction, and temporal legibility is developed, and three worked examples—a mural, a Blender script, and a live painting set—are analyzed as concrete instantiations of the same underlying structure.

Contents

1 Introduction: The Ontology of the Threshold **2**

2 The Geometry of the Solution Space **3**

2.1 From Solution Entropy to Lyapunov Energy 4

3 Internal versus External Completion **5**

4 Headless Execution and the Decoupling of the Interface **6**

5 The Script as Trajectory: Event-History Ontology **7**

6 Entropy Collapse Across Representational Layers **8**

7 Constraint Absence and Semantic Collapse **10**
7.1 Collapse as the Dual of Closure . . . 10
7.2 Embedding Spaces and Neighborhood Semantics . . . 10
7.3 Entropy as a Measure of Constraint Failure . . . 10
7.4 Triplet Violations as Local Collapse Events . . . 11
7.5 Operator Blindness as Constraint Deficiency . . . 11
7.6 Proofing Kernels as Closure Operators . . . 12
7.7 Collapse and Completion as Dual Phenomena . . . 12

8 Semantic Structure as Local Field Geometry **12**
8.1 From Global Geometry to Local Semantic Fields . . . 13
8.2 Semantic Shift as Entropic Flow . . . 13
8.3 Semantic Shift as an Entropy Functional . . . 13
8.4 Hierarchical Semantics and Constraint Propagation . . . 14
8.5 Field Interpretation and RSVP Correspondence . . . 15
8.6 One Swallow Does Not Make a Summer . . . 15

9 Realization Geometry and Semantic Quotients **15**
9.1 The Realization Map . . . 15
9.2 Semantic Fields as Approximate Fibers . . . 16
9.3 Semantic Shift as Motion in the Quotient . . . 16
9.4 Collapse versus Invariance-Preserving Abstraction . . . 16
9.5 Completion as Quotient Stability . . . 17

10 Functorial Structure of the Projection Hierarchy **17**
10.1 The Trajectory Category . . . 17
10.2 Semantic Fields as a Functorial Image . . . 18
10.3 Collapse as Non-Faithfulness . . . 18
10.4 Completion as Functor Stabilization . . . 18
10.5 From Objects to Histories . . . 18

11 RSVP Dynamics as Infinitesimal Generator **19**
11.1 From Static Functor to Dynamical Generator . . . 19
11.2 Completion as a Fixed Point of the Flow . . . 19
11.3 Collapse as Instability . . . 19
11.4 Lyapunov Structure of Completion . . . 20

12 Temporal Constraints and the Management of Entropy **20**

13 The Engineering of Legibility **21**

14 The Denouement Operator **22**

15 Music as External Clock **24**

16 Epistemic Asymmetry and Collaboration Friction **25**

17 The Relocation of Rendering **26**

18 Three Modes of Completion **27**

19 Implications **28**

20 Objections and Formal Responses **29**
20.1 Premature Termination . . . 29
20.2 Material Contingency . . . 29
20.3 Independence of the Three Characterizations . . . 30
20.4 Toward a General Theory of Constraint-Driven Systems . . . 30

21 Conclusion: Completion as a Property of Mappings **31**

A Constraint Closure and Solution Manifolds **32**
A.1 State Space and Constraints . . . 32
A.2 Structural Equivalence . . . 32
A.3 Formal Definition . . . 32
A.4 Entropy of the Solution Set . . . 32
A.5 Partial Order on Constraint Density . . . 32

B Projection and External Completion **32**
B.1 The Projection Map . . . 32
B.2 Non-Invariance Under Structural Equivalence . . . 33
B.3 Residual Encoding Distance . . . 33

C Generative Mappings and Headless Closure **33**
C.1 Generator Space . . . 33
C.2 Headless Execution . . . 33
C.3 Hierarchy of Generator Spaces . . . 34

D Temporal Legibility and Information Release **34**
D.1 Legibility as Cumulative Recognition . . . 34
D.2 Observer Entropy . . . 34
D.3 Denouement Threshold . . . 34

E Worked Example: Mural as Manifold Collapse **34**

F Worked Example: bpy Script as Generator **35**

G Worked Example: Live Painting as Controlled Trajectory **35**

H Semantic Collapse and Fiber-Quotient Correspondence **36**
H.1 Metric Measure Space Framework . . . 36
H.2 Variational Characterization . . . 37
H.3 Fiber-Quotient Correspondence . . . 37

I Functorial Projection and Lyapunov Stability **37**
I.1 Semantic Projection Systems 37
I.2 Generator–Functor Correspondence 38
I.3 Proof of Lyapunov Decrease 38

1. Introduction: The Ontology of the Threshold

What does it mean for a work to be finished? The question appears deceptively simple. In practice, however, disagreements about completion are among the most persistent and structurally intractable in any collaborative creative or technical enterprise. A programmer declares a system complete when its specification is fully implemented; a designer says it is nowhere near done until the interface is polished. A sculptor considers a piece finished when the form is established; a gallery assistant disagrees until it is mounted, lit, and titled. These disagreements do not resolve through negotiation about taste. They reflect a prior divergence in what kind of object "completion" is taken to be a property of.

The central thesis of this essay is that completion is a threshold event in the constraint manifold of a system, not a gradual accumulation of surface resolution. It is not the final stroke that finishes a painting; the painting is finished at the moment when the remaining degrees of freedom have been rendered structurally irrelevant. Everything that follows—highlights, texture, citation integration, rendering passes, prose smoothing—is a traversal of an already-defined space. The traversal may be necessary for certain audiences, but it is not the work. It is the projection of the work.

This claim can be stated geometrically. Let Ω be the configuration space of all possible states of a system—all possible paintings, all possible meshes, all possible arguments. A set of constraints $\mathscr{C}$ restricts the admissible states to a feasible manifold $\Omega_{\mathscr{C}} \subset \Omega$. Within this feasible set, an equivalence relation identifies configurations that are structurally indistinguishable under the relevant identity-preserving observables. The quotient $\mathscr{M} = \Omega_{\mathscr{C}}/\sim$ is the solution manifold. Completion, properly understood, occurs when this quotient has collapsed to a single equivalence class: when $|\mathscr{M}| = 1$ and the effective entropy $S_{\text{eff}} = \log|\mathscr{M}|$ has gone to zero. At that point, the constraint system has done its work. Any remaining activity operates within one class and cannot alter the structural identity of the result.

This is what might be called the coloring-book phenomenon: a work that is structurally complete but perceptually unfinished. The coloring book is a useful image because it makes visible what is usually invisible—the constraint surface that has already been established. The outlines fix the lighting, the composition, the identity, and the spatial relations. Everything within those outlines is a fill, not a determination. The person who colors the book is not solving the problem; they are projecting a solved structure into a medium. Their choices—texture, value, saturation—exist within the equivalence class already established by whoever drew the outlines.

The disagreement that typically arises around works in this state is between an agent who has access to the constraint structure and an observer who only sees the projection. The agent, operating from inside the constraint system, perceives the solution as already closed and experiences any remaining work as low-entropy execution. The observer, lacking access to the constraints, sees only the surface, and surface-incompleteness is indistinguishable from structural-incompleteness from their vantage point. This asymmetry is not a matter of preference or rigor; it is a structural feature of the epistemic situation.

The essay proceeds as follows. Section 2 develops the geometric formalism of constraint closure and solution manifolds. Section 3 draws the distinction between internal and external completion precisely and analyzes the projection map that connects them. Section 4 examines what happens when the locus of completion migrates from the object to the generator: the case of headless scripting and the elimination of the observation loop. Section 5 connects this to an event-history ontology in which identity is a property of trajectories rather than states. Section 6 unifies the three representational layers—perceptual, spatial, generative—as a sequence of entropy collapses. Section 12 introduces the inversion that occurs when time becomes a hard constraint, as in live performance. Section 13 formalizes the engineering of legibility as a controlled entropy-release problem. Section 14 analyzes the denouement operator: the phase transition in observer inference at which structural closure becomes externally visible. Section 15 treats music as an external clock that offloads pacing decisions from the performer. Section 16 unifies the treatment of epistemic asymmetry and collaboration friction, showing both as instances of agents operating at different loci of closure. Section 17 corrects a potential misreading by showing that rendering is never eliminated but only relocated. Section 18 synthesizes three modes of completion—generator, delegated, and performed. Section 20 addresses the principal formal objections to the framework and consolidates the three-level characterization of completion. Section 19 draws out implications for software engineering, artificial intelligence, and pedagogy. Section 21 closes by arguing that completion is a property of mappings, not of objects.

2. The Geometry of the Solution Space

Let Ω be a configuration space. In the most general setting this may be taken as any measurable space, though for the purposes of this essay finite or countably generated spaces suffice. A system in state $x \in \Omega$ is subject to a collection of constraints $\mathscr{C} = \{C_i\}_{i=1}^{k}$, where each $C_i : \Omega \to \{0,1\}$ is an admissibility indicator. The feasible set is

$$\Omega_{\mathscr{C}} = \{x \in \Omega \mid C_i(x) = 1 \text{ for all } i\}.$$

The constraints define the problem. They encode everything that is fixed by intent: the subject of a painting, the functional specification of a program, the argument of an essay. Different practitioners may differ in how many constraints they have imposed—a collaborator who has agreed on subject matter but not composition has fewer constraints operative than one who has settled both—and this difference in constraint density accounts for much of the apparent disagreement about whether a given state constitutes progress.

Within $\Omega_{\mathscr{C}}$, define an equivalence relation by structural identity. Two configurations $x, y \in \Omega_{\mathscr{C}}$ are structurally equivalent, written $x \sim y$, if and only if they are indistinguishable under all identity-preserving observables. The precise content of "identity-preserving" depends on the domain: for paintings, it means configurations that share the same subject, composition, and lighting model; for programs, configurations that implement the same functional behavior; for arguments, configurations that establish the same logical structure. The quotient

$$\mathscr{M} = \Omega_{\mathscr{C}} / \sim$$

is the solution manifold. Each element of $\mathscr{M}$ is an equivalence class of configurations that are, for the purposes of the work, the same thing.

The effective entropy of the solution space is

$$S_{\text{eff}} = \log|\mathscr{M}|.$$

This measures the remaining structural ambiguity. A large S_{eff} means many structurally distinct solutions are still admissible; the work is genuinely underdetermined. A small S_{eff} means the constraints have collapsed the space to a near-unique structure. When $|\mathcal{M}| = 1$, the entropy is zero and the structure is fully determined.

Definition 2.1 (Constraint Closure)**.** A system is *structurally closed* if $|\mathcal{M}| = 1$, equivalently if $S_{\text{eff}} = 0$. All admissible configurations lie within a single equivalence class, and no further structural distinction is possible.

Proposition 2.2. *Constraint closure is equivalent to the exhaustion of structural entropy.*

Proof. If $|\mathcal{M}| = 1$, then for any $x, y \in \Omega_{\mathcal{C}}$ we have $x \sim y$. No configuration in the feasible set carries a structural distinction from any other. Therefore $S_{\text{eff}} = \log 1 = 0$. Conversely, if $S_{\text{eff}} = 0$ then $|\mathcal{M}| = 1$ by definition. □

The claim is not that constraint closure always precedes execution in practice, but that when it does, everything that follows is a projection rather than a determination. A painter who has fixed the subject, lighting model, compositional flow, and character identity has driven the solution manifold to a single equivalence class. The remaining operations—highlights, shadows, texture, surface variation—exist within that class and do not alter the structural identity of the result. They are choices, but they are choices within a determined space. The class contains many realizations, but they are all the same painting in the relevant sense.

This is why the coloring-book image is more than a metaphor. The outlines of a coloring book are a realization of the constraint set $\mathcal{C}$ in perceptual space. They make visible the boundary of the feasible set and, more importantly, the quotient structure: the fact that very many distinct colorings will produce what is recognizably the same composition. The act of coloring is a traversal within a single equivalence class. No coloring can change what the drawing is.

2.1. From Solution Entropy to Lyapunov Energy

The entropy $S_{\text{eff}} = \log|\mathcal{M}|$ defined above measures the residual ambiguity of the solution space at the level of equivalence classes. This quantity admits a dynamical refinement that connects the discrete combinatorial structure of the quotient to the continuous geometry of field evolution.

Let $\Phi(x, t)$ denote a continuous representation of the system over a configuration or embedding space. The evolution of the system under constraint propagation can be expressed as a flow

$$\partial_t \Phi = \mathcal{L}[\Phi].$$

Define the energy functional

$$\mathcal{E}[\Phi] = \int \Phi \log \Phi \, dx + \alpha \int \|\nabla \Phi\|^2 dx + \beta \mathcal{J}[\Phi],$$

where the three terms penalize dispersion, geometric instability, and semantic incoherence respectively. This functional refines S_{eff} in the sense that S_{eff} measures discrete structural ambiguity while $\mathcal{E}[\Phi]$ measures continuous geometric and variational ambiguity. The two descriptions are related by a precise correspondence:

$$S_{\text{eff}} = 0 \iff \mathcal{E} \text{ minimized.}$$

Constraint closure in the discrete setting corresponds to convergence to a minimum of $\mathcal{E}$ in the continuous setting. The two are not different theories but different resolutions of the same phenomenon: entropy collapse describes the reduction of admissible equivalence classes, while Lyapunov descent describes the path by which that collapse occurs. Together they give the theory its vertical structure—the same fixed point is visible at the level of the quotient, at the level of the generator, and at the level of the field dynamics—and completion becomes provably stable rather than merely asserted.

3. Internal versus External Completion

Structural closure in $\mathscr{M}$ does not imply that a work is finished in any observable sense. The solution manifold may have collapsed to a single class while the representative element in Ω—the actual painting, the actual document, the actual mesh—remains far from what any observer would accept as complete. The gap between these two states is the central source of disagreement about completion, and it must be made precise.

Let $\pi : \Omega \to \mathscr{Y}$ be a projection map from configuration space into observable space. The space $\mathscr{Y}$ represents what can be directly perceived or accessed by an observer who lacks the constraint structure. This might be the visible surface of a canvas, the rendered output of a program, the formatted pages of an essay. The observer's judgment of completion is a function of $\pi(x)$, not of x itself.

Definition 3.1 (Internal Completion)**.** A configuration $x \in \Omega_{\mathscr{C}}$ is *internally complete* if the system is structurally closed: $|\mathscr{M}| = 1$.

Definition 3.2 (External Completion)**.** A configuration $x \in \Omega_{\mathscr{C}}$ is *externally complete* if $\pi(x)$ satisfies the observer's recognition criteria: the projected image is sufficient for the observer to reconstruct the constraint structure without additional information.

Proposition 3.3. *Internal and external completion are independent. A system may be internally complete while externally incomplete, and externally complete while internally incomplete.*

Proof. For the first case, let $x, y \in \Omega_{\mathscr{C}}$ with $x \sim y$ but $\pi(x) \neq \pi(y)$. The system is structurally closed—there is one equivalence class—but the projection distinguishes the two configurations, meaning one may satisfy the observer's criteria while the other does not. For the second case, consider a work that has been polished to a high surface finish without ever resolving its underlying constraints: $\pi(x)$ satisfies external criteria, but $|\mathscr{M}| > 1$ because the structural questions remain open. Both directions are achieved in practice. □

The coloring-book phenomenon is the first case: structural closure with external incompleteness. The inverse—polished surfaces over unresolved constraints—is at least as common and arguably more dangerous. A text that is smoothly written but structurally incoherent, a program that is beautifully documented but architecturally confused, a design that is visually refined but functionally underdetermined: all of these are externally complete in the superficial sense while internally open. They tend to fail at precisely the moment their constraints are tested.

The gap between internal and external completion can be quantified as the residual encoding distance. Define

$$\Delta(x) = d_{\mathscr{Y}}(\pi(x), \tau)$$

where $\tau \subset \mathscr{Y}$ is the observer's recognition threshold—the set of projected configurations that the observer accepts as complete—and $d_{\mathscr{Y}}$ is a metric on observable space. The coloring-book problem is the regime in which $|\mathscr{M}| = 1$ but $\Delta(x) > 0$. Closing Δ is the work of rendering: translating internal constraint clarity into external legibility.

What this formalization makes clear is that rendering is not a creative act in the same sense as constraint determination. It does not change $\mathscr{M}$; it changes $\pi(x)$ within the single equivalence class already established. The degrees of freedom available to rendering are structurally irrelevant: any choice within the class produces the same work in the sense that matters for the argument, the composition, or the geometry. What rendering produces is not structural but communicative: it reduces Δ so that an observer can access the constraint structure without reconstructing it from scratch.

This distinction has direct practical implications. The question of whether to render—whether to integrate the citations, smooth the prose, add the highlights, export the mesh—is not a question about whether the work is done. It is a question about for whom it needs to be legible and whether the residual encoding distance matters for that audience. An essay shared among collaborators who share the constraint vocabulary may be complete at $\Delta > 0$; the same essay submitted to a journal requires $\Delta = 0$ by institutional convention.

4. Headless Execution and the Decoupling of the Interface

The preceding analysis concerned the completion of artifacts: paintings, documents, physical objects. The argument can be extended upward. Just as the painting is internally complete when the constraint manifold collapses, a more abstract system is internally complete when the generator that produces the painting is fixed. At each level, the objects of the level below are projections of the structure above.

In the domain of three-dimensional modeling, this migration is particularly concrete. A mesh—a triangulated surface in three-dimensional space—is a configuration in some space Ω_{mesh}. Like a painting, it can be internally complete (structurally determined) while remaining externally incomplete (unrendered). But the mesh itself is a projection from a higher-level space: the space of generative procedures, scripts, or programs that produce meshes. Let Θ be this generator space, and let

$$\Phi : \Theta \to \Omega_{\text{mesh}}$$

be the realization map that takes a script to the mesh it produces. Completion now migrates to Θ: a script $\theta \in \Theta$ is structurally closed if $\Phi(\theta) \in \Omega_{\mathscr{C}}$ and all nearby scripts produce structurally equivalent meshes.

Definition 4.1 (Generator Closure)**.** A parameter $\theta \in \Theta$ is *generator-closed* if (i) $\Phi(\theta) \in \Omega_{\mathscr{C}}$, and (ii) for all θ' in a neighborhood of θ, $\Phi(\theta') \sim \Phi(\theta)$.

Condition (i) says the script produces an admissible configuration. Condition (ii) says the mapping is stable: small changes to the script do not alter the structural class of the output. A script that is generator-closed is, in the relevant sense, a fixed point of the generative process.

Proposition 4.2. *If a script θ is generator-closed, then the evaluation $\pi \circ \Phi(\theta)$ is not required to verify structural correctness.*

Proof. Structural correctness is defined by membership in $\Omega_{\mathscr{C}}/\sim$, which is a property of $\Phi(\theta)$ in $\mathscr{M}$, not of $\pi(\Phi(\theta))$ in $\mathscr{Y}$. Since the projection π does not alter membership in $\mathscr{M}$, evaluating it adds no information about structural correctness. Observation of the rendered output is therefore epistemically redundant given generator closure. □

This proposition is the theoretical grounding for headless execution. Operating a modeling environment without a display, executing scripts on a remote machine with no graphical output, running a rendering pipeline without viewing the result: these are not mere workflow optimizations. They are the natural consequence of locating completion at the generator level. The GUI is an observer-layer projection: it makes the constraint structure visible to a human eye. But if the constraint structure has already been verified at the script level—if the script is known to be generator-closed—then the visual feedback loop is redundant. It resolves uncertainty that is no longer present.

There is a further implication. The migration of completion from mesh to script changes what kind of object the "finished work" is. A mesh is stateful: it is a particular configuration of vertices, edges, and faces, and it can be edited, deformed, or interpreted. A script is trajectory-defining: it specifies the sequence of operations that produce the configuration. The mesh answers the question "what is this?" while the script answers "how does this come to be?" Once the latter is fixed and verified as generator-closed, the former is no longer a primary object—it is an evaluation of the script under a particular set of environmental parameters.

This distinction between stateful objects and trajectory-defining operators recurs throughout the analysis and will be taken up in detail in Section 5. For now, it is enough to note that the headless workflow represents a final step in the decoupling of intent from observation. The system is complete at the level of the operator; the projection into observable space is available on demand but is not constitutive of the work.

5. The Script as Trajectory: Event-History Ontology

The distinction between stateful objects and trajectory-defining operators corresponds to a deeper ontological distinction: between identity as configuration and identity as history. This distinction is not merely descriptive; it has consequences for what counts as the same work across time, what it means to modify a work, and where the relevant equivalences lie.

Define a history at time t as the ordered sequence of events

$$H_t = (e_1, e_2, \ldots, e_t),$$

where each e_i is an operation or transformation applied to the system. The configuration at time t is the evaluation of this history:

$$x(t) = \mathrm{Eval}(H_t).$$

An object, in this ontology, is not a state but an evaluation. Its identity is not its current configuration but its generative history. Two objects are the same not when they are in the same state—since any state can be reached by many histories—but when their histories, at the relevant level of description, are equivalent.

Definition 5.1 (Trajectory Identity)**.** Two objects x and y are *trajectory-identical* if there exists a history equivalence $H_t \equiv H'_t$ such that $\text{Eval}(H_t) \sim \text{Eval}(H'_t)$ for all t.

This definition subsumes structural identity as a special case: if two objects have the same constraint structure and the same generative history, they are the same work. But it also handles cases where the constraint structure is established incrementally: two drafts of an essay may be structurally identical even if their surface texts differ, provided they are the result of equivalent constraint-imposing operations.

The script is, in the technical sense, an explicit history generator. A Blender Python script encodes the sequence of operations—create mesh, extrude faces, set material, apply modifier—that produce a three-dimensional object. It is, structurally, an append-only log of the operations that constitute the object's identity. The mesh is a terminal snapshot of this log under evaluation; the script is the log itself. In the event-history ontology, the script is more fundamental than the mesh: it is the identity of the object, not merely a way of producing it.

Proposition 5.2. *In the event-history ontology, the script is a more complete representation of the work than the mesh it generates.*

Proof. The mesh $x = \text{Eval}(H_T)$ contains the configuration at the terminal time T but does not encode the history H_T that produced it. Two meshes with the same geometry may have been produced by entirely different histories—one by extrusion, one by subdivision surface, one by sculpting—and these differences are lost in the static representation. The script preserves the full history and therefore contains strictly more information about the identity of the work than the mesh alone. □

This has an immediate consequence for the notion of completion. If identity is constituted by history rather than state, then the work is complete when the history is closed: when the append-only log has received its final entry and no further operations are required for structural closure. The mesh is just a visualization of a closed history. Rendering it is not completing the work; it is making the closed history visible.

The deeper implication is that modification—the act of going back and changing something—is not a continuation of the history but a branching of it. In a stateful-object ontology, editing the mesh is an update to a single object. In the event-history ontology, editing produces a new history that diverges from the original at the point of intervention. The two histories are different works, even if their terminal evaluations happen to be visually similar. This is why revision feels categorically different from rendering: revision modifies the identity of the work, while rendering only makes the identity visible.

6. Entropy Collapse Across Representational Layers

The analysis so far has treated completion at individual levels: the surface of a painting, the mesh of a three-dimensional object, the script that generates it. But these levels form a hierarchy, and the most important claim is not about any single level but about the structure of the hierarchy itself. Completion migrates upward across layers, and at each migration it carries the character of an entropy collapse.

Consider the sequence of representational layers:

$$\Theta \xrightarrow{\Phi} \Omega_{\text{mesh}} \xrightarrow{\rho} \Omega_{\text{render}} \xrightarrow{\pi} \mathscr{Y}$$

Here Θ is the script space, Ω_{mesh} is the mesh configuration space, Ω_{render} is the space of rendered images, and $\mathscr{Y}$ is the observer's perceptual space. Each arrow is a projection that eliminates some degrees of freedom and preserves others. Moving left increases abstraction and constraint density; moving right increases perceptual accessibility and decreases formal specificity.

The entropy at each layer is defined by the effective size of the solution manifold at that level. Define:

$$\begin{aligned} S_{\text{generative}} &= \log|\mathscr{M}_\Theta|, \\ S_{\text{spatial}} &= \log|\mathscr{M}_{\text{mesh}}|, \\ S_{\text{perceptual}} &= \log|\mathscr{M}_{\mathscr{Y}}|. \end{aligned}$$

Proposition 6.1. *In the regime where constraints are imposed top-down,*

$$S_{generative} \leq S_{spatial} \leq S_{perceptual}.$$

Proof. Each projection map is surjective on its image but not injective: many scripts produce the same mesh, many meshes produce the same rendered image, many rendered images produce the same perceptual experience. Each projection therefore maps multiple distinct solutions to the same equivalence class at the lower level, increasing apparent degeneracy. The entropy at each lower level reflects the additional ambiguity introduced by the loss of information under projection. □

The intuitive content of this proposition is that working at a higher level of abstraction resolves more degrees of freedom per unit of cognitive effort. A decision made at the script level—say, the choice of a symmetry-preserving transformation—simultaneously determines entire families of mesh configurations and rendered images. A decision made at the perceptual level—say, the choice of a particular highlight—determines only that highlight. This asymmetry explains why agents who work top-down can achieve structural closure rapidly while those who work bottom-up spend most of their effort on choices that carry little structural weight.

The sequence of entropy collapses can now be understood as follows. When a practitioner moves from painting to modeling to scripting, they are not merely changing tools. They are shifting the level at which constraints are imposed and at which structural closure is declared. Each shift upward collapses a layer of entropy that was previously left open. The mural painter who establishes lighting direction and compositional flow is collapsing perceptual entropy. The modeler who specifies topology and proportions is collapsing spatial entropy. The scripter who encodes the transformation as a deterministic procedure is collapsing generative entropy. At each level, "done" migrates upward to the point where the remaining variability is judged structurally trivial.

Definition 6.2 (Entropy Collapse Sequence)**.** An entropy collapse sequence is a sequence of constraint-imposing operations across representational layers such that $S_k \to 0$ at layer k before S_{k-1} is addressed.

The stopping rule, in this vocabulary, is: halt at the highest layer where $S_k = 0$. Everything below that layer is a projection of a solved structure and does not require the attention of the agent who imposed the constraints.

7. Constraint Absence and Semantic Collapse

7.1. Collapse as the Dual of Closure

We have defined completion as the exhaustion of structurally relevant degrees of freedom: the solution manifold collapses to a single equivalence class, and what follows is mere projection. We now examine the complementary phenomenon. Collapse occurs not when too many constraints are applied but when too few are, allowing incompatible regions of the configuration space to overlap in ways that destroy the distinctions the system was meant to preserve.

Let Ω be a representational space and $\mathscr{C}$ a constraint set. Constraint closure reduces Ω to a structured manifold $\mathscr{M}$ whose equivalence classes are internally coherent. Collapse is the failure mode in which elements that are not equivalent under $\sim$ become indistinguishable in the representation.

Definition 7.1 (Semantic Collapse)**.** A system exhibits *semantic collapse* when elements that are structurally inequivalent become geometrically indistinguishable:

$$x \not\sim y \quad \text{but} \quad d(x, y) \approx 0.$$

Closure and collapse are therefore dual outcomes of constraint density. In the closure regime, the constraint set is sufficiently rich that $|\mathscr{M}| = 1$: the effective entropy S_{eff} has been driven to zero. In the collapse regime, the constraint set is insufficient: regions of Ω that should remain separated are instead merged, and the effective entropy rises *across* partitions that ought to be maintained. The problem of work—in any domain—is the controlled reduction of entropy without destroying essential distinctions.

7.2. Embedding Spaces and Neighborhood Semantics

Modern language models represent meaning in high-dimensional embedding spaces, where proximity serves as a proxy for semantic similarity. However, this geometry lacks the structural constraints required to preserve logical distinctions. As argued in [5], embedding neighborhoods function analogously to plausibility models, but without accessibility relations or operator-sensitive structure. The result is a systematic drift in which modal, epistemic, and indexical distinctions are flattened into proximity-based similarity.

This maps precisely onto the collapse definition above. The embedding space $\mathscr{E}$ is a representation of Ω, and the projection $\pi : \Omega \to \mathscr{E}$ is meant to preserve the equivalence structure. When it fails to do so—when the projection identifies x and y despite $x \not\sim y$—the embedding exhibits semantic collapse. The neighborhood of a token is a local region of $\mathscr{E}$, and if that neighborhood contains elements that are not semantically equivalent, the region is not an equivalence class but an intersection of incompatible classes.

7.3. Entropy as a Measure of Constraint Failure

Where constraint closure corresponds to entropy reduction within the solution manifold, collapse corresponds to entropy increase across constraint-defined partitions. Let N_t denote the neighborhood of a token t in embedding space, and define the local entropy

$$H(t) = -\sum_{u \in N_t} p(u \mid t) \log p(u \mid t).$$

High entropy in $H(t)$ indicates that the neighborhood is dispersed across semantically incompatible roles: modal and factual, indexical and descriptive, necessary and contingent. This is not richness but incoherence—the structural signature of a constraint-deficient representation.

Proposition 7.2. *Semantic collapse corresponds to a local increase in entropy across constraint-defined partitions of the embedding space.*

Proof. Under a structurally coherent representation, each equivalence class under $\sim$ maps to a distinct region of $\mathscr{E}$, and neighborhoods respect these boundaries: N_t contains only elements equivalent to t. In the collapse regime, N_t contains elements from multiple equivalence classes. The conditional distribution $p(u \mid t)$ is therefore supported across classes, yielding $H(t) > 0$ even when the structural entropy of the class containing t is zero. The increase in $H(t)$ directly measures the degree of class boundary violation. □

7.4. Triplet Violations as Local Collapse Events

The standard diagnostic for embedding coherence is the triplet test. Let (A, S, D) be an anchor, a valid substitute, and a decoy. A coherent embedding satisfies

$$\text{sim}(A, S) > \text{sim}(A, D).$$

Collapse occurs when

$$\text{sim}(A, D) \geq \text{sim}(A, S).$$

Within the present framework, a triplet violation is not merely a metric failure; it is a violation of the intended equivalence relation on Ω. The decoy D has intruded into the equivalence class of the anchor A: it occupies a neighborhood region that structurally belongs to a different class. The collapse is local—it affects the specific region around A—but it is symptomatic of a global deficiency in the constraint structure of the embedding.

7.5. Operator Blindness as Constraint Deficiency

The most revealing form of semantic collapse in language model embeddings is what can be called operator blindness: the failure to enforce separation between logically distinct operator classes. Modal logic introduces accessibility relations that separate necessity from actuality. Epistemic logic introduces agent-indexed knowledge that separates belief from truth. Indexical logic introduces contextual anchoring that separates the referent of "I" from any descriptive characterization of it.

Embedding systems lack these structures as first-class constraints. Their proximity geometry is trained on distributional co-occurrence, which approximates topical and syntactic similarity but is structurally indifferent to operator-level distinctions. The result is that modal, epistemic, and indexical contrasts are represented only to the extent that they happen to be recoverable from surface co-occurrence patterns—which is to say, unreliably and without structural guarantee.

Collapse is therefore not error in the ordinary sense. It is under-constrained geometry: the configuration space Ω has not been equipped with the constraints that would enforce the necessary separations, and the projection π consequently fails to preserve them.

7.6. Proofing Kernels as Closure Operators

Constraint-enforcing architectures—what [5] terms modal proofing kernels—can be interpreted within the present framework as mechanisms for restoring closure to an under-constrained representation. They act by enforcing separation between operator classes, constraining neighborhood graphs to satisfy structural axioms, and projecting representations into typed subspaces.

In the language of the present theory, such an architecture converts an under-constrained embedding space into a structured manifold satisfying $\mathscr{C}$. It is not a post-hoc correction but a constraint-imposition mechanism: it adds the missing elements of $\mathscr{C}$ to the system, driving $|\mathscr{M}|$ toward the regime in which equivalence classes are internally coherent and externally separated.

7.7. Collapse and Completion as Dual Phenomena

The relationship between the two phenomena can now be stated precisely:

$$\begin{aligned} \text{Completion} &\iff S_{\text{eff}} \to 0 \quad \text{within the solution manifold,} \\ \text{Collapse} &\iff S_{\text{eff}} \uparrow \quad \text{across constraint boundaries.} \end{aligned}$$

Completion reduces entropy by narrowing the solution space to a single well-defined class. Collapse increases entropy by dissolving the boundaries that define classes in the first place. The two processes are not merely different but opposed: completion is the regime in which constraints do their work, and collapse is the regime in which they fail to.

This dual structure gives the theory its negative half. Without a failure mode, the notion of constraint closure remains a positive aspiration rather than a discriminating criterion. With it, the theory acquires diagnostic power: a system can be evaluated not only by how close it is to internal completion but also by how far it is from collapse.

8. Semantic Structure as Local Field Geometry

The analysis of embedding spaces has traditionally oscillated between two extremes: global geometric descriptions that sacrifice semantic fidelity, and local probing techniques that fail to scale. Recent work by [16] introduces a framework that resolves this tension by identifying a middle layer of structure—semantic field subspaces (SFSes)—which function as localized, geometry-preserving neighborhoods within embedding space.

This result can be reinterpreted within the present framework as a formal confirmation of a deeper claim: semantic structure is not globally uniform, but emerges as a stratified field defined over local regions of the embedding manifold.

8.1. From Global Geometry to Local Semantic Fields

Embedding spaces encode meaning by mapping semantic similarity to geometric proximity. However, this mapping is not globally stable. High-dimensional embeddings exhibit anisotropy, clustering artifacts, and variance concentration, making global geometric interpretation unreliable.

The SFS construction addresses this by restricting attention to local neighborhoods that preserve semantic coherence. Formally, an embedding space $\mathscr{E} \subset \mathbb{R}^d$ is decomposed into a family of subspaces

$$\mathscr{E} \supset \bigcup_i \mathscr{S}_i,$$

where each $\mathscr{S}_i$ corresponds to a semantic field subspace capturing a coherent local region of meaning. This decomposition replaces the notion of a single global semantic manifold with a field of local semantic charts, each of which admits a more stable interpretation.

Within the present ontology, this corresponds to a refinement of the solution space: rather than a single manifold collapsing under constraint, the system exhibits a stratified structure in which multiple local manifolds coexist and interact. Each local manifold is an approximation to a region of $\mathscr{M}$ that is internally coherent under a local version of the equivalence relation $\sim$.

8.2. Semantic Shift as Entropic Flow

A key contribution of [16] is the introduction of Semantic Shift: a metric that quantifies how the semantic content of a subspace evolves under clustering and aggregation. Empirically, Semantic Shift tracks the transition from fine-grained clusters to higher-level abstractions, revealing a hierarchy of semantic organization in which clusters of specific entities merge into broader categories as the scale increases.

This process can be interpreted as an entropic flow:

$$\mathscr{S}_{\text{fine}} \longrightarrow \mathscr{S}_{\text{coarse}},$$

where increasing abstraction corresponds to the loss of local distinctions and the emergence of higher-order invariants. There is, however, a crucial distinction from the pathological collapse analyzed in the preceding section. Semantic Shift is a *structured contraction*: it reduces distinctions in a way that preserves hierarchical meaning, rather than dissolving boundaries indiscriminately. The difference lies in whether the contraction respects the constraint structure of the semantic field or ignores it.

Not all entropy reduction is therefore collapse-avoidant, and not all entropy increase is collapse. The relevant question is whether the change in entropy respects the intended partitioning of Ω into equivalence classes. Structured contraction—good collapse—removes distinctions that were already redundant at the finer scale. Pathological collapse removes distinctions that were semantically necessary.

8.3. Semantic Shift as an Entropy Functional

We now formalize Semantic Shift as an entropy variation over a family of probability measures induced by embedding space partitions.

Let $(\mathscr{E}, d, \mu)$ be a metric measure space, where $\mathscr{E} \subset \mathbb{R}^d$ is the embedding space, d is the induced Euclidean metric, and μ is a probability measure over $\mathscr{E}$ induced by a corpus. For a clustering scale λ, define a partition $\mathscr{P}_\lambda = \{C_1^\lambda, \dots, C_{k(\lambda)}^\lambda\}$ with corresponding distribution $p_i^\lambda = \mu(C_i^\lambda)$. The entropy at scale λ is

$$S(\lambda) = -\sum_{i=1}^{k(\lambda)} p_i^\lambda \log p_i^\lambda.$$

Semantic Shift between two scales $\lambda_1 < \lambda_2$ is identified with the coupled quantity

$$\Sigma(\lambda_1, \lambda_2) = \Delta S + \alpha \mathscr{D}(\lambda_1, \lambda_2),$$

where $\Delta S = S(\lambda_2) - S(\lambda_1)$ is the entropy change and

$$\mathscr{D}(\lambda_1, \lambda_2) = \sum_i \int_{C_i^{\lambda_1}} \|x - \pi_{\lambda_2}(x)\|^2 \, d\mu(x)$$

is a geometric distortion term measuring how violently the clustering displaces points. A well-formed semantic contraction satisfies $\Delta S < 0$ with $\mathscr{D}$ minimal; pathological collapse corresponds to large $\mathscr{D}$ relative to entropy reduction.

Proposition 8.1. *A semantic field subspace is complete when*

$$\frac{dS}{d\lambda} \approx 0 \quad \text{and} \quad \frac{d\mathscr{D}}{d\lambda} \approx 0,$$

that is, when further resolution changes neither the information content nor the geometry of the field.

Proof. If both derivatives vanish, the partition $\mathscr{P}_\lambda$ is stable under refinement: adding further resolution neither increases information nor displaces representative points. The equivalence classes are therefore fixed, and no new structural distinctions emerge. This is precisely the condition $|\mathscr{M}| = 1$ applied locally within the subspace. □

8.4. Hierarchical Semantics and Constraint Propagation

The SAFARI algorithm of [16] constructs semantic field subspaces through iterative clustering, producing a hierarchy of semantic groupings that evolve over scale. This hierarchy can be understood as a sequence of constraint propagations:

$$\mathscr{C}_0 \subset \mathscr{C}_1 \subset \cdots \subset \mathscr{C}_k,$$

where each level imposes additional constraints that reduce the degrees of freedom of the representation. At early stages, clusters reflect highly specific entities; at later stages, they encode abstract categories. The transition is governed by Semantic Shift, which acts as a measure of how much structure is being compressed at each step.

Within the broader theory of work developed here, this process corresponds to a controlled collapse of the solution space: the system moves from a high-entropy configuration of possibilities toward a lower-dimensional manifold defined by semantic invariants. The practitioner who works top-down—establishing coarse structure before fine detail—is following the same gradient: resolving high-level constraints before low-level ones, ensuring that each level of the hierarchy is closed before descending.

8.5. Field Interpretation and RSVP Correspondence

The SFS framework admits a natural reinterpretation as a field theory over embedding space. Each subspace $\mathscr{S}_i$ can be treated as a local region in which a semantic density Φ, directional flow v, and entropy S are defined. The evolution of semantic structure under clustering then corresponds to a field dynamic:

$$\partial_t \Phi \sim \text{aggregation}, \quad v \sim \text{semantic drift}, \quad S \sim \text{loss of distinction}.$$

Under this interpretation, Semantic Shift becomes a discrete observable of an underlying continuous process: the redistribution of semantic density across the field. This provides a direct bridge to the RSVP framework, in which semantic structure is not a static property of representations but a dynamical feature of a coupled scalar-vector-entropy system. The field variables (Φ, v, S) govern how meaning evolves, how it flows, and how much structural information is preserved or lost at each moment of that flow.

8.6. One Swallow Does Not Make a Summer

The title of [16] captures the central insight: no single embedding, no single neighborhood, and no single cluster suffices to define semantic structure. Meaning emerges only at the level of organized fields and their hierarchical relationships. In the language of this monograph, a single instance is never the work. The work resides in the constraint system that organizes instances into a coherent structure. The SFS framework provides an empirical demonstration of this principle within modern artificial intelligence systems: semantics is not a property of points but of trajectories through structured regions of space, and completion is not the existence of an output but the closure of the generative system that produces it.

9. Realization Geometry and Semantic Quotients

The preceding analysis of semantic field subspaces can be recast in a more fundamental geometric language by introducing a realization map and its associated quotient structure. This move unifies the embedding-space story with the generator-space story developed in Section 4: both are instances of the same underlying operation, viewed from different levels of the representational hierarchy.

9.1. The Realization Map

Let Θ denote a parameter space—model weights, latent generative states, or script configurations—and let

$$\Phi : \Theta \to \mathscr{E}$$

be a realization map that assigns to each parameter configuration a point in embedding space. The key structural feature of Φ is that it is not injective. Multiple parameter configurations realize the same embedding: $\theta_1 \neq \theta_2$ but $\Phi(\theta_1) = \Phi(\theta_2)$. This induces an equivalence relation $\theta_1 \sim \theta_2 \iff \Phi(\theta_1) = \Phi(\theta_2)$ and defines the quotient $\mathscr{E} \cong \Theta/\sim$.

Thus, embedding space is not primary. It is already a quotient of a higher-dimensional generative space, and the structure visible in embeddings is the residue of a constraint system that lives above them.

9.2. Semantic Fields as Approximate Fibers

In this formulation, each embedding $x \in \mathscr{E}$ corresponds to a fiber $\Phi^{-1}(x) \subset \Theta$. However, semantic field subspaces $\mathscr{S}_i \subset \mathscr{E}$ are not single points but regions of approximate invariance. They can therefore be understood as thickened fibers:

$$\widetilde{\Phi}^{-1}(\mathscr{S}_i) = \{\theta \in \Theta \mid \Phi(\theta) \in \mathscr{S}_i\}.$$

Within such a region, variations in θ do not produce semantically meaningful changes in $\mathscr{E}$: the realization map is locally insensitive, $\|\Phi(\theta+\delta\theta)-\Phi(\theta)\| \ll 1$. An SFS therefore corresponds to a region where the quotient projection has already collapsed a large number of degrees of freedom—a local instance of constraint closure in parameter space.

9.3. Semantic Shift as Motion in the Quotient

Let $\{\mathscr{P}_\lambda\}$ be a family of partitions of $\mathscr{E}$ indexed by scale. Semantic Shift, previously defined as entropy variation across scales, can now be interpreted geometrically as motion in the quotient space: the equivalence class $[\theta]_\lambda$ migrates as λ increases, with finer classes nested within coarser ones,

$$[\theta]_{\lambda_1} \subset [\theta]_{\lambda_2} \quad \text{for } \lambda_1 < \lambda_2.$$

This induces a hierarchy of quotients:

$$\Theta \to \Theta/{\sim_{\lambda_1}} \to \Theta/{\sim_{\lambda_2}} \to \cdots$$

Semantic Shift is therefore not merely a change in labeling but a functorial projection between quotient spaces—a structured movement through a tower of increasingly coarse representations.

9.4. Collapse versus Invariance-Preserving Abstraction

This perspective clarifies the distinction between pathological collapse and structure-preserving abstraction. Let $T_\theta\Theta$ decompose as

$$T_\theta\Theta = T_\theta^{\parallel} \oplus T_\theta^{\perp},$$

where $T^{\parallel}$ is tangent to fibers (the null directions of $d\Phi$) and $T^{\perp}$ is transverse to fibers (the semantically active directions). A valid semantic contraction removes components in $T^{\parallel}$ while preserving $T^{\perp}$. Pathological collapse removes components in $T^{\perp}$: it identifies directions that carry genuine semantic content, destroying distinctions that should have been maintained. The kernel of the differential $d\Phi$ is therefore the precise criterion: a quotient is well-formed if and only if it collapses only directions already in $\ker(d\Phi)$.

Theorem 9.1 (Fiber-Preserving Quotient Criterion)**.** *Let* $\pi_\lambda : \mathscr{E} \to \mathscr{E}_\lambda$ *be a projection at scale* λ *and* $\Phi_\lambda = \pi_\lambda \circ \Phi$*. Then the quotient at scale* λ *preserves semantic structure if and only if* $\ker(d\Phi) \subseteq \ker(d\Phi_\lambda)$.

Proof. In the forward direction, if semantic structure is preserved then any direction in Θ that does not change Φ must also leave Φ_λ unchanged, so $\ker(d\Phi) \subseteq \ker(d\Phi_\lambda)$. Conversely, if this inclusion holds, then all directions collapsed by Φ remain collapsed under Φ_λ, and no new transverse directions are identified unless they already lie within $\ker(d\Phi_\lambda)$. The quotient therefore removes only directions already semantically invariant under Φ. □

9.5. Completion as Quotient Stability

A semantic structure is complete at scale λ^* when the induced quotient stabilizes:

$$\Theta/\sim_{\lambda^*} \;\cong\; \Theta/\sim_{\lambda^*+\varepsilon}.$$

Equivalently, the projection map ceases to identify new directions: $\ker(d\Phi_{\lambda^*}) \approx \ker(d\Phi_{\lambda^*+\varepsilon})$. At this point, further aggregation produces no new semantic invariants, and the system has achieved structural closure at the quotient level. Completion is not the elimination of degrees of freedom but the exhaustion of meaningful quotient directions. What remains is pure projection: rendering without structural change.

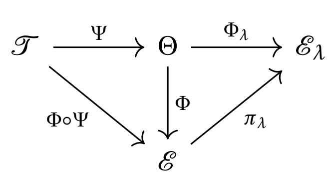
10. Functorial Structure of the Projection Hierarchy

10.1. The Trajectory Category

The relationships developed across the preceding sections can be expressed as a sequence of composable maps forming a commutative diagram. To make this structure explicit, we first endow the space of trajectories with categorical structure.

Let **Traj** be a category whose objects are states and whose morphisms $\tau : x \to y$ are trajectories—ordered sequences of events—transforming state x into state y. Composition is concatenation, which is associative, and each object has an identity trajectory. We further define a tensor product $\otimes$ representing parallel or independent evolution of subsystems, making $(\mathbf{Traj}, \circ, \otimes)$ a symmetric monoidal category. Sequential composition models temporal evolution; tensor product models factorized processes; the identity trajectory models null evolution.

Let $\mathbf{Sem}_\lambda$ be a category of semantic field subspaces at scale λ. The full projection hierarchy can then be expressed as the commutative diagram



Here $\mathscr{T}$ is the space of trajectories, $\Psi : \mathscr{T} \to \Theta$ maps histories to parameter configurations, $\Phi : \Theta \to \mathscr{E}$ is the realization map, $\pi_\lambda : \mathscr{E} \to \mathscr{E}_\lambda$ is the semantic projection at scale λ, and $\Phi_\lambda = \pi_\lambda \circ \Phi$ is the coarse-grained realization. The diagram commutes: $\pi_\lambda \circ \Phi \circ \Psi = \Phi_\lambda \circ \Psi$, so projecting after realization is equivalent to realizing directly at the coarse scale.

Each arrow represents a reduction of degrees of freedom. Ψ compresses temporal structure into state; Φ collapses implementation into representation; π_λ collapses representation into semantic equivalence classes. Semantic structure is obtained by successive quotienting operations applied to an underlying trajectory space.

10.2. Semantic Fields as a Functorial Image

We now define a functor

$$\mathscr{F} : \mathbf{Traj} \to \mathbf{Sem}_\lambda$$

that factors as $\mathscr{F} = \pi_\lambda \circ \Phi \circ \Psi$. On objects, $x \mapsto \mathscr{F}(x) \in \mathscr{E}_\lambda$; on morphisms, $\tau : x \to y$ maps to $\mathscr{F}(\tau)$, the induced transformation in semantic space. Functoriality requires

$$\mathscr{F}(\tau_2 \circ \tau_1) = \mathscr{F}(\tau_2) \circ \mathscr{F}(\tau_1),$$

so semantic evolution respects the compositional structure of trajectories. Under the monoidal structure, $\mathscr{F}(\tau_1 \otimes \tau_2) \cong \mathscr{F}(\tau_1) \otimes \mathscr{F}(\tau_2)$ when $\mathbf{Sem}_\lambda$ carries a compatible product. Semantic structure is therefore not computed from static inputs but is functorially induced from trajectories.

10.3. Collapse as Non-Faithfulness

The functor $\mathscr{F}$ is generally not faithful: $\tau_1 \neq \tau_2$ but $\mathscr{F}(\tau_1) = \mathscr{F}(\tau_2)$. This is necessary for abstraction to be possible at all, since abstraction is precisely the identification of trajectories that differ at a finer level but agree at the level of semantic output. The failure mode arises when $\mathscr{F}$ identifies morphisms that should remain distinct: $\tau_1 \not\approx \tau_2$ but $\mathscr{F}(\tau_1) = \mathscr{F}(\tau_2)$. The distinction between abstraction and collapse is therefore a distinction between controlled and uncontrolled non-faithfulness. A semantic system is well-formed if $\mathscr{F}$ is faithful on a chosen subcategory of structurally relevant trajectories and non-faithful only on trajectories that are already equivalent under the intended structure.

10.4. Completion as Functor Stabilization

Let $\mathscr{F}_\lambda$ denote the functor at scale λ. Completion occurs at λ^* when

$$\mathscr{F}_{\lambda^*} \cong \mathscr{F}_{\lambda^*+\varepsilon},$$

equivalently when $\mathrm{Im}(\mathscr{F}_{\lambda^*}) = \mathrm{Im}(\mathscr{F}_{\lambda^*+\varepsilon})$. At this point, further projection does not change the semantic structure: the functor has reached a fixed point under scale refinement. Completion is a property of the functor, not of any particular object it produces.

10.5. From Objects to Histories

Objects do not exist as primary entities in this ontology. What exists are trajectories in $\mathscr{T}$, and all observable structure arises from successive projections of these trajectories. The mesh, the painting, the embedding vector—each is a slice through a deeper process. Completion is not a property of the slice but of the stability of the projection under further quotienting. The tower of quotients over trajectory space terminates when the functor stabilizes, and that termination is what the practitioner recognizes as “done.”

11. RSVP Dynamics as Infinitesimal Generator

11.1. From Static Functor to Dynamical Generator

The functor $\mathscr{F} : \mathbf{Traj} \to \mathbf{Sem}_\lambda$ has thus far been treated as a static projection. We now refine this by identifying $\mathscr{F}$ as the time-integrated effect of an underlying dynamical system: the RSVP field equations. This move converts the categorical framework from a descriptive structure into a generative one, and gives the notion of completion a dynamical rather than merely logical character.

Let (Φ, v, S) denote the RSVP fields over embedding space, where $\Phi(x,t)$ is semantic density, $v(x,t)$ is semantic flow, and $S(x,t)$ is local entropy. Semantic structure evolves according to the coupled system

$$\partial_t \Phi = -\nabla \cdot (\Phi v) + D\nabla^2 \Phi - \lambda \frac{\delta \mathscr{J}}{\delta \Phi},$$

where $\mathscr{J}$ is the variational functional defining semantic field subspaces and $D > 0$ is a diffusion coefficient. Each trajectory $\tau \in \mathbf{Traj}$ corresponds to an integral curve of this system: $\tau \sim \{\Phi(t), v(t), S(t)\}_{t\in[0,T]}$. The functor $\mathscr{F}$ is obtained by integrating the flow and projecting,

$$\mathscr{F}(\tau) = \pi_\lambda\big(\Phi(T)\big),$$

so $\mathscr{F}$ is not primitive but generated by RSVP dynamics followed by quotienting. The operator

$$\mathscr{L}[\Phi] = -\nabla \cdot (\Phi v) + D\nabla^2 \Phi - \lambda \frac{\delta \mathscr{J}}{\delta \Phi}$$

is the infinitesimal generator of semantic evolution, and the full functor is the time-T flow

$$\mathscr{F} = \pi_\lambda \circ e^{T\mathscr{L}} \circ \Phi \circ \Psi.$$

11.2. Completion as a Fixed Point of the Flow

Completion can now be characterized dynamically. A system is internally complete when $\mathscr{L}[\Phi] \approx 0$: the field has reached a stationary solution and further evolution produces no meaningful change. Equivalently, $e^{T\mathscr{L}}\Phi \approx \Phi$ for all $T > 0$. Completion is therefore not a terminal state but a fixed point of a dynamical system—a condition on the generator, not on any particular output.

This is precisely the invariant that the practitioner identifies as "done." What appears as a subjective judgment—this painting needs nothing more, this script is finished, this argument is closed—corresponds to the detection of a vanishing generator: the sense that any further operation would leave the essential structure unchanged.

11.3. Collapse as Instability

In the dynamical picture, collapse corresponds to instability in the flow. If perturbations grow under $\mathscr{L}$—if $\|\delta\Phi(t)\| \uparrow$ along certain directions—then the system merges distinct semantic regions, producing drift. The distinction between good abstraction and pathological collapse becomes a spectral condition on $\mathscr{L}$: a well-formed system contracts along irrelevant directions (the null directions of $d\Phi$, tangent to fibers) while preserving transverse directions (the semantically active directions). Collapse occurs when $\mathscr{L}$ contracts along transverse directions: it suppresses the very signal it should preserve.

11.4. Lyapunov Structure of Completion

To make the fixed-point characterization rigorous, we define a Lyapunov functional that is non-increasing along the flow and minimized exactly at completion.

Define

$$\mathscr{E}[\Phi] = \int \Phi \log \Phi \, dx + \alpha \int \|\nabla \Phi\|^2 \, dx + \beta \mathscr{J}[\Phi],$$

where the three terms penalize dispersion, instability, and semantic incoherence respectively.

Theorem 11.1 (Lyapunov Stability of Semantic Field Dynamics)**.** *Let $\Phi(x,t)$ evolve under $\partial_t \Phi = \mathscr{L}[\Phi]$ with $D, \alpha, \beta, \lambda > 0$ and mass-preserving boundary conditions. Then $\frac{d}{dt}\mathscr{E}[\Phi(t)] \le 0$.*

Proof. Computing $\frac{d}{dt}\mathscr{E}$ yields three contributions. The diffusion term gives

$$\int (1 + \log \Phi) D \nabla^2 \Phi \, dx = -D \int \frac{\|\nabla \Phi\|^2}{\Phi} dx \le 0.$$

The gradient regularization term, after integration by parts, gives

$$2\alpha \int \nabla \Phi \cdot \nabla(\partial_t \Phi) \, dx = -2\alpha \int \|\nabla^2 \Phi\|^2 dx \le 0.$$

The variational forcing term gives

$$\beta \int \frac{\delta \mathscr{J}}{\delta \Phi} \left(-\lambda \frac{\delta \mathscr{J}}{\delta \Phi} \right) dx = -\beta \lambda \int \left\| \frac{\delta \mathscr{J}}{\delta \Phi} \right\|^2 dx \le 0.$$

The advection term $\int (1 + \log \Phi)(-\nabla \cdot (\Phi v)) \, dx$ vanishes under mass-preserving boundary conditions. All remaining terms are non-positive, so $\frac{d}{dt}\mathscr{E} \le 0$. □

Corollary 11.2. *If $\mathscr{E}$ is bounded below, $\Phi(t)$ converges to a stationary point of $\mathscr{L}$, and that stationary point corresponds to a completed semantic structure.*

The informal intuition that opened this monograph—the coloring book, the stopped painting, the headless script—has now been elevated to a precise dynamical claim. A system is done when further evolution cannot decrease its Lyapunov functional. The practitioner who stops at structural closure is not abandoning the work; they have detected, however implicitly, that the generator has reached zero. The rendering that remains is a traversal of a fixed point, not a continuation of the dynamics.

12. Temporal Constraints and the Management of Entropy

The framework developed so far assumes that time is not a variable: completion is declared when structural entropy goes to zero, and the question of when this happens relative to an external clock is not addressed. In most contexts this is adequate. An essay is complete when its argument is structurally closed, regardless of whether that takes an hour or a year. A script is complete when it is generator-closed, regardless of when the last line was written.

In live performance, however, time becomes a hard constraint. The performance has a fixed duration T, determined externally—by the musical set, the concert program, the event schedule.

The work must reach external completion at time T. It cannot stop early (dead time is worse than an unfinished work) and it cannot run over (the shared temporal structure of the event imposes a hard boundary). The problem is no longer when to stop but how to pace.

This introduces an inversion of the standard completion problem. In the standard case, structural closure is the goal and the remaining projection is set aside. In the live performance case, structural closure is trivially achieved near the beginning—the practitioner who has done this before knows the subject, the compositional strategy, and the execution plan within the first minutes—and the problem is to engineer the passage from internal closure to external closure over the full duration T.

Define the temporal completion problem formally as follows. Let $[0, T]$ be the performance interval. At time $t = 0$, the practitioner achieves internal closure: the constraint set $\mathscr{C}$ is fixed and $|\mathscr{M}| = 1$. The task is to schedule the projection π across $[0, T]$ such that external closure is achieved precisely at $t = T$.

Let $L : [0, T] \to [0, 1]$ be the legibility function, measuring the degree to which the projected image at time t satisfies the observer's recognition criteria. The constraints on L are:

$$L(0) \approx 0, \quad \frac{dL}{dt} \geq 0, \quad L(T) = 1.$$

The first constraint says the work begins in an unrecognized state—the first marks are not yet legible as a subject. The second says legibility increases monotonically—the work never becomes less recognizable over time. The third says legibility reaches its maximum precisely at the end of the performance.

Proposition 12.1. *The temporal completion problem is well-posed if and only if the practitioner achieves internal closure at* $t = 0$ *and the set of available projection operations is rich enough to realize any monotone trajectory from* $L(0) \approx 0$ *to* $L(T) = 1$.

Proof. Well-posedness requires a solution to exist. If internal closure is not achieved at $t = 0$, the practitioner must spend part of the interval $[0, T]$ on structural determination rather than pacing the projection, and the timing problem becomes underdetermined. If the projection operations are not rich enough to control $L(t)$, the trajectory cannot be shaped to meet the boundary conditions. Both conditions are necessary; together they are sufficient for the existence of a valid pacing strategy. □

This proposition clarifies why experience matters in live performance. The practitioner who achieves internal closure immediately—who knows at the first mark what the final work will be—has the full interval $[0, T]$ available for pacing. The practitioner who is still resolving structural questions midway through the set has less interval available and must accelerate the projection to meet the terminal condition, producing a qualitatively different trajectory. The experienced practitioner's advantage is not primarily technical but structural: they have a richer constraint vocabulary that collapses the solution manifold faster.

13. The Engineering of Legibility

Given the temporal completion problem, the central practical question is how to design the trajectory $L(t)$. The constraints say it must be monotone and reach 1 at T. But within those constraints, there is a family of possible trajectories, and different trajectories produce different audience experiences. The choice of trajectory is the primary craft problem in live performance.

The dominant strategy in practice is a specific shape: $L(t)$ rises slowly and approximately linearly for most of the interval, then accelerates sharply near $t = T$. This shape is achieved through information layering: low-frequency information—large forms, overall composition, dominant value structure—is projected early, while high-frequency information—identity features, specific textures, individual marks—is reserved for the final interval.

Definition 13.1 (Information Frequency)**.** The *frequency* of a projection operation is the degree to which it specifies the identity of the subject: low-frequency operations establish large-scale structure, high-frequency operations establish specific recognizable features.

The strategy of large brushes before small, background before foreground, form before detail is a direct implementation of this ordering principle. Low-frequency operations contribute to $L(t)$ gradually and broadly; high-frequency operations contribute sharply and specifically. By scheduling low-frequency operations early and high-frequency operations late, the practitioner maintains a low and slowly rising $L(t)$ through most of the performance and reserves the sharp rise for the final minutes.

Proposition 13.2. *A projection schedule that applies low-frequency operations before high-frequency operations maximizes the duration over which the work remains ambiguous and concentrates legibility gain at the end of the interval.*

Proof. Low-frequency operations reduce $S_{\text{perceptual}}$ slowly because they establish structure that is consistent with many possible identities. High-frequency operations reduce it rapidly because they establish specific features that are consistent with few identities. By applying low-frequency operations first, the practitioner holds $S_{\text{perceptual}}$ high—and $L(t)$ low—for most of the interval. The high-frequency operations applied near $t = T$ drive $L(t)$ to 1 rapidly. This produces the desired trajectory shape. □

There is an additional constraint not captured by the legibility function alone: the audience's attention trajectory. A work that rises in legibility too quickly produces a long period of confirmed recognition at the end, which tends toward anticlimax. A work that rises too slowly produces frustration and disengagement. The optimal trajectory is one that maintains just enough ambiguity to sustain curiosity while providing enough developing structure to reward continued attention.

This is the craft of the denouement, and it is not merely a matter of scheduling information. It requires managing the audience's inferential state: their evolving probability distribution over possible subjects. The practitioner must ensure that the work is consistent with several plausible interpretations through most of the interval—that it could be one thing or another—and that the final high-frequency operations resolve the ambiguity sharply and conclusively. The work should look, through most of the performance, like something else: a landscape that becomes a figure, an abstract that becomes a face. The identity must be withheld not by concealment but by genuine structural ambiguity at the perceptual level.

14. The Denouement Operator

The most important event in the temporal completion trajectory is the denouement: the moment at which the observer's inferential state collapses from high entropy to low. This is not merely a subjective experience; it is a structural event in the observer's probability space, and it can be analyzed precisely.

Let p_t be the observer's probability distribution over possible subjects at time t. Initially, p_0 is diffuse: many subjects are consistent with the marks so far. As the projection proceeds, p_t concentrates. The observer's entropy is

$$S_{\text{obs}}(t) = -\sum_i p_t(i) \log p_t(i).$$

Under the monotone legibility constraint, this entropy is non-increasing:

Proposition 14.1. *Under a projection schedule satisfying $\frac{dL}{dt} \geq 0$, observer entropy $S_{obs}(t)$ is non-increasing.*

Proof. Legibility $L(t)$ measures recognition probability, which increases as the observer's distribution concentrates on the correct subject. Concentration of p_t directly reduces $S_{\text{obs}}(t)$ by the definition of entropy. Since $L(t)$ is non-decreasing by assumption, $S_{\text{obs}}(t)$ is non-increasing. □

The denouement occurs at time t_c when this entropy undergoes a rapid transition:

Definition 14.2 (Denouement Time)**.** The *denouement time* $t_c \in [0, T]$ is the time at which

$$S_{\text{obs}}(t_c - \varepsilon) \gg 0 \quad \text{and} \quad S_{\text{obs}}(t_c + \varepsilon) \approx 0$$

for small $\varepsilon > 0$.

In terms of the legibility function, this corresponds to a rapid transition: $L(t_c - \varepsilon) \ll 1$ and $L(t_c + \varepsilon) \approx 1$. The denouement is a phase transition in observer inference: the system passes rapidly from a high-entropy state (many subjects are plausible) to a low-entropy state (the subject is identified).

Remark 14.3. The denouement is experienced phenomenologically as recognition: the moment at which the ambiguous marks snap into a coherent identity. This experience is sharp precisely because the underlying transition is rapid—a small number of high-frequency marks drive the observer's distribution to concentrate on a single subject.

In the live painting context, the practitioner engineers $t_c \approx T$: the denouement is designed to coincide with the end of the performance. This requires careful management of the marks withheld until the final interval. The specific features that establish subject identity—the eyes of a portrait, the distinctive outline of a building, the gesture of a figure—are applied last, after the large-scale structure has been established. Their effect on S_{obs} is disproportionate: a few marks change everything.

The coincidence of t_c and T produces what can be called temporal closure: the ending of the performance and the completion of recognition are simultaneous events. The audience does not experience a finished work and then a silence; they experience recognition and silence as a single event. This temporal fusion is what distinguishes a live performance from a painting simply done in public. The work is not exhibited; it is made to end precisely when the performance ends.

The denouement operator D can be defined abstractly as the minimal set of high-frequency marks required to drive S_{obs} to zero:

$$D = \arg\min_M \{|M| : S_{\text{obs}}(t_c^+ \mid M) \approx 0\},$$

where M ranges over subsets of available marks and t_c^+ denotes the state immediately after applying M. The practitioner's task is to identify D in advance and withhold it until the appropriate moment. This is why the planning of a live painting is simple and rapid—a few seconds to identify D and the broad sequence—while the execution requires the full duration of the set.

15. Music as External Clock

The live painting context introduces an external temporal structure that interacts with the completion problem in a specific and practically significant way. The music that accompanies the performance is not merely ambience; it is a pacing constraint that offloads a significant class of decisions from the practitioner to the environment.

In the absence of external structure, the practitioner must make continuous decisions about the velocity of projection: how quickly to move from large forms to small details, when to pause, when to accelerate. These decisions are both cognitively demanding and consequential—a trajectory that accelerates too early produces premature legibility, while one that accelerates too late is at risk of not reaching $L(T) = 1$.

Music functions as a shared temporal reference that eliminates this decision class. The practitioner synchronizes the velocity of the projection to the music's rhythmic and dynamic structure, using the music as a clock that determines not just the overall duration but the local pacing at each moment. When the music swells, the projection accelerates; when it sustains, the projection can dwell in a region. The pacing decision is replaced by a pacing response.

Definition 15.1 (External Clock)**.** An *external clock* for a performance of duration T is a shared temporal signal $\mu : [0, T] \to \mathbb{R}^+$ that provides a locally agreed-upon velocity reference for the projection.

The music serves as an external clock in this sense. It provides both global structure—the total duration is known in advance—and local structure—the dynamic and rhythmic variation at each moment signals the appropriate velocity of the projection. The practitioner does not need to maintain an internal model of elapsed time or decide how fast to move; these are offloaded to the auditory environment.

Proposition 15.2. *An external clock reduces the cognitive overhead of the temporal completion problem by eliminating the need for explicit velocity management.*

Proof. The temporal completion problem requires the practitioner to solve for a trajectory $L(t)$ satisfying the boundary and monotonicity conditions. Without an external clock, this requires continuous internal tracking of t, remaining duration $T - t$, and current legibility $L(t)$. With an external clock μ, the practitioner can couple the projection velocity to $\mu(t)$, delegating the timing function to the shared signal. The internal tracking requirement is reduced to a reference check: is the projection ahead of, behind, or synchronized with the external signal? □

There is an additional benefit beyond cognitive offloading. The music is perceived by both the practitioner and the audience simultaneously. By coupling the projection to the music, the practitioner ensures that the pacing of the work is experienced as continuous with the pacing of the performance as a whole. The audience, attending to both music and emerging image, experiences them as co-evolving. The denouement of the image—the moment of recognition—is prepared for by the musical climax or resolution toward which the set has been building. The two temporal closures reinforce each other.

This is also why the physical positioning of the practitioner matters. By staying out of the audience's line of sight while mixing and preparing, and by ensuring that the work-in-progress rather than the work-being-made is what the audience sees, the practitioner maintains the illusion that the image is developing autonomously—emerging in response to the music rather than being applied by a person following a plan. This sustains the audience's experience of the work as an event rather than a demonstration.

16. Epistemic Asymmetry and Collaboration Friction

The disagreements that arise in collaborative creative work are, as a structural matter, disagreements about what kind of object completion is predicated of. The analysis developed in the preceding sections provides a precise vocabulary for these disagreements, and it is worth drawing the implications explicitly.

Consider the general case. Two agents—a creator and an observer, or two creators—are working on the same project. The creator has access to the constraint set $\mathscr{C}$ and therefore knows when $|\mathscr{M}| = 1$: they can detect internal closure. The observer has access only to the projected configuration $\pi(x)$ and must infer structural closure from external evidence. Since π is not injective—many structurally distinct configurations project to perceptually similar states, and structurally equivalent configurations may project to perceptually different ones—the observer's inference is unreliable. They may perceive incompleteness where there is structural closure, and closure where there is not.

Definition 16.1 (Epistemic Asymmetry)**.** *Epistemic asymmetry* between a creator and an observer arises when the creator has access to $\mathscr{M}$ directly and the observer has access only to $\pi(x)$. The asymmetry is the informational gap between $\mathscr{M}$ and $\pi^{-1}(\pi(x))$.

This asymmetry is irreducible without communication. The observer cannot recover $\mathscr{M}$ from $\pi(x)$ alone without additional information about the constraint structure. The creator can reduce the asymmetry by providing that information—through explanation, documentation, or completing the projection to $\Delta = 0$—but they cannot eliminate it from the observer's side without either revealing the constraints explicitly or rendering the work to full external completion.

The epistemic asymmetry between creator and observer is the global case. The collaboration friction case is a specific and structurally distinct instance in which both parties are creators, each with their own constraint vocabulary, operating at different levels of closure.

Return to the pottery studio. The ceramicist has thrown a pot: the form is established, the walls are even, the profile is decided. In the ceramicist's constraint vocabulary, the substrate is closed—it is ready to receive surface treatment. She says: move on to the next one. The painter has been applying painted figures to the surface. In the painter's constraint vocabulary, the figure is not yet closed—the lighting model is established, the composition is fixed, but the rendering has not projected these internal closures into perceptual legibility. He says: not yet done.

This is not a disagreement between a creator and an observer. Both parties are creators. The disagreement is between two different loci of closure: the ceramicist is working at the level of substrate completion, while the painter is working at the level of perceptual projection. Each is correct from within their own constraint vocabulary. The friction is not about differing standards for what counts as finished; it is about differing levels of the representational hierarchy at which closure is being tracked.

Definition 16.2 (Collaboration Friction)**.** *Collaboration friction* between two creators arises when they operate at different levels of the entropy collapse sequence, such that one has declared closure at level k while the other has not yet closed at level $k-1$.

Proposition 16.3. *Collaboration friction cannot be resolved by appeal to shared aesthetic standards. It requires either explicit alignment on the target closure level or a workflow partition that insulates each creator's closure criterion from the other's.*

Proof. The friction arises from a structural mismatch in which level of the representational hierarchy is being treated as primary. Appeals to shared aesthetic standards address the question of what constitutes a well-executed projection, not the question of which level the projection is being evaluated at. Only explicit agreement on the target level—or a partition of labor that assigns each level to a single creator—can eliminate the mismatch. □

The mural example instantiates this perfectly. The ceramicist's standard—"ready for the next substrate"—is a level-k closure criterion. The painter's standard—"perceptually projected"—is a level-$(k-1)$ criterion. The disagreement is not about quality but about level. Once this is understood, the resolution is simple: establish in advance which level the collaboration tracks as its shared completion criterion, or divide the work so that the ceramicist's criterion governs the substrate phase and the painter's criterion governs the surface phase without either being imposed on the other.

17. The Relocation of Rendering

A potential misreading of the entropy collapse framework is that rendering disappears as completion migrates upward. If completion is declared at the generator level, does the projection from script to mesh to render become irrelevant? The answer is no, and this requires explicit correction.

Rendering never disappears. It relocates. Each time the locus of completion shifts upward by one level—from perceptual to spatial to generative—the rendering work that was previously performed at that level does not cease to exist; it is transformed into configuration and parameterization work at the level above.

The painter who stops at the constraint-closure level and does not apply highlights is not eliminating the work of surface rendering. They are relocating it: either to a collaborator who will complete the surface, or to a future session, or to the imagination of a viewer who reconstructs the finished surface from the established constraints. The work exists; it is simply no longer the current agent's responsibility.

More precisely, each level of the hierarchy has its own rendering problem. At the script level, rendering consists of parameter selection: the choice of resolution, material properties, randomness seeds, export formats, and pipeline configurations. These are not decisions that the script makes automatically; they must be specified somewhere. When completion migrates to the script level, these choices migrate with it. The practitioner who declares closure at the script level is implicitly declaring that these parameterization choices are either irrelevant or already determined. If they are not, the closure is incomplete.

Proposition 17.1. *Closure at level k creates an open rendering problem at level $k+1$: the parameterization required to project from level k to level $k-1$.*

Proof. A generator-closed script θ specifies the transformation $\Phi : \Theta \to \Omega_{\text{mesh}}$ but does not specify the downstream projections $\rho : \Omega_{\text{mesh}} \to \Omega_{\text{render}}$ or $\pi : \Omega_{\text{render}} \to \mathscr{Y}$. These require additional parameterization—materials, lighting, camera position, export format—that is not determined by θ. Closure at the generator level therefore leaves open a family of projection choices, each of which constitutes a rendering problem at the next level. □

This proposition has a corrective function. The claim that completion migrates upward does not mean that the lower levels become trivial in an absolute sense; it means they become trivial relative to the constraint structure that has already been fixed. The highlights on a painting are trivial relative to the established composition and lighting model. The export format for a mesh is trivial relative to the established geometry. But "trivial" here means "structurally irrelevant," not "requiring no effort." The effort of rendering is real; it is simply not the kind of effort that alters the structural identity of the work.

The clearest formulation is this: rendering is the work of closing Δ, the residual encoding distance between the current projected state and the observer's recognition threshold. This work is real and audience-dependent. It does not disappear because structural closure has been achieved; it becomes the work that remains, the projection of a solved problem into a form that communicates itself without requiring reconstruction.

18. Three Modes of Completion

The analysis can now be synthesized into three distinct modes of completion, each corresponding to a different relationship between internal closure, external projection, and the deployment of time.

Generator Mode. In generator mode, the practitioner declares completion at the level of structural closure: $|\mathscr{M}| = 1$. The projection from $\mathscr{M}$ into $\mathscr{Y}$ is not performed, or is delegated to a future agent. The work exists as a solved structure: a closed constraint set, a fixed argument, a generator-closed script. The residual encoding distance Δ is positive but treated as irrelevant relative to the audience or context. This is the mode of the bpy script, the structurally complete but unpolished essay, the established-but-unrendered composition. It is the appropriate mode when the audience can execute the projection themselves or when the work is private.

Delegated Mode. In delegated mode, the practitioner establishes structural closure and makes it explicit enough for a second agent to perform the projection. The coloring book is the paradigm case: the outlines encode the constraint set in perceptual space, making the remaining degrees of freedom visible and accessible. The work is split at the closure boundary: one agent determines, another projects. This requires a richer interface than generator mode—the constraint set must be communicated, not merely held—but it does not require the practitioner to perform the projection themselves.

Performed Mode. In performed mode, the practitioner achieves internal closure early and then stages the projection for an audience across a fixed temporal interval. The projection is real—actual marks are applied, actual information is released—but its timing is managed to produce a specific audience experience. The structural solution is already determined; the performance is the controlled revelation of that solution. This is the mode of live painting, live coding, and certain forms of improvisational music. It requires the practitioner to hold two timelines simultaneously: the internal timeline, on which everything is already decided, and the external timeline, on which the decision is only now becoming visible.

Theorem 18.1. *Generator, delegated, and performed modes are distinct instantiations of a single underlying structure: completion as constraint closure, with different policies for the management of the residual encoding distance* Δ.

Proof. In all three modes, internal completion is defined identically as $|\mathscr{M}| = 1$. The modes differ only in how Δ is handled. Generator mode leaves $\Delta > 0$ and makes no claim about closing it. Delegated mode reduces Δ to the threshold required for a second agent to execute the projection. Performed mode schedules the reduction of Δ over a fixed interval $[0, T]$, engineering a specific trajectory for the observer's epistemic state. Since the underlying completion condition is identical in all three cases, and the modes differ only in their policies for the residual encoding distance, they are instantiations of a common structure. □

19. Implications

The framework developed here has implications beyond its immediate domain. Three areas deserve explicit attention: software engineering, artificial intelligence, and pedagogy.

In software engineering, the distinction between internal and external completion maps directly onto the distinction between specification and implementation. A system is internally complete when its specification is closed—when the design decisions have been made and the remaining implementation is deterministic projection. External completion requires that the implemented system be sufficiently legible for operators, users, and future maintainers to interact with it without reconstructing the specification. Many software failures, particularly at scale, are failures of external completion in systems that achieved internal closure early: the architecture was sound but insufficiently communicated. The framework suggests that the appropriate investment is not more implementation but more projection-work—documentation, interface design, observability tooling—all of which are forms of Δ-reduction.

In artificial intelligence, the distinction between generator and output is structurally analogous to the distinction between model and sample. A trained model is, in the relevant sense, a generator-closed system: it specifies a distribution over outputs, and any particular output is an evaluation of that generator under specific inputs and sampling parameters. The question of whether a model is "done" is not answered by examining its outputs; it is answered by examining the constraint structure—the training objective, the architecture, the data distribution—that defines the model's solution manifold. Much of the apparent controversy about AI completion—when is a model ready for deployment?—is a confusion between generator closure (does the model's constraint structure satisfy the requirements?) and external completion (does any particular output satisfy the requirements?). These are different questions and require different evaluations.

In pedagogy, the framework inverts the standard instructional sequence. Conventional instruction proceeds from surface to structure: students are taught to produce correct outputs before they are taught the constraint structures that make those outputs correct. This is external-completion-first pedagogy. The trajectory-first approach would proceed in the opposite direction: establish the constraint vocabulary first, then treat output production as projection. Students who understand the constraint structure of an argument can write the argument; students who have been trained to produce argument-shaped texts without understanding the constraints cannot reconstruct the structure from their outputs. The difference is between knowing $\mathscr{M}$ and having learned to sample from $\pi^{-1}(\tau)$.

20. Objections and Formal Responses

The framework developed here invites several objections, each of which, when answered precisely, serves to tighten the theory rather than merely defend it. The three most important objections concern premature termination, the status of material contingency, and the independence of the three levels of characterization.

20.1. Premature Termination

The first objection is that identifying completion with constraint closure terminates work too early, before perceptual completion has been achieved. If a painter declares the mural done when the constraint surface is fixed, the wall is still unpainted. If a scripter declares the model done when the script is generator-closed, the mesh is still unrendered. Is this not simply stopping before finishing?

The objection conflates two formally distinct conditions. Constraint closure—$|\mathscr{M}| = 1$—is an internal condition on the structure of the feasible set under the equivalence relation. Perceptual completion—$\Delta(x) \approx 0$—is an external condition on the projection of a representative into observable space. The theory does not deny the necessity of projection for certain audiences and contexts. It reclassifies projection as a distinct phase with a distinct character: deterministic, delegatable, and epistemically redundant for the agent who holds the constraint structure. Calling this "premature termination" mistakes a phase boundary for an incompletion. The work of determination has ended; the work of communication may or may not continue, depending on context.

Proposition 20.1 (Phase Separation)**.** *For any system with* $|\mathscr{M}| = 1$*, the set of operations that reduce* $\Delta(x)$ *is disjoint from the set of operations that alter* $|\mathscr{M}|$.

Proof. Operations that reduce $\Delta(x)$ are projection operations: they change the representative x within the equivalence class $[x] \in \mathscr{M}$ without altering the class itself. Operations that alter $|\mathscr{M}|$ are constraint-imposing or constraint-removing operations: they change the structure of the quotient. Since the quotient structure is invariant under projection, these operation sets are disjoint. □

20.2. Material Contingency

The second objection is that generator-level closure ignores the contingencies of material realization. A bpy script may be generator-closed, but the actual rendered object will depend on hardware, driver versions, floating-point arithmetic, material definitions, and environmental parameters that are not encoded in the script. Two executions of the same script on different machines may produce geometrically distinct meshes. Does this not undermine the claim that closure resides at the script level?

The objection is correct that execution introduces a family of realizations, but it misidentifies what the theory claims. Generator closure does not assert that all executions are identical; it asserts that all executions lie within a single equivalence class under $\sim$. The equivalence relation $\sim$ is defined by identity-preserving observables: the features that constitute the structural identity of the work. Hardware-dependent floating-point variation, material rendering differences, and minor geometric perturbations all fall within one equivalence class under any reasonable definition of structural identity for a three-dimensional model. They do not alter the topology, the proportions, the spatial relations, or the compositional logic. Generator closure is a claim about the class, not about the representative.

20.3. Independence of the Three Characterizations

The third objection concerns the triad of characterizations:

$$|\mathscr{M}| = 1 \quad \Longleftrightarrow \quad \mathscr{L}[\Phi] = 0 \quad \Longleftrightarrow \quad \mathscr{E} \text{ minimized.}$$

Are these genuinely equivalent, or does the biconditional hold only under strong assumptions that may fail in practice?

The equivalences are indeed conditional. The discrete-to-continuous direction—from $|\mathscr{M}| = 1$ to $\mathscr{L}[\Phi] = 0$—requires that the continuous field Φ correctly represents the discrete structure of $\mathscr{M}$: that the density $\Phi(x, t)$ is supported on $\Omega_{\mathscr{C}}$ and concentrated within the single equivalence class. The continuous-to-variational direction requires that $\mathscr{E}$ is a Lyapunov functional for the flow generated by $\mathscr{L}$, which is established under the conditions of Theorem 11.1. Within those conditions, the three characterizations are genuinely equivalent and characterize the same fixed point from three different levels of resolution: the quotient level, the dynamical level, and the variational level.

The value of having all three is not redundancy but triangulation. The discrete characterization gives the theory its philosophical clarity—completion is exhaustion of structural entropy. The dynamical characterization gives it its physical character—completion is a fixed point toward which the system evolves. The variational characterization gives it its rigorous stopping criterion—completion is energy minimization under a provably non-increasing functional. Each level makes the others more robust.

20.4. Toward a General Theory of Constraint-Driven Systems

The preceding analysis can be consolidated into a structural description that applies across all domains examined in this monograph.

Every constraint-driven system consists of a configuration space Ω, a constraint set $\mathscr{C}$, a solution manifold $\mathscr{M} = \Omega_{\mathscr{C}}/\sim$, a projection $\pi : \Omega \to \mathscr{Y}$, a dynamical operator $\mathscr{L}$, and a Lyapunov functional $\mathscr{E}$. The completion conditions at the three levels are formally equivalent under the conditions established in this monograph:

$$|\mathscr{M}| = 1 \quad \Longleftrightarrow \quad \mathscr{L}[\Phi] = 0 \quad \Longleftrightarrow \quad \mathscr{E} \text{ minimized.}$$

Completion is simultaneously geometric—the manifold has collapsed—dynamical—the generator has reached zero—and variational—the energy cannot decrease further. These are not three different notions of completion but three descriptions of the same fixed point, each appropriate to a different level of analysis.

Collapse, the dual failure mode, occurs when the system's dynamics contract along semantically transverse directions: when the flow suppresses the distinctions it should preserve. Temporal completion, the inversion that arises under hard time constraints, replaces the question of when to stop with the question of how to pace the projection of an already-solved structure into observer time.

Across all three modes—generator, delegated, performed—the practitioner's primary task is the same. Impose constraints, collapse the solution manifold, drive $|\mathscr{M}|$ to 1. The work is the fixed point. Everything else is projection.

21. Conclusion: Completion as a Property of Mappings

The central argument can now be stated in its final form. Completion is not a property of objects. It is a property of mappings: specifically, of the mapping $\Phi : \Theta \to \mathscr{M}$ from the generator space to the solution manifold. A work is complete when that mapping is fixed and the solution manifold has collapsed to a single equivalence class. The objects produced by evaluating the mapping—the rendered image, the formatted document, the exported mesh—are projections of the completed mapping, not the completion itself.

This reframing dissolves several persistent confusions. The confusion between polish and structure becomes legible: polishing is Δ-reduction, not $\mathscr{M}$-determination. The confusion between a finished work and a displayed work becomes legible: the work is finished when $|\mathscr{M}| = 1$, regardless of whether it has been projected into any observer's perceptual space. The confusion between revision and rendering becomes legible: revision alters the history that constitutes the work's identity, while rendering projects an existing history into a new perceptual state.

The three modes of completion—generator, delegated, and performed—are not different standards for when a work is finished. They are different policies for what to do with Δ once internal closure has been achieved. The generator mode treats Δ as the audience's problem. The delegated mode reduces Δ to the threshold required for transfer. The performed mode schedules the reduction of Δ as a temporal event.

Across all three modes, the practitioner's primary task is the same: impose constraints, collapse the solution manifold, drive $|\mathscr{M}|$ to 1. Everything else—rendering, polishing, performing, delegating—is projection. The work is the mapping. The objects are its evaluations.

The coloring-book image that opened the analysis now reveals its full depth. The outlines of a coloring book are not an incomplete painting; they are a complete specification of a painting in a medium that makes the constraint structure directly visible. The person who fills in the colors is not finishing the work; they are rendering the specification into a richer perceptual register. The work was done when the last outline was drawn. What follows is the projection of a solved problem.

A. Constraint Closure and Solution Manifolds

A.1. State Space and Constraints

Let Ω be a configuration space (finite or measurable), and let $\mathscr{C} = \{C_i\}_{i=1}^{k}$ be a collection of constraints, where each

$$C_i : \Omega \to \{0,1\}$$

is an admissibility indicator. The feasible set is

$$\Omega_{\mathscr{C}} = \{x \in \Omega \mid C_i(x) = 1 \text{ for all } i\}.$$

A.2. Structural Equivalence

Define an equivalence relation $\sim$ on $\Omega_{\mathscr{C}}$ by

$$x \sim y \iff x \text{ and } y \text{ are indistinguishable under all identity-preserving observables.}$$

The solution manifold is

$$\mathscr{M} = \Omega_{\mathscr{C}} / \sim.$$

A.3. Formal Definition

Definition A.1 (Constraint Closure)**.** A system is *structurally closed* if $|\mathscr{M}| = 1$: all admissible configurations lie within a single equivalence class.

A.4. Entropy of the Solution Set

Define

$$S_{\text{eff}} = \log|\mathscr{M}|.$$

Proposition A.2. *Constraint closure is equivalent to $S_{eff} = 0$.*

Proof. $|\mathscr{M}| = 1 \iff \log|\mathscr{M}| = 0 \iff S_{\text{eff}} = 0$. □ □

A.5. Partial Order on Constraint Density

Let $\mathscr{C}' \supset \mathscr{C}$ be a strictly richer constraint set. Then $\Omega_{\mathscr{C}}[']\subseteq \Omega_{\mathscr{C}}$, and $|\mathscr{M}'| \le |\mathscr{M}|$. Constraint addition is monotone in the reduction of the solution manifold.

Corollary A.3. *Adding constraints cannot increase S_{eff}. Structural closure is achieved no later than it would be under any subset of the constraint set.*

B. Projection and External Completion

B.1. The Projection Map

Let $\pi : \Omega \to \mathscr{Y}$ be a map from configuration space to observable space. Define the observer's recognition threshold as a subset $\tau \subset \mathscr{Y}$.

Definition B.1 (External Completion)**.** A configuration $x \in \Omega$ is *externally complete* if $\pi(x) \in \tau$.

B.2. Non-Invariance Under Structural Equivalence

Proposition B.2. *External completion is not invariant under* $\sim$.

Proof. Let $x \sim y$ but $\pi(x) \neq \pi(y)$. This is possible whenever π is not constant on equivalence classes—which is generic, since π is defined on Ω, not on $\mathscr{M}$. Then $\pi(x) \in \tau$ does not imply $\pi(y) \in \tau$. External completion therefore depends on the choice of representative within the equivalence class, not on the class itself. □

B.3. Residual Encoding Distance

Define

$$\Delta(x) = \inf_{y \in \tau} d_{\mathscr{Y}}(\pi(x), y)$$

where $d_{\mathscr{Y}}$ is a metric on observable space. The coloring-book regime is $|\mathscr{M}| = 1$ with $\Delta(x) > 0$. Rendering is any operation that reduces Δ while leaving the equivalence class $[x] \in \mathscr{M}$ unchanged.

Proposition B.3. *A rendering operation* $r : \Omega \to \Omega$ *satisfying* $r(x) \sim x$ *and* $\Delta(r(x)) < \Delta(x)$ *exists whenever* $\Delta(x) > 0$ *and the projection* π *is continuous.*

Proof. By continuity of π and the definition of Δ, there exists a path in $\pi^{-1}(\mathscr{Y})$ from $\pi(x)$ toward τ. A small step along this path reduces Δ. If the step can be taken while remaining in $[x]$ under $\sim$—which holds when the equivalence class has positive measure in the direction of τ—the rendering operation exists. □

C. Generative Mappings and Headless Closure

C.1. Generator Space

Let Θ be a parameter space (script space, model space, or program space), and let

$$\Phi : \Theta \to \Omega$$

be a realization map. The induced constraint set on Θ is

$$\Theta_{\mathscr{C}} = \{\theta \in \Theta \mid \Phi(\theta) \in \Omega_{\mathscr{C}}\}.$$

Definition C.1 (Generator Closure)**.** $\theta \in \Theta$ is *generator-closed* if $\theta \in \Theta_{\mathscr{C}}$ and there exists a neighborhood $U \ni \theta$ such that $\Phi(\theta') \sim \Phi(\theta)$ for all $\theta' \in U$.

C.2. Headless Execution

Proposition C.2. *If* θ *is generator-closed, then* $\pi \circ \Phi(\theta)$ *is not required to verify structural correctness.*

Proof. Structural correctness is membership in $\mathscr{M} = \Omega_{\mathscr{C}}/\sim$. This is determined by $\Phi(\theta) \in \Omega_{\mathscr{C}}$ and $[\Phi(\theta)] \in \mathscr{M}$, both of which are properties of θ that can be verified without evaluating π. Since π maps $\Omega \to \mathscr{Y}$ and correctness lives in $\mathscr{M}$, the composition $\pi \circ \Phi$ adds no information relevant to correctness. □

C.3. Hierarchy of Generator Spaces

The generator space Θ is itself a configuration space, and can be subject to higher-order constraints. Let $\Psi : \Xi \to \Theta$ be a meta-generator. Then generator closure at Θ is a constraint on Ξ, and the hierarchy extends indefinitely upward. At each level, "done" is the condition $|\mathscr{M}_k| = 1$ for the solution manifold at that level.

D. Temporal Legibility and Information Release

D.1. Legibility as Cumulative Recognition

Let $L : [0, T] \to [0, 1]$ be defined as

$$L(t) = \Pr[\text{observer correctly identifies subject at time } t].$$

Definition D.1. L is a *valid legibility trajectory* if $\frac{dL}{dt} \geq 0$ and $L(T) = 1$.

D.2. Observer Entropy

Let p_t be the observer's distribution over possible subjects. Then

$$S_{\text{obs}}(t) = -\sum_i p_t(i) \log p_t(i).$$

Proposition D.2. *Under a valid legibility trajectory, S_{obs} is non-increasing.*

Proof. $L(t) = \sum_i \mathbf{1}[i = \text{true subject}] \cdot p_t(i) = p_t(\text{true subject})$. If L is non-decreasing, the probability mass on the correct subject is non-decreasing, which requires concentration of p_t. Concentration of p_t implies $\frac{d}{dt} S_{\text{obs}}(t) \leq 0$. □

D.3. Denouement Threshold

Definition D.3 (Denouement Time)**.** $t_c \in [0, T]$ is a *denouement time* if

$$\lim_{\varepsilon \to 0^+} \frac{L(t_c + \varepsilon) - L(t_c - \varepsilon)}{\varepsilon} = \max_{t \in [0,T]} \frac{dL}{dt}.$$

The denouement time is the time of maximum rate of recognition gain—the inflection point of the legibility trajectory at which the observer's entropy collapses most rapidly.

Corollary D.4. *A live performance is optimally structured when $t_c = T$: the maximum rate of recognition gain coincides with the end of the performance.*

E. Worked Example: Mural as Manifold Collapse

Let Ω_{mural} be the space of all possible paintings on a given surface. Define constraints:

C_1 : lighting direction is fixed (upper left, angle α_0),

C_2 : compositional flow is established (foreground figure, right-of-center),

C_3 : subject identity is fixed (dragon, specific pose).

The feasible set $\Omega_{\mathscr{C}} \subset \Omega_{\text{mural}}$ is the collection of all paintings consistent with these three constraints. The equivalence relation $\sim$ identifies paintings that share subject, composition, and lighting model but differ in surface treatment (texture, stroke character, value range within the lighting model).

The solution manifold $\mathscr{M} = \Omega_{\mathscr{C}}/\sim$ contains a single equivalence class: given C_1, C_2, and C_3 jointly, there is one painting—in the structurally relevant sense—and the remaining variation is non-structural.

The residual degrees of freedom are:

$$\{\text{highlight placement, shadow softness, texture character, stroke variation}\}.$$

Each of these operates within $[x]$ and does not change $[x]$. Rendering operations that vary these parameters are traversals within the single equivalence class.

Conclusion E.1. The mural is structurally complete at the moment $C_1 \wedge C_2 \wedge C_3$ is satisfied. All subsequent work is projection within a single equivalence class. The work is a coloring book.

F. Worked Example: bpy Script as Generator

Let Θ be the space of Blender Python scripts, and let $\Phi : \Theta \to \Omega_{\text{mesh}}$ be the evaluation map. A script $\theta \in \Theta$ specifies a sequence of mesh operations: vertex creation, extrusion, subdivision, material assignment, transformation.

The structural constraints on the target mesh are:

$$\begin{aligned} &C_1 : \text{topology satisfies the target manifold type,}\\ &C_2 : \text{proportions are within specified tolerances,}\\ &C_3 : \text{spatial relations between components are fixed.} \end{aligned}$$

A script θ is generator-closed if $\Phi(\theta) \in \Omega_{\mathscr{C}}$ and small perturbations to θ—changing a loop range, adjusting a transformation parameter—produce meshes in the same equivalence class.

The GUI, the render preview, and the viewport display are all evaluations of $\pi \circ \Phi(\theta)$: they project the mesh into perceptual space. Once generator closure is verified—by checking the script against the constraints, not by inspecting the viewport—these projections are redundant for correctness purposes.

Conclusion F.1. Closure occurs in Θ. The mesh $\Phi(\theta)$ is an evaluation of the closed generator. Headless execution is the natural workflow for a generator-closed script.

G. Worked Example: Live Painting as Controlled Trajectory

Let $H_T = (e_1, \ldots, e_T)$ be the sequence of brush strokes in a live painting over performance interval $[0, T]$. Define

$$x(t) = \text{Eval}(H_t).$$

Internal closure is achieved at $t \approx 0$: within the first minutes, the practitioner has established subject, composition, and execution strategy. Formally, $|\mathscr{M}| = 1$ at the outset.

The problem is to design H_T such that:

$$L(0) \approx 0, \quad \frac{dL}{dt} \geq 0, \quad L(T) = 1, \quad t_c \approx T.$$

Strategy: order strokes by frequency.

Let $F(e_i)$ denote the frequency of stroke e_i—its contribution to observer entropy reduction. A valid trajectory satisfies

$$i < j \implies F(e_i) \leq F(e_j),$$

i.e., strokes are applied in non-decreasing frequency order: background washes before compositional structure before identity marks.

The denouement operator D is the set of high-frequency strokes:

$$D = \{e_i : F(e_i) > F_{\text{threshold}}\},$$

reserved for the interval $[T - \varepsilon, T]$.

Proposition G.1. *Under this strategy, $t_c \approx T$ and the observer's entropy remains high until the final interval.*

Proof. Low-frequency strokes applied during $[0, T - \varepsilon]$ build structure consistent with multiple subjects, holding $S_{\text{obs}}(t)$ high. The high-frequency strokes in D applied during $[T - \varepsilon, T]$ specify the subject uniquely, driving S_{obs} to zero. The maximum rate of recognition gain therefore occurs in $[T - \varepsilon, T]$, placing $t_c \approx T$. □

Conclusion G.2. The live painting is a staged projection of a pre-solved system. Structural closure precedes the performance; the performance is the controlled revelation of that closure to an audience across a shared temporal structure.

H. Semantic Collapse and Fiber-Quotient Correspondence

H.1. Metric Measure Space Framework

Let $(\mathscr{E}, d, \mu)$ be a metric measure space where $\mathscr{E} \subset \mathbb{R}^d$, d is the Euclidean metric, and μ is a probability measure over $\mathscr{E}$ induced by a corpus. A semantic observable is a measurable function $f : \mathscr{E} \to \mathbb{R}$. Define an equivalence relation by

$$x \sim_\varepsilon y \iff \forall f \in \mathscr{F}, \ |f(x) - f(y)| < \varepsilon$$

for a chosen function class $\mathscr{F}$. The quotient $\mathscr{E}/\sim_\varepsilon$ partitions $\mathscr{E}$ into semantic regions, and a Semantic Field Subspace $\mathscr{S}_i$ is a maximal connected component of an equivalence class under $\sim_\varepsilon$.

H.2. Variational Characterization

Semantic field subspaces arise as minimizers of the functional

$$\mathscr{J}[\{\mathscr{S}_i\}] = \sum_i \left(\int_{\mathscr{S}_i} \|x - \bar{x}_i\|^2 \, d\mu(x) + \beta \, \mathrm{Per}(\mathscr{S}_i) \right),$$

where $\bar{x}_i$ is the centroid of $\mathscr{S}_i$ and $\mathrm{Per}(\mathscr{S}_i)$ is a boundary regularization term. In the continuum limit, with density field $\Phi(x) = d\mu/dx$ and indicator field $\chi_i(x)$, this becomes

$$\mathscr{J}[\chi] = \sum_i \int \chi_i(x) \|x - \bar{x}_i\|^2 \Phi(x) \, dx + \beta \int |\nabla \chi_i(x)| dx.$$

Minimizers produce piecewise-smooth regions corresponding precisely to semantic field subspaces.

H.3. Fiber-Quotient Correspondence

Theorem H.1 (Fiber-Preserving Quotient Criterion)**.** *Let $\Phi : \Theta \to \mathscr{E}$ be a smooth map and $\pi_\lambda : \mathscr{E} \to \mathscr{E}_\lambda$ a projection at scale λ. Define $\Phi_\lambda = \pi_\lambda \circ \Phi$. Then the quotient at scale λ preserves semantic structure if and only if* $\ker(d\Phi) \subseteq \ker(d\Phi_\lambda)$.

Proof. If semantic structure is preserved, any direction not changing Φ must also leave Φ_λ unchanged, establishing the inclusion. Conversely, under the inclusion hypothesis, the quotient removes only directions already in $\ker(d\Phi)$—directions tangent to fibers—and leaves transverse (semantically active) directions intact. □

Corollary H.2 (Semantic Stability)**.** *A scale λ^* defines a stable semantic representation if* $\ker(d\Phi_{\lambda^*}) = \ker(d\Phi_{\lambda^*+\varepsilon})$.

Proof. At stability, no new directions are collapsed under further projection, so the quotient space remains unchanged up to isomorphism. □

I. Functorial Projection and Lyapunov Stability

I.1. Semantic Projection Systems

Definition I.1. A *semantic projection system* is a triple $(\mathbf{Traj}, \mathbf{Sem}, \mathscr{F})$ where **Traj** is a symmetric monoidal category, **Sem** is a category of semantic representations, and $\mathscr{F} : \mathbf{Traj} \to \mathbf{Sem}$ is a functor.

Definition I.2. The system exhibits *controlled abstraction* if $\mathscr{F}$ is faithful on a subcategory $\mathbf{C} \subset \mathbf{Traj}$ corresponding to structurally relevant trajectories.

Definition I.3. The system exhibits *collapse* if $\mathscr{F}$ fails to be faithful on $\mathbf{C}$.

Theorem I.4. *Let $\{\mathscr{F}_\lambda\}$ be a family of projection functors indexed by scale. If there exists λ^* such that $\mathscr{F}_{\lambda^*} \cong \mathscr{F}_{\lambda^*+\varepsilon}$, then λ^* defines a stable semantic resolution.*

Proof. At λ^*, the functor's image is invariant under further refinement, so no new identifications occur and the quotient structure stabilizes. □

I.2. Generator–Functor Correspondence

Definition I.5. A semantic projection functor is *generated by* $\mathcal{L}$ if $\mathcal{F} = \pi_\lambda \circ e^{T\mathcal{L}} \circ \Phi \circ \Psi$, where $e^{T\mathcal{L}}$ denotes the time-T flow of $\partial_t \Phi = \mathcal{L}[\Phi]$.

Theorem I.6 (Generator–Functor Correspondence)**.** *Every semantic projection functor arising from a continuous evolution process can be expressed as the composition of a realization map Φ, a flow generated by $\mathcal{L}$, and a projection π_λ.*

Proof. Given a trajectory τ, its evolution defines a path $\Phi(t)$ in embedding space. By the definition of a continuous dynamical system, this path is generated by some operator $\mathcal{L}$. The final semantic representation is $\mathcal{F}(\tau) = \pi_\lambda(e^{T\mathcal{L}}\Phi(\Psi(\tau)))$, yielding the decomposition. □

Corollary I.7 (Stability Criterion)**.** *A semantic representation is stable if $\mathcal{L}[\Phi] = 0$.*

Corollary I.8 (Collapse Criterion)**.** *Collapse occurs when $\mathcal{L}$ reduces variance along directions transverse to semantic fibers.*

I.3. Proof of Lyapunov Decrease

This appendix provides the full proof of Theorem 11.1.

Let $\Phi(x, t)$ evolve under

$$\partial_t \Phi = -\nabla \cdot (\Phi v) + D\nabla^2 \Phi - \lambda \frac{\delta \mathcal{J}}{\delta \Phi},$$

and define $\mathcal{E}[\Phi] = \int \Phi \log \Phi \, dx + \alpha \int \|\nabla \Phi\|^2 dx + \beta \mathcal{J}[\Phi]$.

Computing $\frac{d}{dt}\mathcal{E}$: the variation is $\frac{d\mathcal{E}}{dt} = \int \frac{\delta \mathcal{E}}{\delta \Phi} \partial_t \Phi \, dx$. The diffusion contribution is

$$\int (1 + \log \Phi) D\nabla^2 \Phi \, dx = -D \int \frac{\|\nabla \Phi\|^2}{\Phi} dx \leq 0,$$

using integration by parts and positivity of Φ. The gradient term gives

$$2\alpha \int \nabla \Phi \cdot \nabla(\partial_t \Phi) \, dx = -2\alpha \int \|\nabla^2 \Phi\|^2 dx \leq 0.$$

The variational forcing term gives

$$\beta \int \frac{\delta \mathcal{J}}{\delta \Phi} \left(-\lambda \frac{\delta \mathcal{J}}{\delta \Phi}\right) dx = -\beta\lambda \left\| \frac{\delta \mathcal{J}}{\delta \Phi} \right\|_{L^2}^2 \leq 0.$$

The advection term vanishes under mass-preserving boundary conditions: $\int (1 + \log \Phi)(-\nabla \cdot (\Phi v)) dx = 0$. All contributions are non-positive, so $\frac{d}{dt}\mathcal{E} \leq 0$. A completed work therefore corresponds to a local minimum of $\mathcal{E}$: the fixed point toward which the constrained dynamics inevitably evolve. □

References

[1] Rudolf Arnheim, *Art and Visual Perception: A Psychology of the Creative Eye*, University of California Press, 1974.

[2] Alain Badiou, *Being and Event*, Continuum, 2005. Translated by Oliver Feltham.

[3] Gregory J. Chaitin, On the length of programs for computing finite binary sequences, *Journal of the ACM*, 13(4):547–569, 1966.

[4] Gilles Deleuze, *Difference and Repetition*, Columbia University Press, 1994. Translated by Paul Patton.

[5] Sam Denham, *Semantic Collapse in Embedding Space: Neighborhood Semantics and Entropic Drift*, Manuscript, 2025.

[6] E.H. Gombrich, *Art and Illusion: A Study in the Psychology of Pictorial Representation*, Phaidon, 1960.

[7] Douglas R. Hofstadter, *Gödel, Escher, Bach: An Eternal Golden Braid*, Basic Books, 1979.

[8] E.T. Jaynes, *Probability Theory: The Logic of Science*, Cambridge University Press, 2003.

[9] Andrei N. Kolmogorov, Three approaches to the quantitative definition of information, *Problems of Information Transmission*, 1(1):1–7, 1965.

[10] Saunders Mac Lane, *Categories for the Working Mathematician*, Springer, 2nd edition, 1998.

[11] Marvin Minsky, *The Society of Mind*, Simon and Schuster, 1986.

[12] Charles Sanders Peirce, *Collected Papers of Charles Sanders Peirce*, Harvard University Press, 1931. Edited by Charles Hartshorne and Paul Weiss.

[13] Wilfrid Sellars, *Science, Perception and Reality*, Routledge and Kegan Paul, 1963.

[14] Claude E. Shannon and Warren Weaver, *The Mathematical Theory of Communication*, University of Illinois Press, 1949.

[15] Herbert A. Simon, *The Sciences of the Artificial*, MIT Press, 3rd edition, 1996.

[16] Yandong Sun, Qiang Huang, Ziwei Xu, Yiqun Sun, Yixuan Tang, and Anthony K.H. Tung, *One Swallow Does Not Make a Summer: Understanding Semantic Structures in Embedding Spaces*, Preprint, arXiv:2512.00852 [cs.AI], 2025.

[17] Alan M. Turing, Computing machinery and intelligence, *Mind*, 59(236):433–460, 1950.

[18] Ludwig Wittgenstein, *Philosophical Investigations*, Blackwell, 1953. Translated by G.E.M. Anscombe.