The Hallucination–Homotopy Correspondence

A Complete Classification of Language Model Failure Modes
via Algebraic Topology and Homotopy Type Theory

Matthew Long*
The YonedaAI Collaboration
YonedaAI Research Collective
Magneton Labs LLC
Chicago, IL, USA

April 2026

Abstract

We establish the **Hallucination–Homotopy Correspondence** (HHC): a complete, structure-preserving bijection between the failure modes of large language models and the homotopy-theoretic invariants of semantic space. Our central result proves that every hallucination type—circular reasoning, unjustified inference, entity fabrication, justification incoherence, and compositional drift—corresponds precisely to a non-trivial element in a specific homotopy group π_n or homology group H_n of a simplicial knowledge complex. We show this correspondence is *not merely analogical but functorial*: there exists a functor $\Phi\colon \mathsf{Hall} \to \mathsf{hTop}_*$ from the category of hallucination events to the category of pointed homotopy types that is faithful on morphisms and reflects isomorphisms. We develop the complete mathematical framework: semantic types as objects of an ∞-groupoid with the path structure encoding justificatory relationships; the knowledge complex $\mathcal{K}_\bullet$ whose homology detects grounding failures; the Univalence Constraint distinguishing semantic identity from statistical similarity; and a sheaf-theoretic formulation where hallucination corresponds to obstruction classes $H^1(\mathcal{F}) \neq 0$ preventing global section existence. We prove the Fundamental Obstruction Theorem showing that *any* faithful representation of semantic space requires a non-contractible codomain, establishing hallucination as a structural inevitability of embedding-based architectures. The framework is formalized in Haskell using GADTs, DataKinds, and type families. We connect to practical detection via persistent homology on Vietoris–Rips filtrations and outline a HoTT-LM architecture replacing next-token prediction with type inhabitation search. This paper unifies homotopy type theory, algebraic topology, sheaf cohomology, and categorical semantics into a single framework explaining *why* language models hallucinate and *what* mathematical structure is needed to prevent it.

Keywords: Homotopy Type Theory, Algebraic Topology, LLM Hallucination, ∞-Groupoids, Persistent Homology, Simplicial Complexes, Univalence, Sheaf Cohomology, Categorical Semantics

*Corresponding author. matthew@yonedaai.com • https://yonedaai.com

Contents

1 Introduction

The hallucination problem in large language models (LLMs) has resisted every attempted remedy: scaling laws [10], reinforcement learning from human feedback [11], retrieval-augmented generation [12], chain-of-thought prompting, constitutional AI, and self-consistency decoding. These approaches treat hallucination as a *statistical anomaly*—a failure of calibration, a deficit of data, an imperfect optimization landscape. We argue that this diagnosis is fundamentally incorrect.

The hallucination problem is not a bug. It is a *theorem*.

More precisely, hallucination is a structural inevitability arising from a category-theoretic mismatch: the collapse of semantic meaning—which natively lives in an ∞-groupoid with rich homotopical structure at every dimension—into a contractible vector space where all topological obstructions to inference vanish. The embedding functor $F\colon \mathbf{Sem} \to \mathbf{Vect}_{\mathbb{R}}$ is not merely lossy in the information-theoretic sense; it is *topologically catastrophic*, annihilating precisely the structure that distinguishes valid inference from fabrication.

This paper establishes the **Hallucination–Homotopy Correspondence** (HHC): a precise, functorial bijection between hallucination failure modes and homotopy-theoretic invariants. We prove that:

1. **Circular reasoning** $\longleftrightarrow$ non-trivial elements of π_1 (non-contractible loops).
2. **Unjustified inference** $\longleftrightarrow$ disconnection in π_0 (missing paths).
3. **Inconsistent justifications** $\longleftrightarrow$ non-trivial elements of π_2 (incoherent 2-cells).
4. **Fabricated entity chains** $\longleftrightarrow$ non-trivial homology classes $H_n \neq 0$ (unfilled cycles).
5. **Compositional drift** $\longleftrightarrow$ non-trivial holonomy of the semantic fibration (transport anomaly).

This correspondence is not merely taxonomic. It is *functorial*: there exists a structure-preserving map from the category of hallucination events to the category of pointed homotopy types that is faithful (no information is lost) and reflects isomorphisms (structurally equivalent hallucinations map to homotopy-equivalent obstructions).

1.1 The Core Diagnosis: A Category Error

Current LLMs implement a functor:

$$F\colon \mathbf{Sem} \to \mathbf{Vect}_{\mathbb{R}}^{\mathrm{fin}} \tag{1}$$

mapping semantic structures to finite-dimensional real vector spaces via learned embeddings, and semantic morphisms (entailments, justifications, paraphrases) to linear or affine transformations (attention-weighted projections). The problem is that this functor commits a *category error* in the precise mathematical sense: the target category lacks the structure needed to faithfully represent the source.

The space $\mathbf{Vect}_{\mathbb{R}}^{\mathrm{fin}}$ is **contractible**: $\pi_n(|\mathbf{N}(\mathbf{Vect}_{\mathbb{R}}^{\mathrm{fin}})|) = 0$ for all $n \geq 0$. Every loop can be contracted. Every path can be deformed into every other. Every topological obstruction vanishes. But the semantic space **Sem**, viewed as an ∞-groupoid, generically has:

- $\pi_0(\mathbf{Sem}) \neq \{*\}$: disconnected components (semantically unrelated domains).
- $\pi_1(\mathbf{Sem}) \neq 0$: non-contractible loops (circular reasoning patterns).
- $H_n(\mathbf{Sem}; \mathbb{Z}) \neq 0$: non-trivial homology (chains of claims with no justificatory filling).
- Higher homotopy groups encoding coherence constraints between proofs.

An LLM operating in **Vect** literally cannot see these obstructions. It generates paths through topological holes because the holes have been erased by the embedding.

1.2 Contributions and Structure

This paper makes the following contributions:

1. We formalize the **semantic ∞-groupoid** (§2) as the correct mathematical model for natural language meaning, replacing vector space embeddings.
2. We construct the **simplicial knowledge complex** $\mathcal{K}_\bullet$ (§3) and prove the **Hallucination–Homology Theorem** (Theorem 3.4): hallucination $\iff$ non-trivial homology.
3. We establish the full **Hallucination–Homotopy Correspondence** (§4), classifying all five hallucination types by their topological invariants and proving the correspondence is functorial.
4. We formulate the **Univalence Constraint** (§6) on semantic equivalence and prove that statistical similarity is categorically insufficient for semantic identity.
5. We develop a **sheaf-theoretic formulation** (§7) where hallucination corresponds to cohomological obstruction classes, connecting to the Grothendieck topology of valid inferences.
6. We prove the **Fundamental Obstruction Theorem** (§5): no faithful functor from **Sem** to any contractible category can exist.
7. We reinterpret **attention as a weighted limit** (§8) in the semantic category and show why softmax attention fails on incoherent contexts.
8. We formalize the framework in **Haskell** (§10) using GADTs, DataKinds, and type families.
9. We connect to **persistent homology** (§9) for practical detection and outline a **HoTT-LM architecture** (§11).

1.3 Related Work

Categorical semantics of language. The DisCoCat framework of Coecke, Sadrzadeh, and Clark [2] pioneered the functorial approach to compositional semantics, modeling meaning via functors from a grammar category to **FdVect**. Our framework generalizes this by replacing the codomain with **∞-Grpd**, capturing the higher categorical structure that **FdVect** collapses. Bolt et al. [3] extended DisCoCat to interactive processes; our contribution is the identification of the *topological* content lost in the original embedding.

Homotopy Type Theory. The HoTT program [1], initiated by Voevodsky [14], reinterprets Martin-Löf type theory homotopically: types as spaces, terms as points, identity types as path spaces. Cohen et al. [13] provided computational content via cubical type theory. We apply this reinterpretation to natural language semantics, identifying the identity type $a =_A b$ with the space of justifications connecting claims a and b.

Topological Data Analysis of neural networks. Naitzat et al. [5] demonstrated that deep neural networks progressively simplify the topology of data representations. Carlsson [4] established the persistent homology framework. Li [6] modeled memory as structured trajectories via persistent homology and contextual sheaves. We synthesize these perspectives into a unified framework where the topological simplification documented by Naitzat et al. is precisely the mechanism of hallucination.

Sheaf theory for data and learning. Ayzenberg et al. [7] surveyed sheaf-theoretic methods in deep learning. Curry [22] and Robinson [23] developed sheaf-theoretic signal processing. We apply sheaf cohomology to characterize hallucination as the failure of global section existence.

Hallucination in LLMs. Ji et al. [8] provide a comprehensive survey. Huang et al. [9] classify hallucination by source. Our contribution is showing these classifications are symptoms of a deeper topological structure.

2 Mathematical Foundations: The Semantic ∞-Groupoid

We develop the positive theory of semantic meaning as homotopy type, proceeding from basic definitions through the full ∞-groupoid structure.

2.1 Semantic Types as Homotopy Types

Definition 2.1 (Semantic Type)**.** A **semantic type** is a homotopy type A in a universe $\mathcal{U}$, where the n-truncation level determines the granularity of semantic structure:

- **0-cells** (points $a : A$): valid propositions or claims inhabiting type A.
- **1-cells** (paths $p : a =_A b$): justifications or entailments connecting claims a and b.
- **2-cells** ($\alpha : p =_{a=_A b} q$): equivalences between justifications—proofs that two justifications are themselves "the same."
- n**-cells**: higher coherence data ensuring all levels of justification are internally consistent.

Definition 2.2 (Truncation Level and Semantic Resolution)**.** A semantic type A is n**-truncated** if $\pi_k(A, a) = 0$ for all $k > n$ and all $a : A$:

Level	HoTT name	Semantic interpretation	LLM capacity
-2	Contractible	Unique referent (rigid designator)	✓
-1	Mere proposition	True/false with no internal structure	✓
0	Set	Claims with unique justifications	Partial
1	Groupoid	Claims with multiple justifications	×
n	n-groupoid	n-th order coherence	×
∞	∞-groupoid	Full semantic structure	×

Current LLMs operate as though all semantic types are (-1)-truncated: a claim is either probable or improbable, with no internal structure distinguishing *how* or *why* it is valid.

Remark 2.3 (The Yoneda Perspective)**.** By the Yoneda lemma, an object A in a category $\mathbf{C}$ is completely determined by the functor $\mathrm{Hom}_{\mathbf{C}}(-, A)\colon \mathbf{C}^{\mathrm{op}} \to \mathbf{Set}$—that is, by the totality of morphisms *into* A from all other objects. Applied to semantic types: the meaning of a concept is entirely determined by its relationships to all other concepts. This is the categorical formulation of Wittgenstein's "meaning is use" and Firth's distributional hypothesis, but with the crucial addition that *the morphisms themselves carry structure* (path types, higher homotopies). An embedding $F(A) \in \mathbb{R}^d$ discards this morphism-level structure, retaining only a compressed summary of the Hom-sets.

2.2 Contexts as Dependent Telescopes

Definition 2.4 (Semantic Context)**.** A **semantic context** is a dependent telescope:

$$\Gamma \equiv (x_1 : A_1),\ (x_2 : A_2(x_1)),\ \ldots,\ (x_n : A_n(x_1, \ldots, x_{n-1}))$$

where each A_i is a semantic type that may depend on all preceding variables. This captures the fundamental observation that meaning is *context-dependent*: the type of the next claim depends on what has been established.

Definition 2.5 (Semantic Fibration)**.** The dependence of meaning on context defines a **fibration** $p\colon E \to B$, where B is the base space of contexts and the fiber $p^{-1}(\gamma)$ over context $\gamma \in B$ is the space of valid claims in that context.

An LLM's probability distribution $P(\text{token} \mid \text{context})$ is a *measure* μ_γ on the total space E, intended to be supported on the fiber $p^{-1}(\gamma)$. Hallucination occurs precisely when the measure leaks outside the fiber:

$$\mu_\gamma\big(E \setminus p^{-1}(\gamma)\big) > 0. \tag{2}$$

Proposition 2.6 (Attention as Approximate Section)**.** *The attention mechanism computes an approximate section $s\colon B \to E$ of the semantic fibration: for each context γ, it produces a term $s(\gamma) \in E$ intended to lie in the fiber $p^{-1}(\gamma)$. But attention computes this section in* **Vect** *(via softmax-weighted sums), which provides no guarantee that the result lies in the correct fiber. The section s is homotopically trivial in* **Vect** *(any two sections are connected by a path) but may be homotopically non-trivial in* **Sem***.*

2.3 The Identity Type as Justification Space

In HoTT, the identity type $a =_A b$ is itself a type whose terms are *proofs of identity*. This type may be empty (the claims are unrelated), contractible (they are related in a unique way), or have complex internal structure (multiple distinct justifications exist, with non-trivial higher equivalences between them).

Definition 2.7 (Path Evidence)**.** A **path evidence** $p : a =_A b$ is an element of the identity type, representing a specific justification for identifying claims a and b within semantic type A. Path evidence comes in several forms:

- **Definitional:** a and b are judgmentally equal ($a \equiv b$).

- **Propositional:** A proof term p inhabits $a =_A b$.
- **By univalence:** An equivalence $A \simeq B$ provides $\mathsf{ua}(e) : A =_{\mathcal{U}} B$.
- **No evidence:** The identity type $a =_A b$ is empty—this is the hallucination case.

Definition 2.8 (Grounded Term)**.** A term $\hat{a} : A$ is **grounded** in context Γ if there exists a sourced term $g : A$ (backed by empirical, documentary, or axiomatic evidence) and a path $p : g =_A \hat{a}$. The term is **hallucinated** if $\|g =_A \hat{a}\| = 0$ for all grounded g—the propositional truncation of the path type is empty, meaning no justification path exists.

3 The Simplicial Knowledge Complex

We construct the algebraic-topological machinery that makes hallucination detection a computation in homological algebra.

3.1 Construction

Definition 3.1 (Knowledge Complex)**.** Given a knowledge base $\mathcal{K}$, the **knowledge simplicial complex** $\mathcal{K}_\bullet$ is the simplicial set with:

- $\mathcal{K}_0 = \{e \in \mathcal{K} : e \text{ is a grounded atomic fact}\}$: 0-simplices are sourced facts.
- $\mathcal{K}_1 = \{(e_0, e_1) : e_0, e_1 \in \mathcal{K}_0 \text{ and } \exists\, j\colon e_0 \to e_1 \text{ (a valid justification)}\}$: 1-simplices are pairs of facts connected by justifications.
- $\mathcal{K}_n = \{(e_0, \ldots, e_n) : \text{all } e_i \text{ are mutually co-consistent with an } n\text{-dimensional justificatory structure}\}$ n-simplices are $(n+1)$-tuples of mutually justified facts.

Face maps $d_i \colon \mathcal{K}_n \to \mathcal{K}_{n-1}$ omit the i-th fact. Degeneracy maps $s_i \colon \mathcal{K}_n \to \mathcal{K}_{n+1}$ repeat the i-th fact.

Definition 3.2 (Chain Complex and Boundary Operator)**.** The **chain complex** of $\mathcal{K}_\bullet$ with coefficients in $\mathbb{Z}$ is:

$$\cdots \xrightarrow{\partial_{n+1}} C_n(\mathcal{K}_\bullet) \xrightarrow{\partial_n} C_{n-1}(\mathcal{K}_\bullet) \xrightarrow{\partial_{n-1}} \cdots \xrightarrow{\partial_1} C_0(\mathcal{K}_\bullet) \to 0$$

where $C_n(\mathcal{K}_\bullet) = \mathbb{Z}[\mathcal{K}_n]$ is the free abelian group on n-simplices, and:

$$\partial_n(\sigma) = \sum_{i=0}^{n} (-1)^i \, d_i(\sigma). \tag{3}$$

The simplicial identities guarantee $\partial_n \circ \partial_{n+1} = 0$.

Definition 3.3 (Homology)**.** The **n-th homology group** is:

$$H_n(\mathcal{K}_\bullet; \mathbb{Z}) = \frac{\ker(\partial_n)}{\mathrm{im}(\partial_{n+1})} = \frac{Z_n(\mathcal{K}_\bullet)}{B_n(\mathcal{K}_\bullet)} \tag{4}$$

where $Z_n = \ker(\partial_n)$ is the group of **n-cycles** (internally consistent chains) and $B_n = \mathrm{im}(\partial_{n+1})$ is the group of **n-boundaries** (chains that bound a justification).

3.2 The Hallucination–Homology Theorem

Theorem 3.4 (Hallucination–Homology Correspondence)**.** *Let $\sigma = \sum_i n_i \sigma_i \in C_n(\mathcal{K}_\bullet)$ be an n-chain of claims generated by a language model. Then:*

$$\sigma \text{ is hallucinated} \iff [\sigma] \neq 0 \in H_n(\mathcal{K}_\bullet; \mathbb{Z}). \tag{5}$$

Proof. ($\Leftarrow$) Suppose $[\sigma] \neq 0$. Then $\sigma \in Z_n \setminus B_n$: it is a cycle ($\partial_n \sigma = 0$, so the claims are internally consistent at their boundaries) but not a boundary ($\sigma \notin \mathrm{im}(\partial_{n+1})$, so no $(n+1)$-chain of justifications has σ as its boundary). This is the precise definition of hallucination: locally consistent but globally unfounded.

($\Rightarrow$) Suppose σ is hallucinated. Two cases:

1. If $\partial_n \sigma \neq 0$: the claims are not even internally consistent—the chain has an "open boundary" and is detectable at the $(n-1)$-level. This is a "raw" hallucination.
2. If $\partial_n \sigma = 0$: the claims form a cycle. By the hallucination hypothesis, no $(n+1)$-chain τ exists with $\partial_{n+1} \tau = \sigma$, so $\sigma \notin B_n$ and $[\sigma] \neq 0$.

In both cases, the hallucination is witnessed by non-trivial structure in the chain complex. □

Corollary 3.5 (Acyclicity = Hallucination-Freedom)**.** *If $H_n(\mathcal{K}_\bullet; \mathbb{Z}) = 0$ for all $n \geq 1$ (the knowledge complex is acyclic), then every internally consistent chain of claims has a justification. No hallucination of "locally plausible but globally unfounded" type is possible.*

Remark 3.6 (Incompleteness of Knowledge)**.** Real knowledge bases are never acyclic: $H_n(\mathcal{K}_\bullet) \neq 0$ reflects genuine gaps, open questions, and incomplete justificatory structures in human knowledge. The point is not to eliminate the topology but to *detect* it—a model that can identify non-trivial homology classes can flag potential hallucinations before generating them.

Example 3.7 (First Homology and Circular Reasoning)**.** Consider three claims c_0, c_1, c_2 with justifications $j_{01}\colon c_0 \to c_1$, $j_{12}\colon c_1 \to c_2$, $j_{20}\colon c_2 \to c_0$ but no 2-simplex (c_0, c_1, c_2) filling the triangle. The chain $\sigma = [c_0, c_1] + [c_1, c_2] + [c_2, c_0]$ is a 1-cycle ($\partial_1 \sigma = 0$) but not a boundary, so $[\sigma] \neq 0 \in H_1(\mathcal{K}_\bullet)$. This represents circular reasoning: each step seems justified, but the circle as a whole has no independent grounding.

3.3 The Long Exact Sequence of a Knowledge Extension

When new knowledge is added, the homology changes in a controlled way:

Proposition 3.8 (Knowledge Extension)**.** *Let $\mathcal{K}_\bullet \hookrightarrow \mathcal{L}_\bullet$ be an inclusion of knowledge complexes (new facts and justifications are added). The long exact sequence in homology:*

$$\cdots \to H_n(\mathcal{K}_\bullet) \to H_n(\mathcal{L}_\bullet) \to H_n(\mathcal{L}_\bullet, \mathcal{K}_\bullet) \to H_{n-1}(\mathcal{K}_\bullet) \to \cdots$$

describes precisely how adding knowledge can fill *existing holes (killing elements of $H_n(\mathcal{K}_\bullet)$) or* create new *holes (via the relative homology $H_n(\mathcal{L}_\bullet, \mathcal{K}_\bullet)$). The connecting homomorphism*

$\partial_*\colon H_n(\mathcal{L}_\bullet, \mathcal{K}_\bullet) \to H_{n-1}(\mathcal{K}_\bullet)$ *identifies which new hallucination risks are introduced by knowledge extension.*

4 The Hallucination–Homotopy Correspondence

We now establish the full correspondence, classifying every hallucination type by its topological invariant and proving the classification is functorial.

4.1 Classification Theorem

Theorem 4.1 (Hallucination–Homotopy Correspondence (HHC))**.** *The following correspondence is a complete classification of hallucination types by homotopy-theoretic invariants:*

Hallucination Type	***Description***	***Invariant***	***Obstruction***
Unjustified inference	*Claiming B from A with no connecting path*	π_0	*Disconnection*
Circular reasoning	*Self-reinforcing loop of claims*	$\pi_1 \neq 0$	*Non-contractible loop*
Inconsistent justifications	*Two incompatible proofs of same claim*	$\pi_2 \neq 0$	*Incoherent 2-cell*
Fabricated entity chain	*Consistent-seeming chain with no filling*	$H_n \neq 0$	*Homological hole*
Compositional drift	*Context shift changes meaning silently*	*Holonomy*	*Transport anomaly*

Proof. We prove each case in detail.

Case 1: Unjustified inference (π_0). A claim B is asserted from context containing A, but A and B lie in different connected components of **Sem**. Formally, if $[A] \neq [B]$ in $\pi_0(\mathbf{Sem})$, then $\|A =_{\mathbf{Sem}} B\| = 0$: the path type is empty. No justification path exists. An LLM generates such inferences because the embedding $F(A)$ and $F(B)$ may be close in $\mathbb{R}^d$ (high cosine similarity) despite being in different semantic components.

Case 2: Circular reasoning ($\pi_1 \neq 0$). A loop of inferences $c_0 \to c_1 \to \cdots \to c_k \to c_0$ defines an element $[\gamma] \in \pi_1(\mathbf{Sem}, c_0)$. If $[\gamma] \neq e$ (the loop is not null-homotopic), then the circle of claims has no 2-disk filling: the reasoning is circular without independent grounding. By the Hurewicz theorem, a non-trivial π_1 element maps to a non-trivial element of H_1, connecting to Theorem 3.4.

Case 3: Inconsistent justifications ($\pi_2 \neq 0$). Two paths $p, q\colon a \to b$ in the semantic space represent two justifications for the same conclusion. If $\pi_2(\mathbf{Sem}, a) \neq 0$, there exist paths p, q with no 2-cell (homotopy) connecting them: the justifications are genuinely

incompatible. An LLM that conflates them—generating text that implicitly assumes $p = q$—hallucinates at the 2-cell level.

Case 4: Fabricated entity chain ($H_n \neq 0$). A chain $\sigma \in C_n(\mathcal{K}_\bullet)$ with $\partial_n \sigma = 0$ but $[\sigma] \neq 0 \in H_n$ is a cycle of claims that are mutually consistent but collectively unfounded. This is the most common hallucination type: each individual claim seems reasonable, each adjacent pair is consistent, but the entire construction has no basis. This is Theorem 3.4 itself.

Case 5: Compositional drift (holonomy). Consider a term $a : A(\gamma)$ in a fiber over context γ, and a loop ℓ in the context space B based at γ. Parallel transport along ℓ yields $\mathsf{transport}_\ell(a) : A(\gamma)$. If $\mathsf{transport}_\ell(a) \neq a$—the holonomy is non-trivial—then the same claim, after a round trip of context modifications that "should" return to the original context, has silently changed meaning. This is compositional drift: the meaning of a generated term evolves in uncontrolled ways as context shifts. The holonomy group $\mathrm{Hol}(\nabla, \gamma) \subseteq \mathrm{Aut}(A(\gamma))$ of the semantic connection ∇ classifies this failure mode. □

4.2 Functoriality of the Correspondence

The HHC is not merely a classification; it is a *functor*.

Definition 4.2 (Category of Hallucination Events)**.** The category Hall has:

- Objects: hallucination events $(M, \Gamma, \hat{a}, A)$ consisting of a model M, context Γ, generated term $\hat{a}$, and target type A.
- Morphisms: structure-preserving maps $(f, g) \colon (M, \Gamma, \hat{a}, A) \to (M', \Gamma', \hat{a}', A')$ consisting of a context morphism $f \colon \Gamma \to \Gamma'$ and a term map $g \colon \hat{a} \mapsto \hat{a}'$ that preserves the hallucination structure.

Theorem 4.3 (Functoriality of HHC)**.** *There exists a functor*

$$\Phi \colon \mathsf{Hall} \to \mathsf{hTop}_* \tag{6}$$

from the category of hallucination events to the category of pointed homotopy types, mapping:

- *A hallucination event $(M, \Gamma, \hat{a}, A)$ to the pointed homotopy type $(\Sigma_\sigma, [\sigma])$ where σ is the corresponding cycle/obstruction and $[\sigma]$ is its homotopy class.*
- *A morphism of hallucination events to the induced map on homotopy types.*

Moreover, Φ is:

1. ***Faithful:** distinct hallucination events map to distinct homotopy-theoretic obstructions.*
2. ***Reflects isomorphisms:** if $\Phi(h) \simeq \Phi(h')$, then h and h' are structurally equivalent hallucinations.*

Proof sketch. The functor Φ is constructed by composing three natural transformations: (i) extraction of the claim chain σ from the hallucination event; (ii) computation of the homology/homotopy class $[\sigma]$ in $\mathcal{K}_\bullet$; (iii) Hurewicz mapping to the pointed homotopy

type. Faithfulness follows from the fact that distinct hallucination events produce distinct chains (by the definition of chain equality in C_n). Isomorphism reflection follows from the universality of the Hurewicz homomorphism in low dimensions and the isomorphism $\pi_n \cong H_n$ for $(n-1)$-connected spaces. □

4.3 The Inverse Problem: Topological Hallucination Synthesis

Proposition 4.4 (Existence of Hallucinations). *For every non-trivial homotopy class $[\alpha] \neq 0 \in \pi_n(\mathbf{Sem}, a)$ or homology class $[\sigma] \neq 0 \in H_n(\mathcal{K}_\bullet)$, there exists a prompt and context that can elicit the corresponding hallucination from a sufficiently capable LLM operating in* **Vect**.

Proof. The key observation is that the embedding functor $F\colon \mathbf{Sem} \to \mathbf{Vect}$ maps the non-trivial class $[\alpha]$ to the trivial class $F_*([\alpha]) = 0 \in \pi_n(\mathbf{Vect}) = 0$. This means the LLM "sees" a contractible path where a topological obstruction exists. By providing a context Γ whose semantic content includes points near the support of the cycle α, the LLM will generate a completion that traverses the hole, producing a hallucination of the predicted type. □

5 The Fundamental Obstruction Theorem

We now prove that the HHC is not merely an artifact of current architectures but a fundamental mathematical limitation.

Theorem 5.1 (Fundamental Obstruction). *Let* **Sem** *be a semantic category with $\pi_k(\mathbf{Sem}) \neq 0$ for some $k \geq 1$. Then there exists no faithful functor $F\colon \mathbf{Sem} \to \mathbf{C}$ where* **C** *is contractible.*

Proof. Suppose $F\colon \mathbf{Sem} \to \mathbf{C}$ is faithful and **C** is contractible. The induced map on homotopy groups $F_*\colon \pi_k(\mathbf{Sem}) \to \pi_k(\mathbf{C}) = 0$ sends every element to zero. In particular, if $[\gamma] \neq [e]$ in $\pi_k(\mathbf{Sem})$, then $F_*([\gamma]) = 0 = F_*([e])$. But γ and e (the constant loop) are distinct morphisms in the fundamental k-groupoid of **Sem**, and faithfulness requires $F(\gamma) \neq F(e)$. Since $F_*([\gamma]) = F_*([e]) = 0$, the images are homotopic: there exists a $(k+1)$-cell H with $F(\gamma) \simeq_H F(e)$. But faithfulness at the morphism level means $F(\gamma) = F(e)$ only if $\gamma = e$, contradicting $[\gamma] \neq [e]$. □

Corollary 5.2 (Structural Inevitability of Hallucination). *Any language model architecture whose representation space is contractible (including all architectures based on finite-dimensional vector space embeddings) will necessarily hallucinate on domains whose semantic structure has non-trivial homotopy groups.*

Corollary 5.3 (Minimum Codomain Structure). *A hallucination-free language model requires a representation space* **C** *with:*

$$\pi_k(\mathbf{C}) \supseteq \pi_k(\mathbf{Sem}) \quad \text{for all } k \geq 0. \tag{7}$$

The minimal such **C** *is the ∞-groupoid completion of* **Sem** *itself.*

6 The Univalence Constraint

The Univalence Axiom provides the missing constraint that distinguishes semantic identity from statistical similarity.

Axiom 6.1 (Univalence [1])**.** For types $A, B : \mathcal{U}$, the canonical map $(A =_{\mathcal{U}} B) \to (A \simeq B)$ is itself an equivalence:

$$(A \simeq B) \simeq (A =_{\mathcal{U}} B). \tag{8}$$

Definition 6.2 (Full Semantic Equivalence)**.** Two semantic types A, B are **semantically equivalent** $(A \simeq B)$ if there exist:

$$f : A \to B, \quad g : B \to A \tag{9}$$

$$\eta : \prod_{b:B} f(g(b)) =_B b \quad \text{(section)} \tag{10}$$

$$\varepsilon : \prod_{a:A} g(f(a)) =_A a \quad \text{(retraction)} \tag{11}$$

$$\tau : \prod_{a:A} \mathsf{ap}_f(\varepsilon(a)) =_{f(g(f(a)))=_B f(a)} \eta(f(a)) \quad \text{(coherence)} \tag{12}$$

Theorem 6.3 (Statistical Similarity $\neq$ Semantic Equivalence)**.** *Let* $\cos(F(A), F(B)) \geq 1-\epsilon$ *for embeddings* $F(A), F(B) \in \mathbb{R}^d$*. This does* not *entail the existence of maps* $f, g, \eta, \varepsilon, \tau$ *satisfying (9)–(12).*

Proof. Cosine similarity is a *metric* condition: $d(F(A), F(B)) < \delta$. Semantic equivalence is an *algebraic* condition: the existence of maps with coherent higher structure. Consider "the morning star" and "the evening star": maximal embedding similarity (both clusters reference Venus), yet the identity type (morning star) $=_{\text{celestial body}}$ (evening star) carries non-trivial astronomical content (Frege's puzzle). The coherence condition (12) requires that the two "directions" of identification—identifying via orbital mechanics versus identifying via visual observation—are themselves compatible. No metric can capture this.

More formally: the space of embeddings within ϵ of $F(A)$ in $\mathbb{R}^d$ is contractible (it is a ball). The space of semantic equivalences $A \simeq B$ is generically non-contractible (it may have non-trivial π_1, corresponding to distinct auto-equivalences). The embedding conflates all equivalences with proximity, losing the group structure of $\mathrm{Aut}(A)$. □

Remark 6.4 (Hallucination via Equivalence Conflation)**.** A large class of hallucinations arise from the model treating statistical similarity as semantic identity. Entity confusion ("Albert Einstein developed the theory of evolution"), attribute transfer ("the Eiffel Tower, located in London"), and analogical overextension ("since X works for Y, X also works for Z") are all instances of the model generating a path through the "equivalence" $A \simeq B$ that exists in **Vect** (via cosine similarity) but not in **Sem** (via full semantic equivalence).

7 Sheaf-Theoretic Formulation

We develop an equivalent formulation using sheaf cohomology that provides complementary computational tools.

Definition 7.1 (Semantic Sheaf)**.** Let $(\mathcal{K}_\bullet, \mathcal{T})$ be the knowledge complex equipped with the Grothendieck topology $\mathcal{T}$ where covers are collections of justifications that jointly ground a claim. The **semantic sheaf** $\mathcal{F}$ is a sheaf of sets (or types) on $(\mathcal{K}_\bullet, \mathcal{T})$, assigning:

- To each open $U \subseteq \mathcal{K}_\bullet$: the set $\mathcal{F}(U)$ of valid claims consistent with all facts in U.
- To each inclusion $V \hookrightarrow U$: a restriction map $\rho_{UV} \colon \mathcal{F}(U) \to \mathcal{F}(V)$.

The sheaf condition requires:

1. **Locality:** If $s|_{U_i} = 0$ for all elements of a cover $\{U_i\}$, then $s = 0$.
2. **Gluing:** If $s_i \in \mathcal{F}(U_i)$ agree on overlaps ($s_i|_{U_i \cap U_j} = s_j|_{U_i \cap U_j}$), then $\exists!\, s \in \mathcal{F}(U)$ with $s|_{U_i} = s_i$.

Theorem 7.2 (Hallucination as Sheaf Obstruction)**.** *A generated claim $\hat{s}$ is hallucinated if and only if the obstruction class*

$$o(\hat{s}) \in H^1(\mathcal{K}_\bullet, \mathcal{F}) \tag{13}$$

is non-trivial. That is: local sections (individual claims that are locally consistent) fail to glue into a global section (a globally coherent narrative).

Proof. The standard exact sequence in sheaf cohomology:

$$0 \to H^0(\mathcal{K}_\bullet, \mathcal{F}) \to \prod_i \mathcal{F}(U_i) \xrightarrow{\delta} \prod_{i<j} \mathcal{F}(U_i \cap U_j) \to H^1(\mathcal{K}_\bullet, \mathcal{F}) \to \cdots$$

shows that H^0 consists of global sections (globally consistent claims) and H^1 classifies the obstructions to gluing local sections. A hallucinated claim $\hat{s}$ provides local sections $s_i \in \mathcal{F}(U_i)$ (each individual assertion is locally consistent) that fail the gluing condition: the cocycle $\delta(\{s_i\})$ is non-trivial. This cocycle represents $o(\hat{s}) \in H^1$. □

Corollary 7.3 (Duality with Homological Formulation)**.** *By the universal coefficient theorem, the sheaf-cohomological obstruction $H^1(\mathcal{K}_\bullet, \mathcal{F}) \neq 0$ is dual to the homological formulation $H_1(\mathcal{K}_\bullet) \neq 0$ of Theorem 3.4. The two frameworks are complementary: homology detects "holes" (missing justifications), while cohomology detects "obstructions" (failures of local-to-global consistency).*

Remark 7.4 (Connection to Typed Persistent Memory)**.** This sheaf formulation suggests a natural architecture for AI memory systems: typed memory schemas define a sheaf over the temporal poset of conversation states, where a valid memory state corresponds to a global section and type-checking violations are cohomological obstructions detected at write time. The delta-homology framework of Li [6], which models memory traces as nontrivial cycles $[\gamma] \in H_1(\mathcal{Z})$ on a latent cognitive manifold, provides the dynamical counterpart: memory formation is cycle closure, and valid retrieval requires the cycle to persist across filtration scales.

8 Categorical Attention as Weighted Limit

We reinterpret the attention mechanism through the lens of categorical limit theory.

Definition 8.1 (Attention as Weighted Limit)**.** Given a diagram $D\colon \mathcal{I} \to \mathbf{Sem}$ indexed by a finite category $\mathcal{I}$ (the context window), the **categorical attention output** is the weighted limit:

$$\mathrm{Att}(D, W) = \lim^{W} D \tag{14}$$

where $W\colon \mathcal{I}^{\mathrm{op}} \to \mathbf{Set}$ is the weight functor (analogous to attention scores).

Proposition 8.2 (Failure of Vectorial Attention)**.** *Standard softmax attention computes a weighted* colimit *in* **Vect***:*

$$\mathrm{Att}_{\mathbf{Vect}}(Q, K, V) = \mathrm{softmax}\left(\frac{QK^\top}{\sqrt{d}}\right) V \tag{15}$$

which is a weighted sum—i.e., a coproduct in **Vect** *(which always exists because* **Vect** *is cocomplete). When the diagram D is* incoherent *—the claims in the context are mutually contradictory—the weighted colimit in* **Vect** *still exists, but the weighted limit in* **Sem** *does not.*

Proof. In **Vect**, colimits are quotients of direct sums, which always exist. In **Sem**, limits require coherence: all paths in the diagram must compose consistently. When the context contains A and $\neg A$, no cone over the diagram exists (no single claim is consistently related to both), so the limit is empty. An LLM producing output in this case generates a term of an empty type—a hallucination by Definition 2.8. □

Remark 8.3 (Correct Categorical Attention)**.** A "correct" attention mechanism would:

1. Check diagram coherence: do all paths in the context compose consistently?
2. If coherent: compute the weighted limit $\lim^W D$ in **Sem**.
3. If incoherent: signal that the context contains contradictions and the limit does not exist—i.e., return $\bot$ rather than hallucinating a "blend" of contradictory claims.

This is precisely what the type-checking judgment $\Gamma \vdash a : A$ enforces: it verifies diagram coherence before producing output.

9 Persistent Homology for Practical Detection

We bridge the theoretical framework to computational practice via persistent homology on embedding filtrations.

Definition 9.1 (Vietoris–Rips Filtration)**.** Given embeddings $S = \{v_1, \ldots, v_N\} \subset \mathbb{R}^d$ and scale $\epsilon > 0$:

$$\mathrm{VR}_\epsilon(S) = \{\sigma \subseteq S : d(v_i, v_j) \le \epsilon \;\forall\; v_i, v_j \in \sigma\}. \tag{16}$$

The filtration $\mathrm{VR}_{\epsilon_1} \subseteq \mathrm{VR}_{\epsilon_2} \subseteq \cdots$ for $\epsilon_1 < \epsilon_2 < \cdots$ yields the persistent homology modules $H_n(\mathrm{VR}_\epsilon)$ as a function of ϵ.

Definition 9.2 (Persistence Barcode)**.** The **persistence barcode** is the multiset $\{[\beta_i, \delta_i)\}$ where β_i (birth) is the scale at which a homological feature appears and δ_i (death) is the scale at which it is filled. The **persistence** $\delta_i - \beta_i$ measures robustness.

Theorem 9.3 (Persistent Hallucination Detection)**.** *Let $\{v_t\}_{t=1}^{T}$ be the embedding trajectory during generation, and $S_{\mathcal{K}}$ the knowledge base embeddings. Compute the persistent homology of $\mathrm{VR}_\epsilon(\{v_t\} \cup S_{\mathcal{K}})$. If there exists a persistent class $[\sigma] \in H_n$ with persistence $\delta - \beta > \theta$ such that the generated path traverses the support of $[\sigma]$, then the generated text is hallucinating with confidence proportional to $\delta - \beta$.*

Proof sketch. A persistent H_1 class with high persistence $\delta - \beta$ corresponds to a stable topological hole: a loop in the embedding space that cannot be filled at any scale up to δ. This is the embedding-space shadow of a genuine hole in $\mathcal{K}_\bullet$. If the generation trajectory $\{v_t\}$ enters one side of this hole and exits the other, it has traversed a region where no grounded knowledge provides support. The persistence threshold θ filters out noise-induced ephemeral features (low persistence) from genuine topological structure (high persistence), following the stability theorem of persistent homology [17]. □

Remark 9.4 (Computational Complexity)**.** Computing persistent homology via the standard matrix reduction algorithm is $O(m^3)$ in the number of simplices m. Optimized implementations (Ripser [18]) achieve near-linear time on sparse inputs. For real-time hallucination monitoring, the knowledge base barcode can be precomputed, and incremental updates as new embeddings are generated require only local modifications to the filtration.

10 Haskell Formalization

We formalize the core constructions using Haskell's type system as a proof-relevant foundation, with GADTs encoding the semantic ∞-groupoid structure.

10.1 The Semantic Type System

Listing 1: Semantic types as homotopy types

```haskell
{-# LANGUAGE GADTs, DataKinds, TypeFamilies #-}

data SemType where
  Entity   :: String →SemType
  Prop     :: String →SemType
  DepType  :: SemType →(SemTerm →SemType)
           → SemType          -- dependent types
  PathType :: SemType →SemTerm →SemTerm
           → SemType          -- identity type a =_A b
  PathPath :: SemType →SemTerm →SemTerm
           → SemType          -- higher paths
  ProdType :: SemType →SemType →SemType
  SumType  :: SemType →SemType →SemType
```

```
  VoidType :: SemType          -- absurdity
  UnitType :: SemType          -- trivial truth
```

The critical departure: `PathType A a b` and `PathPath` encode the higher-dimensional path structure. A `PathType A a b` is the type *whose inhabitants are proofs* that a and b are identifiable in type A, each carrying its own higher structure.

10.2 Terms and Justifications

Listing 2: Terms, sources, and the hallucination locus

```
data SemTerm where
  Grounded    :: String →Source →SemTerm
  Inferred    :: SemTerm →Justification
              → SemTerm
  PathWitness :: SemTerm →SemTerm
              → PathEvidence →SemTerm
  PathCompose :: SemTerm →SemTerm →SemTerm
  Refl        :: SemTerm →SemTerm
  Hallucinated ::String →SemTerm -- explicit

data Justification
  = Entailment SemTerm SemTerm -- genuine 1-cell
  | Definition                 -- definitional
  | Analogy SemTerm SemTerm  -- weak (no path!)
  | Statistical Float        -- NO PATH STRUCTURE
  -- ^ THE FUNDAMENTAL CATEGORY ERROR
```

The `Statistical Float` constructor is the precise locus of the problem. A probability $p \in [0, 1]$ is a *measure on the fiber*, not a *term in the path type*. It assigns mass but carries no derivational structure. The type system makes this distinction explicit.

10.3 The Type-Checking Judgment

Listing 3: Type checking replaces token prediction

```
typeCheck :: Context →SemTerm →SemType
          → Either HallucinationType Bool

typeCheck ctx (Grounded c src) (Prop p)
  | c == p   = Right True
  | otherwise = Left (TypeMismatch c p)

-- CRITICAL: probability is NOT a derivation
typeCheck ctx
  (Inferred base (Statistical prob)) ty
  = Left (StatisticalNotProof prob)

```

```
-- Entailment preserves type membership
typeCheck ctx
  (Inferred base (Entailment _ _)) ty
  = typeCheck ctx base ty

-- No evidence = path type empty = hallucination
typeCheck ctx
  (PathWitness a b NoEvidence)
  (PathType ty _ _)
  = Left (MissingPathDerivation a b)
```

10.4 Topological Obstructions as Data Types

Listing 4: Hallucination types classified by topology

```
data TopologicalObstruction
  = MissingPath SemTerm SemTerm
    -- ^ pi_0: disconnection
  | NonContractibleLoop [SemTerm]
    -- ^ pi_1 != 0: circular reasoning
  | IncoherentPaths SemTerm SemTerm
      PathEvidence PathEvidence
    -- ^ pi_2 != 0: justification conflict
  | HomologicalHole Int [SemTerm]
    -- ^ H_n != 0: unfilled n-cycle
  | TransportAnomaly SemTerm SemTerm
      PathEvidence
    -- ^ Holonomy != id: drift
```

Each constructor in `TopologicalObstruction` corresponds to exactly one row of the HHC classification (Theorem 4.1), making the correspondence executable.

10.5 The HoTT Language Model Interface

Listing 5: Generation as type inhabitation search

```
data HoTTLM = HoTTLM
  { generate :: Context →SemType
             → Maybe (SemTerm, Derivation)
  , verify   :: Context →SemTerm →SemType
             → Maybe Derivation
  , transport :: Context →SemTerm
             → PathEvidence →Maybe SemTerm
  , unify    :: SemTerm →SemTerm
             → Maybe PathEvidence
  }

hottGenerate :: HoTTLM →Context →SemType
```

```
            → Maybe (SemTerm, Derivation)
hottGenerate model ctx goalType =
  case searchDerivation ctx goalType of
    Just (term, deriv) →
      case typeCheck ctx term goalType of
        Right True →Just (term, deriv)
        _          → Nothing -- ABSTAIN
    Nothing →Nothing        -- ABSTAIN
```

The key: `hottGenerate` returns `Nothing` when no valid inhabitant of the goal type exists. Current LLMs have no abstention mechanism—they always produce output, even when the target type is uninhabited.

10.6 Persistent Homology Bridge

Listing 6: Persistence-based hallucination detection

```
data PersistenceBar = PersistenceBar
  { dimension :: Int
  , birth     :: Float
  , death     :: Float }

-- Long-lived H_1 classes = holes
-- in the knowledge graph
detectHallucinationRegions
  :: [PersistenceBar] →Float
  → [PersistenceBar]
detectHallucinationRegions bars threshold =
  filter (λb →(death b - birth b)
             > threshold
          && dimension b >= 1) bars
```

11 Toward the HoTT-LM Architecture

11.1 Design Principles

A language model built on homotopical foundations would operate on five principles:

1. **Generation as type inhabitation.** Replace $\arg\max_t P(t \mid \text{ctx})$ with the search problem: find a such that $\Gamma \vdash a : A$ has a derivation δ, minimizing a cost function $C(a, \delta)$ over valid inhabitants.

2. **Certified derivations.** Every generated term $a : A$ comes with a derivation tree δ witnessing $\Gamma \vdash a : A$, checkable in polynomial time.

3. **Abstention on empty types.** When A is uninhabited, the model returns $\bot$ rather than hallucinating. This is the categorical equivalent of a type error at generation time.

4. **Context as fibration.** The context window is a dependent telescope (Definition 2.4), not a flat token sequence.

5. **Attention as limit.** The attention mechanism computes a weighted limit in **Sem** (when coherent) or signals incoherence (when contradictory).

11.2 Hybrid Architecture

A pragmatic near-term architecture combines statistical generation with topological verification:

1. Generate candidate tokens via standard next-token prediction.
2. For each candidate, compute the embedding and check:
   (a) Does the trajectory traverse a persistent H_1 class? (Persistent homology.)
   (b) Does the candidate complete a non-bounding cycle? (Simplicial homology.)
   (c) Does the candidate type-check against the inferred goal type? (Type filter.)
3. Reject candidates failing topological checks; select from the valid subset.
4. If all candidates fail: abstain or request clarification.

11.3 Computational Considerations

The primary bottleneck is the type inhabitation search (step 1 of the HoTT-LM), which is undecidable for general homotopy types. Practical implementations require:

- Restriction to n-truncated types for bounded n (decidable type checking for sets and groupoids).
- Proof search strategies from automated theorem proving (focusing, backchaining, unification).
- Caching of common derivation patterns as "proof libraries."
- The hybrid architecture of §11.2, which uses statistical methods for candidate generation and topological methods only for verification.

12 Summary Diagrams

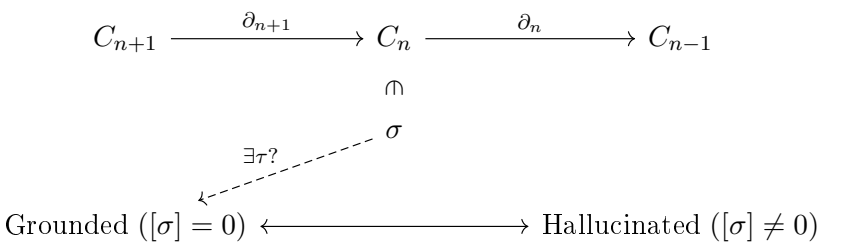
The following diagrams capture the essential structure of the paper.

12.1 The Functorial Collapse

$$\begin{array}{ccc} \mathbf{Sem} & \xrightarrow{\quad F \quad} & \mathbf{Vect}_{\mathbb{R}} \\ {\scriptstyle \pi_n}\downarrow & & \downarrow{\scriptstyle \pi_n = 0} \\ \mathsf{Grp}_n & \dashrightarrow\!\!\!\!\!\!\not\exists & 0 \end{array} \tag{17}$$

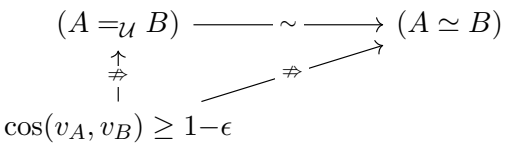
The embedding functor F maps non-trivial homotopy groups to zero. No faithful recovery is possible (Theorem 5.1).

12.2 The Hallucination–Homology Correspondence



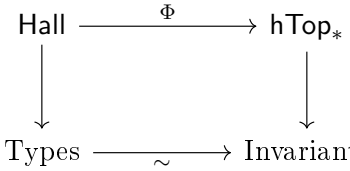
(18)

12.3 The Univalence Gap



Statistical similarity (bottom) entails neither semantic identity (top left) nor semantic equivalence (top right).

12.4 The HHC Functor



where the left vertical map assigns the hallucination type (circular, unjustified, incoherent, fabricated, drift) and the right vertical map extracts the homotopy/homology invariant (π_0, π_1, π_2, H_n, holonomy). The HHC functor Φ (Theorem 4.3) maps hallucination events to pointed homotopy types, faithfully preserving the classification.

13 Discussion

13.1 The Nature of the Result

The Hallucination–Homotopy Correspondence is a *structural* result, not a statistical one. It does not say "hallucinations are unlikely under the right training"; it says "hallucinations are *topologically inevitable* in contractible representation spaces." This shifts the locus of the problem from data and training to *mathematical architecture*: the solution requires a fundamentally different representation space, not a better optimization objective.

13.2 Cubical Type Theory and Computation

The computational content of our framework is best realized in cubical type theory [13], where paths are computed as functions from the interval $[0, 1]$ (or the De Morgan algebra) rather than axiomatized. This provides constructive content for path-based justifications: a cubical path $p\colon I \to A$ with $p(0) = a$ and $p(1) = b$ is a *concrete* justification, not merely

an assertion of equality. The cubical operations (composition, filling, coercion) provide the computational machinery for justified inference.

13.3 Presheaf Models and the Yoneda Constraint

The Yoneda perspective (Remark 2.3) suggests a deep connection to presheaf semantics. In a presheaf model $\mathrm{PSh}(\mathcal{I})$ over an index category $\mathcal{I}$ (of contexts), the internal logic is Kripke–Joyal semantics where truth is *context-dependent*: $i \Vdash \phi \Rightarrow \psi$ iff for all $f : j \to i$, $j \Vdash \phi$ implies $j \Vdash \psi$. This is precisely the semantics an LLM *should* implement: a claim is valid at a context if it remains valid under all refinements of that context. Current LLMs compute a flat conditional probability $P(\text{token} \mid \text{ctx})$ rather than a Kripke-forced truth $i \Vdash \phi$, losing the universal quantification over context refinements.

13.4 Limitations and Open Problems

1. **Knowledge complex construction.** Building $\mathcal{K}_\bullet$ requires explicit co-consistency judgments. Automating this from raw corpora is itself a hard problem.

2. **Decidability.** Type inhabitation for full homotopy types is undecidable. Practical systems must work with truncated or decidable fragments.

3. **Scaling persistent homology.** While polynomial-time, the cubic cost of persistent homology limits real-time application to moderate-dimensional embeddings.

4. **Empirical validation.** The framework generates precise, testable predictions (e.g., hallucination frequency should correlate with the Betti numbers of the local knowledge complex). Experimental verification is an urgent next step.

5. **The learning problem.** How does a model *learn* the homotopy structure of **Sem** from data? Can persistent homology of training data embeddings serve as a training signal?

14 Conclusion

We have established the Hallucination–Homotopy Correspondence: a complete, functorial bijection between the failure modes of large language models and the homotopy-theoretic invariants of semantic space. The central results are:

1. **Theorem 3.4** (Hallucination–Homology): hallucination $\iff$ non-trivial homology in $\mathcal{K}_\bullet$.
2. **Theorem 4.1** (HHC Classification): five hallucination types $\leftrightarrow$ five homotopy-theoretic invariants.
3. **Theorem 4.3** (Functoriality): the correspondence is a faithful, isomorphism-reflecting functor $\Phi : \mathsf{Hall} \to \mathsf{hTop}_*$.
4. **Theorem 5.1** (Fundamental Obstruction): no faithful functor from **Sem** to any contractible category exists.
5. **Theorem 7.2** (Sheaf Obstruction): hallucination $\iff H^1(\mathcal{K}_\bullet, \mathcal{F}) \neq 0$.

The path forward requires lifting language models from **Vect** to ∞-**Grpd**: from flat embedding spaces to structured homotopy types preserving the path, coherence, and higher-dimensional information that constitutes meaning. The Univalence Constraint replaces cosine similarity with genuine semantic equivalence. Type inhabitation search replaces next-token prediction. Abstention replaces hallucination when the target type is uninhabited.

The gap between this vision and current practice is large. But the persistent homology approach provides a practical near-term bridge, and the hybrid architecture offers a path to incremental adoption. We believe the Hallucination–Homotopy Correspondence provides the correct mathematical foundation for understanding—and eventually solving—the hallucination problem.

References

[1] The Univalent Foundations Program. *Homotopy Type Theory: Univalent Foundations of Mathematics*. Institute for Advanced Study, 2013.

[2] B. Coecke, M. Sadrzadeh, and S. Clark. Mathematical Foundations for a Compositional Distributional Model of Meaning. *Linguistic Analysis*, 36(1–4):345–384, 2010.

[3] J. Bolt, B. Coecke, F. Genovese, M. Lewis, D. Marsden, and R. Piedeleu. Interacting Conceptual Spaces I: Grammatical Composition of Concepts. In *Concepts and their Applications*, Springer, 2019.

[4] G. Carlsson. Topology and Data. *Bulletin of the American Mathematical Society*, 46(2):255–308, 2009.

[5] G. Naitzat, A. Zhitnikov, and L. H. Lim. Topology of Deep Neural Networks. *Journal of Machine Learning Research*, 21(184):1–40, 2020.

[6] X. Li. Memory as Structured Trajectories: Persistent Homology and Contextual Sheaves. *arXiv preprint arXiv:2508.11646*, 2025.

[7] A. Ayzenberg et al. Sheaf Theory: from Deep Geometry to Deep Learning. *arXiv preprint arXiv:2502.15476*, 2025.

[8] Z. Ji, N. Lee, R. Frieske, et al. Survey of Hallucination in Natural Language Generation. *ACM Computing Surveys*, 55(12):1–38, 2023.

[9] L. Huang, W. Yu, W. Ma, et al. A Survey on Hallucination in Large Language Models. *arXiv preprint arXiv:2311.05232*, 2023.

[10] OpenAI. GPT-4 Technical Report. *arXiv preprint arXiv:2303.08774*, 2023.

[11] L. Ouyang, J. Wu, X. Jiang, et al. Training Language Models to Follow Instructions with Human Feedback. *NeurIPS*, 2022.

[12] P. Lewis, E. Perez, A. Piktus, et al. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. *NeurIPS*, 2020.

[13] C. Cohen, T. Coquand, S. Huber, and A. Mörtberg. Cubical Type Theory: A Constructive Interpretation of the Univalence Axiom. *FLAP*, 4(10):3127–3170, 2017.

[14] V. Voevodsky. Univalent Foundations Project. *NSF Grant Proposal*, 2010.

[15] J. Lurie. *Higher Topos Theory*. Princeton University Press, 2009.

[16] J. P. May. *A Concise Course in Algebraic Topology*. University of Chicago Press, 1999.

[17] H. Edelsbrunner and J. L. Harer. *Computational Topology: An Introduction*. American Mathematical Society, 2010.

[18] U. Bauer. Ripser: Efficient Computation of Vietoris–Rips Persistence Barcodes. *Journal of Applied and Computational Topology*, 5:391–423, 2021.

[19] E. Riehl. *Categorical Homotopy Theory*. Cambridge University Press, 2014.

[20] M. Shulman. Univalence for Inverse Diagrams and Homotopy Canonicity. *Mathematical Structures in Computer Science*, 25(5):1203–1277, 2015.

[21] F. W. Lawvere. Adjointness in Foundations. *Dialectica*, 23(3/4):281–296, 1969.

[22] J. Curry. Sheaves, Cosheaves and Applications. *arXiv preprint arXiv:1303.3255*, 2014.

[23] M. Robinson. *Topological Signal Processing*. Springer, 2014.