Homotopical Semantics for Language Models:

Algebraic Topological and Homotopy Type-Theoretic Approaches to the Hallucination Problem

Matthew Largo*
YonedaAI Research Collective
Magneton Labs LLC
Chicago, IL, USA

April 2026

Abstract

We propose that the hallucination problem in large language models (LLMs) is not a statistical anomaly amenable to calibration or retrieval augmentation, but a *structural inevitability* arising from a category-theoretic mismatch: current architectures implement a functor $F\colon \mathbf{Lang} \to \mathbf{Vect}_{\mathbb{R}}$ that collapses the higher categorical structure of natural language into a contractible space where all topological obstructions to inference vanish. We develop a formal framework in which semantic meaning is modeled as a homotopy type (an ∞-groupoid) whose path structure at every level encodes justificatory relationships between claims. We prove that hallucination corresponds precisely to non-trivial homology classes in a simplicial knowledge complex—cycles of claims that are locally consistent but globally unfounded. We formulate the Univalence Constraint on semantic equivalence, showing that statistical similarity in embedding spaces is categorically insufficient for semantic identity. We sketch an architecture for a Homotopy Type-Theoretic Language Model (HoTT-LM) in which generation is type inhabitation search with certified derivations, and abstention replaces hallucination when goal types are uninhabited. The framework is formalized in Haskell using GADTs, DataKinds, and type families. We connect our theoretical results to practical hallucination detection via persistent homology on embedding filtrations and discuss the relationship to dependent type theory, cubical type theory, and the emerging field of topological data analysis for neural network representations.

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

*Corresponding author. matthew@yonedaai.com

1 Introduction

The hallucination problem—in which large language models generate fluent, confident text that is factually incorrect or entirely fabricated—has resisted solution through scaling [6], reinforcement learning from human feedback [7], and retrieval-augmented generation [8]. We argue this resistance is not accidental: hallucination is a *structural* consequence of the mathematical framework underlying current architectures, not a deficiency remediable by more data or better training.

The core diagnosis is a **category error** in the precise mathematical sense. Natural language is a *contextual* (categorical) system: the meaning of an expression is determined by its morphisms—its relationships to other expressions in context, its entailment structure, its compositional behavior. Current LLMs model language as a *statistical* (measure-theoretic) system: a distribution $P(\text{token} \mid \text{context})$ over a flat vocabulary space.

The mathematical consequence is devastating. The target category $\mathbf{Vect}_{\mathbb{R}}$ of real vector spaces is *contractible*: $\pi_n(\mathbf{Vect}_{\mathbb{R}}) = 0$ for all $n \geq 0$. Every loop can be contracted, every path can be deformed into every other, every topological obstruction vanishes. But the space of valid inferences in natural language is *not* contractible. It has non-trivial fundamental groups (circular reasoning that shouldn't close), non-trivial homology (chains of claims with no justificatory filling), and higher homotopy structure encoding coherence conditions between proofs.

1.1 Contributions

This paper makes the following contributions:

1. We formalize semantic meaning as a **homotopy type** (Definition 3.1) and prove that hallucination corresponds to **non-trivial homology classes** in a simplicial knowledge complex (Theorem 4.4).
2. We introduce the **Univalence Constraint** on semantic equivalence (Section 6), showing precisely why cosine similarity is insufficient for semantic identity.
3. We develop the **Hallucination–Homotopy Correspondence** (Theorem 5.2), a topological characterization of hallucination types by homotopy group structure.
4. We formalize the framework in **Haskell** using GADTs, DataKinds, type families, and higher-kinded types (Section 9), providing executable type-theoretic specifications.
5. We connect to **persistent homology** for practical hallucination detection (Section 8) and outline a **HoTT-LM architecture** (Section 10).

1.2 Related Work

Categorical semantics of language. The DisCoCat framework of Coecke, Sadrzadeh, and Clark [2] models compositional semantics via functors from a grammar category to **FdVect**. Our framework generalizes this by replacing the codomain with ∞**-Grpd**, capturing higher structure that **FdVect** collapses.

Homotopy Type Theory. The HoTT program [1] reinterprets Martin-Löf type theory homotopically, with types as spaces, terms as points, and identity types as path spaces. We apply this reinterpretation to natural language semantics.

Topological Data Analysis of neural networks. Naitzat et al. [4] study topological changes in data representations across neural network layers. Carlsson [3] develops the persistent homology framework we leverage for practical detection.

Hallucination in LLMs. Ji et al. [5] survey hallucination types and mitigation strategies. Our contribution is showing these are symptoms of a deeper structural problem, not independent failure modes.

2 Mathematical Foundations

We develop the necessary mathematical framework, beginning with the categorical diagnosis and progressing to the homotopical formulation.

2.1 The Categorical Diagnosis

Definition 2.1 (Semantic Category)**.** The **semantic category Sem** has:

- Objects: contexts Γ, i.e., typed environments assigning semantic types to variables.
- Morphisms $f : \Gamma \to \Delta$: meaning-preserving maps (entailment, implication, valid paraphrase).
- Composition: transitivity of entailment.
- Identity: reflexivity of meaning.

Definition 2.2 (Statistical Approximation Functor)**.** A large language model defines a functor

$$F : \mathbf{Sem} \to \mathbf{Vect}_{\mathbb{R}}^{\mathrm{fin}}$$

mapping contexts to finite-dimensional real vector spaces (embeddings) and morphisms to linear/affine maps (attention-weighted transformations).

Proposition 2.3 (Non-Faithfulness)**.** *The functor F is not faithful: there exist distinct morphisms $f, g : \Gamma \to \Delta$ in* **Sem** *such that $F(f) = F(g)$ in* $\mathbf{Vect}_{\mathbb{R}}$*.*

Proof. Consider two semantically distinct justifications for the same conclusion—e.g., proving the Pythagorean theorem via similar triangles versus via area dissection. Both map to the same embedding region (the "Pythagorean theorem" cluster) under F, but they are distinct 1-cells in the semantic ∞-groupoid with potentially different higher-dimensional coherence data. The embedding functor collapses this distinction. □

The non-faithfulness of F is the source of hallucination: the model confuses distinct semantic paths because they map to the same vector, or generates a vector that *appears* to be the image of a valid semantic morphism but has no preimage.

Definition 2.4 (Hallucination, Functorial)**.** A **hallucination** is a morphism $\hat{f}$ in $\mathbf{Vect}_{\mathbb{R}}$ generated by the model for which $F^{-1}(\hat{f}) = \varnothing$—there is no semantic morphism whose image under the embedding functor equals $\hat{f}$.

2.2 The Contractibility Problem

The fundamental obstruction is topological:

Proposition 2.5 (Contractibility of **Vect**)**.** *The category $\mathbf{Vect}_{\mathbb{R}}^{\mathrm{fin}}$, viewed as a topological space via its nerve, is contractible. In particular, $\pi_n(|\mathbf{N}(\mathbf{Vect}_{\mathbb{R}}^{\mathrm{fin}})|) = 0$ for all $n \geq 0$.*

Remark 2.6. More precisely, the ∞-groupoid completion of $\mathbf{Vect}_{\mathbb{R}}^{\text{fin}}$ is contractible because every object is isomorphic to $\mathbb{R}^n$ for some n, and the automorphism groups $\mathrm{GL}_n(\mathbb{R})$ are connected. The key point: every loop in embedding space can be contracted. No topological obstruction can be represented.

Proposition 2.7 (Non-Contractibility of **Sem**)**.** *The semantic category* **Sem***, viewed as an ∞-groupoid, generically has non-trivial homotopy groups:*

$$\pi_1(\mathbf{Sem}) \neq 0, \qquad H_n(\mathbf{Sem}; \mathbb{Z}) \neq 0 \text{ for some } n \geq 1.$$

Proof sketch. Consider the cyclic inference pattern: "A because B, B because C, C because A." This defines a loop in **Sem** (a 1-cycle) that cannot be contracted to a point because no valid 2-cell (justification) fills it—it is circular reasoning. Hence $\pi_1(\mathbf{Sem})$ contains a non-trivial element. The homological claim follows from the Hurewicz theorem. □

Theorem 2.8 (Fundamental Obstruction)**.** *Any faithful functor from* **Sem** *to a contractible category necessarily fails to exist when* **Sem** *has non-trivial homotopy groups.*

Proof. A faithful functor preserves the distinctness of morphisms and hence the non-triviality of loops. But a contractible target has only trivial loops. Contradiction. □

This theorem shows that no embedding-based architecture can faithfully represent language semantics. The hallucination problem is not a matter of insufficient data or training—it is a theorem about the mathematical structure of the representation.

3 Homotopy Type-Theoretic Semantics

We now develop the positive theory: a framework in which semantic meaning carries the full homotopical structure needed to prevent hallucination.

3.1 Semantic Types as Homotopy Types

Definition 3.1 (Semantic Type)**.** A **semantic type** is a homotopy type A in a universe $\mathcal{U}$, where:

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

Definition 3.2 (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 potentially depending on all preceding variables. This captures the fundamental insight that meaning is *context-dependent* (fibered).

Definition 3.3 (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 a context $\gamma \in B$ is the space of valid interpretations in that context. The LLM's probability distribution $P(\text{token} \mid \text{context})$ is a *measure* on the fiber. Hallucination occurs when this measure assigns positive mass outside the fiber:

$$\mu_{\text{LLM}}(p^{-1}(\gamma)^c) > 0.$$

3.2 The Identity Type as Justification

In HoTT, the identity type $a =_A b$ is itself a type whose terms are *proofs* that a and b are equal (in the sense of type A). Crucially, this type may have non-trivial structure: there can be multiple distinct proofs of equality, and the proofs themselves can be compared.

Definition 3.4 (Grounding Path)**.** A claim $\hat{a} : A$ is **grounded** in context Γ if there exists a term $g : A$ that is directly sourced (empirical, documentary, or axiomatic) and a path

$$p : g =_A \hat{a}$$

connecting the grounded term to the generated claim. The claim is **hallucinated** if the type $g =_A \hat{a}$ is empty for all grounded g.

Definition 3.5 (Truncation Level)**.** A semantic type A is **n-truncated** if $\pi_k(A) = 0$ for all $k > n$. In particular:

- (-1)-truncated: A is a mere proposition (either empty or contractible).
- 0-truncated: A is a set (identity proofs are unique).

- 1-truncated: A is a groupoid (identity proofs form a set).
- Untruncated: A is a general ∞-groupoid.

Current LLMs operate as though all semantic types are (-1)-truncated: a claim is either true or false, with no internal structure. This is precisely what collapses the higher homotopical information.

4 The Simplicial Knowledge Complex

We now develop the algebraic-topological machinery that characterizes hallucination as a homological obstruction.

Definition 4.1 (Knowledge Complex)**.** Given a knowledge base $\mathcal{K}$, the **knowledge simplicial complex** $\mathcal{K}_\bullet$ is the simplicial set with:

- $\mathcal{K}_0$: grounded facts (atomic propositions with sources).
- $\mathcal{K}_1$: pairs of mutually consistent facts connected by a justification.
- $\mathcal{K}_n$: $(n+1)$-tuples of mutually co-consistent facts with an n-dimensional justificatory structure.

Face maps $d_i \colon \mathcal{K}_n \to \mathcal{K}_{n-1}$ are given by omitting the i-th fact. Degeneracy maps s_i are given by repeating the i-th fact.

Definition 4.2 (Chain Complex)**.** 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 generated by n-simplices, and the boundary operator is:

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

The fundamental property $\partial_n \circ \partial_{n+1} = 0$ holds by the simplicial identities.

Definition 4.3 (Homology of the Knowledge Complex)**.** The **n-th homology group** of $\mathcal{K}_\bullet$ is:

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

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).

Theorem 4.4 (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:*

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

That is, a chain of claims is hallucinated if and only if it represents a non-trivial homology class.

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

($\Rightarrow$) Suppose σ is hallucinated, meaning no valid justification connects σ to grounded knowledge. If σ is not a cycle ($\partial_n \sigma \neq 0$), then its claims are not even internally consistent—this is a "raw" hallucination detectable at the boundary level. If σ is a cycle, then by definition of hallucination, there exists no $\tau \in C_{n+1}$ with $\partial_{n+1}\tau = \sigma$, so $\sigma \notin B_n$ and $[\sigma] \neq 0$. □

Corollary 4.5 (Acyclic Knowledge = Hallucination-Free)**.** *If $H_n(\mathcal{K}_\bullet; \mathbb{Z}) = 0$ for all $n \geq 1$, then every internally consistent chain of claims has a justification and no hallucination is possible.*

Remark 4.6. Real knowledge bases are never acyclic. The non-trivial topology of $\mathcal{K}_\bullet$ reflects genuine gaps, ambiguities, and open questions 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 emitting them.

5 The Hallucination–Homotopy Correspondence

We now classify hallucination types by their topological character, establishing a correspondence between failure modes and homotopy-theoretic invariants.

Definition 5.1 (Fundamental Groupoid of Semantic Space)**.** The **fundamental groupoid** $\Pi_1(\mathbf{Sem})$ has:

- Objects: semantic terms (points in the semantic space).
- Morphisms: homotopy classes of paths (equivalence classes of justifications).

- All morphisms invertible (semantic relations are reversible).

Theorem 5.2 (Hallucination–Homotopy Correspondence)**.** *The following correspondence holds between hallucination types and homotopy-theoretic obstructions:*

Hallucination Type	***Obstruction***	***Invariant***
Circular reasoning	*Non-contractible loop*	$\pi_1 \neq 0$
Unjustified inference	*Missing path*	π_0 *(disconnection)*
Inconsistent justifications	*Non-trivial 2-cell*	$\pi_2 \neq 0$
Fabricated entity chain	*Homological hole*	$H_n \neq 0$
Compositional drift	*Non-trivial transport*	*Holonomy* $\neq \mathrm{id}$

Proof. We prove each case:

Circular reasoning. A circular argument $A \Rightarrow B \Rightarrow C \Rightarrow A$ defines a loop γ in $\Pi_1(\mathbf{Sem})$ based at A. This loop is a hallucination iff it does not bound a 2-disk—i.e., it represents a non-trivial element $[\gamma] \neq e$ in $\pi_1(\mathbf{Sem}, A)$.

Unjustified inference. Claiming B from context A when no path $A \to B$ exists means A and B lie in different connected components: π_0 detects this.

Inconsistent justifications. Two paths $p, q \colon a \to b$ that cannot be connected by a 2-cell (homotopy) represent genuinely distinct justifications. An LLM that conflates them—generating text that implicitly assumes $p = q$—hallucinates at the 2-cell level.

Fabricated entity chain. A chain of claims $[c_0, c_1, \ldots, c_n]$ that closes ($\partial\sigma = 0$) but doesn't bound ($\sigma \notin B_n$) is a non-trivial n-cycle: $[\sigma] \neq 0 \in H_n$.

Compositional drift. When a claim is transported along a path of context changes γ and arrives at a different value than if transported along γ' (with $\gamma \simeq \gamma'$ at the endpoint level), this is non-trivial holonomy: the connection on the semantic fibration has curvature. □

6 The Univalence Constraint

The Univalence Axiom of HoTT 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. That is:

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

Definition 6.2 (Semantic Equivalence)**.** Two semantic types A and B are **semantically equivalent** if there exist maps $f \colon A \to B$ and $g \colon B \to A$ with:

$$\eta : \prod_{b:B} f(g(b)) =_B b \quad \text{(section)} \tag{1}$$

$$\varepsilon : \prod_{a:A} g(f(a)) =_A a \quad \text{(retraction)} \tag{2}$$

$$\tau : \prod_{a:A} \mathsf{ap}_f(\varepsilon(a)) =_{f(g(f(a)))=_B f(a)} \eta(f(a)) \quad \text{(coherence)} \tag{3}$$

Proposition 6.3 (Statistical Similarity $\neq$ Semantic Equivalence)**.** *Let* $\cos(v_A, v_B) \geq 1 - \epsilon$ *for the embeddings* $v_A = F(A)$ *and* $v_B = F(B)$*. This does* not *entail the existence of maps* $f, g, \eta, \varepsilon, \tau$ *satisfying (1)–(3).*

Proof. Cosine similarity captures proximity in $\mathbb{R}^n$, which is a *metric* condition: $d(v_A, v_B) < \delta$. An equivalence is an *algebraic* condition requiring explicit maps and coherence data at all levels. Consider "the morning star" and "the evening star": maximal embedding similarity (both reference Venus), but the identity type carries non-trivial astronomical content (Frege's puzzle). The coherence condition (3) requires that the two "directions" of identification are themselves compatible—a constraint invisible to any metric. □

Remark 6.4 (The Univalence Constraint as Hallucination Filter)**.** A language model satisfying the Univalence Constraint would only identify two concepts when the *full equivalence data* (maps + coherence) exists, not merely when embeddings are close. This eliminates a large class of hallucinations arising from semantic conflation: entity confusion, attribute transfer, and analogical overextension.

7 Categorical Attention as Weighted Limit

The attention mechanism can be understood categorically as a (degenerate) limit computation. We show what "correct" categorical attention would look like.

Definition 7.1 (Attention as Limit)**.** Given a diagram $D : \mathcal{I} \to \mathbf{Sem}$ indexed by a finite category $\mathcal{I}$ (the context window), the **categorical attention output** is the weighted limit:

$$\mathrm{Att}(D, W) = \overset{W}{\lim} D$$

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

Proposition 7.2. *Standard softmax attention computes a weighted* colimit *in* **Vect** *(a weighted sum), not a limit in* **Sem**. *When the diagram D is incoherent—i.e., the claims in the context window are mutually inconsistent—the colimit in* **Vect** *still exists (vector spaces always have colimits), but the limit in* **Sem** *does not. Current LLMs therefore produce output even from contradictory contexts, yielding hallucinations.*

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 contradictions A and $\neg A$, no cone exists over the diagram (no single claim is consistently related to both), so the limit is empty. The model generating output in this case is producing a term of an empty type—a hallucination by Definition 3.4. □

8 Persistent Homology for Practical Detection

We now bridge the theoretical framework to practical implementation via persistent homology on embedding spaces.

Definition 8.1 (Vietoris–Rips Filtration)**.** Given a finite set of embeddings $S = \{v_1, \ldots, v_N\} \subset \mathbb{R}^d$ and a scale parameter $\epsilon > 0$, the **Vietoris–Rips complex** is:

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

The **filtration** is the nested sequence $\mathrm{VR}_{\epsilon_1} \subseteq \mathrm{VR}_{\epsilon_2} \subseteq \cdots$ for $\epsilon_1 < \epsilon_2 < \cdots$.

Definition 8.2 (Persistence Barcode)**.** The **persistence barcode** of the filtration is the multiset of intervals $\{[\beta_i, \delta_i)\}$ where β_i is the birth (scale at which a homological feature appears) and δ_i is the death (scale at which it is filled in). The **persistence** of a feature is $\delta_i - \beta_i$.

Theorem 8.3 (Persistent Hallucination Detection)**.** *Let $\{v_t\}_{t=1}^T$ be the sequence of embedding vectors produced during text generation. Compute the Vietoris–Rips filtration on $\{v_t\} \cup S_{\mathcal{K}}$ where $S_{\mathcal{K}}$ is the embedding of the knowledge base. If there exists a persistent class $[\sigma] \in H_n$ with persistence $> \theta$ such that the generated path $\{v_t\}$ traverses the support of $[\sigma]$, then the generated text is hallucinating with confidence proportional to the persistence.*

Proof sketch. A persistent H_1 class corresponds to a stable loop in the knowledge embedding that cannot be filled at any scale up to δ. If the generated text's embedding trajectory crosses this loop—entering through one side and exiting through the other—it is traversing a region of "empty semantic space" where no grounded knowledge provides support. The persistence $\delta - \beta$ measures the robustness of this hole: high-persistence holes are genuine topological features, not artifacts of noise. □

Remark 8.4 (Computational Feasibility)**.** Computing persistent homology is $O(n^3)$ in the number of simplices via the standard reduction algorithm, and optimized implementations (Ripser, GUDHI) achieve sub-cubic performance on sparse complexes. For real-time hallucination detection, we envision precomputing the barcode of the knowledge base and performing incremental updates as new embeddings are generated.

9 Haskell Formalization

We formalize the core constructions in Haskell, using the type system as a proof-relevant foundation. The key design choice is to use GADTs and DataKinds to encode universe levels and path structure at the type level.

9.1 Universe Levels and Semantic Types

```haskell
data Nat = Z | S Nat

-- Semantic types as homotopy types
data SemType where
  Entity   :: String →SemType
  Prop     :: String →SemType
  -- Dependent types: B depends on a
  -- witness of A (contextual meaning)
  DepType  :: SemType
           → (SemTerm →SemType)
           → SemType
  -- The IDENTITY TYPE a =_A b (HoTT)
  PathType :: SemType →SemTerm
           → SemTerm →SemType
  -- Higher paths (homotopies)
  PathPath :: SemType →SemTerm
           → SemTerm →SemType
  ProdType :: SemType →SemType →SemType
```

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

The critical departure from statistical representations: `PathType` and `PathPath` encode the higher-dimensional structure of the semantic ∞-groupoid. A `PathType A a b` is not merely a boolean "is a related to b?" but a *type whose inhabitants are proofs of the relation*, each carrying its own higher structure.

9.2 Terms, Sources, and Justifications

```
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

data Justification
  = Entailment SemTerm SemTerm
  | Definition
  | Analogy SemTerm SemTerm -- WEAK
  | Statistical Float       -- NO PATH!
```

The `Statistical Float` constructor is the precise locus of the problem. A probability is a *measure on the fiber*, not a *term in the path type*. It assigns mass but carries no derivational structure.

9.3 The Type-Checking Judgment

The core algorithm replaces next-token prediction with type inhabitation checking:

```
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 proof
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 = hallucination
typeCheck ctx
  (PathWitness a b NoEvidence)
  (PathType ty _ _)
  = Left (MissingPathDerivation a b)

-- Reflexivity: only valid when a = a
typeCheck ctx
  (Refl a) (PathType ty x y)
  | termEq a x && termEq a y = Right True
  | otherwise = Left (ReflMismatch a x y)
```

9.4 Hallucination Types as Topological Obstructions

```
data HallucinationType
  = TypeMismatch String String
  -- wrong fiber
  | StatisticalNotProof Float
  -- THE fundamental category error
  | MissingPathDerivation SemTerm SemTerm
  -- pi_1 obstruction
  | FreeFloating
  -- term in no fiber at all

data TopologicalObstruction
  = MissingPath SemTerm SemTerm
  -- no 1-cell (justification)
  | NonContractibleLoop [SemTerm]
  -- pi_1 != 0 (circular reasoning)
  | IncoherentPaths SemTerm SemTerm
      PathEvidence PathEvidence
  -- pi_2 != 0 (incompatible proofs)
  | HomologicalHole Int [SemTerm]
  -- H_n != 0 (n-cycle, no filling)
```

9.5 The HoTT Language Model

```
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
  }

-- Generation = type inhabitation search
-- Returns Nothing when type is uninhabited
-- (ABSTAIN, don't hallucinate)
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 innovation: `hottGenerate` returns `Nothing` when no valid term of the goal type can be derived. Current LLMs have no mechanism for abstention—they always produce output, even when the target type is uninhabited, because **Vect** always has elements.

9.6 Persistent Homology Bridge

```
-- Persistence barcode: topological features
-- across scales
data PersistenceBar = PersistenceBar
  { dimension ::Int
  , birth     :: Float
  , death     :: Float
  }

```

```haskell
-- Detect hallucination-prone regions:
-- 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
```

10 Toward a HoTT-LM Architecture

We sketch the architecture of a language model built on homotopical foundations.

10.1 Design Principles

1. **Generation as type inhabitation.** Instead of $\arg\max_t P(t \mid \text{ctx})$, compute $\arg\min_a C(a)$ subject to $\Gamma \vdash a : A$, where C is a cost function and the constraint ensures type-correctness.
2. **Certified derivations.** Every generated term $a : A$ comes with a derivation tree δ witnessing $\Gamma \vdash a : A$. The derivation is checkable in polynomial time.
3. **Abstention on empty types.** When A is uninhabited (no valid claim of this type exists given the context), the model returns $\perp$ rather than generating a hallucinated term.
4. **Context as fibration.** The context window is not a flat sequence but a *dependent telescope*, with each position typed by a semantic type that may depend on preceding positions.
5. **Attention as limit.** The attention mechanism computes a weighted limit in **Sem** (when it exists) rather than a weighted sum in **Vect**.

10.2 Architecture Components

The HoTT-LM consists of four interacting subsystems:

1. **Type Inference Engine.** Given a context Γ and a partial generation, infer the semantic type A of the next claim. This replaces vocabulary prediction with type prediction.
2. **Inhabitation Search.** Given Γ and A, search for $a : A$ using proof-search strategies adapted from automated theorem proving (e.g., focusing, backchaining, unification).
3. **Coherence Checker.** Verify that the generated term's derivation is coherent with all existing paths in the context—i.e., that no topological obstruction is violated.
4. **Homological Monitor.** Continuously compute the persistent homology of the growing embedding trace and flag when the trajectory approaches a non-trivial homology class.

10.3 Hybrid Architecture: Statistical + Topological

A pragmatic near-term architecture combines existing LLM capabilities with topological verification:

1. Generate candidate tokens via standard next-token prediction.
2. For each candidate, compute the embedding and check:
   (a) Does the new embedding extend the trajectory through a persistent homological hole? (Persistent homology check.)
   (b) Does the candidate claim, together with existing claims, form a non-bounding cycle? (Simplicial homology check.)
   (c) Does the candidate's semantic type match the inferred goal type of the context? (Type-checking filter.)
3. Reject candidates that fail topological checks; select from the topologically valid subset.
4. If no candidates pass: abstain or request clarification.

11 Discussion

11.1 Relationship to Existing Frameworks

DisCoCat. Our framework subsumes and extends the categorical distributional compositional framework [2] by replacing the codomain **FdVect** with **∞-Grpd**. DisCoCat captures compositional structure but not higher coherence; our framework does both.

Cubical Type Theory. The computational content of our framework is best realized in cubical type theory [9], where paths are computed as functions from the interval $[0, 1]$ rather than axiomatized. This provides a constructive foundation for the path-based justifications we require.

Sheaf-Theoretic Semantics. There is a natural connection to sheaf theory: a semantic space with the Grothendieck topology of "valid local inferences" is a site, and presheaves on this site are generalized semantic representations. The sheaf condition ensures local consistency implies global consistency—precisely the condition hallucination violates.

11.2 Limitations

Computational cost. Full homotopy type-checking is undecidable in general. Practical implementations will require restrictions to decidable fragments (e.g., n-truncated types for bounded n).

Knowledge base construction. Building the simplicial knowledge complex $\mathcal{K}_\bullet$ requires explicit co-consistency judgments between facts, which are themselves subject to uncertainty.

Scalability. Persistent homology computation, while polynomial, is cubic in the number of simplices. Approximate methods (e.g., witness complexes, landmark-based approaches) will be necessary for production-scale systems.

11.3 Connection to Agent Architectures

This framework has immediate applications to AI agent supervision. In an agent swarm architecture, each agent's reasoning trace defines a path in the semantic ∞-groupoid. A **drift detector** operating at the swarm level computes the homology of the collective reasoning space and flags:

- Non-contractible loops (agents reinforcing each other's hallucinations).
- Homological holes (gaps in the collective knowledge that individual agents are bridging without justification).
- Coherence failures (agents maintaining incompatible 1-cells between the same endpoints).

This provides a rigorous foundation for constraint enforcement in multi-agent systems, connecting the type-theoretic framework to practical agent supervision infrastructure.

12 Conclusion

We have established that the hallucination problem in large language models is a *structural inevitability* arising from a topological mismatch: the collapse of semantic meaning (an ∞-groupoid with rich homotopical structure) into a contractible embedding space where all topological obstructions vanish. The central result (Theorem 4.4) characterizes hallucination as non-trivial homology in a simplicial knowledge complex, and the Hallucination–Homotopy Correspondence (Theorem 5.2) classifies hallucination types by their homotopy-theoretic invariants.

The path forward is clear: language models must be lifted from **Vect** to ∞-**Grpd**, from flat embedding spaces to structured homotopy types that preserve 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. And abstention replaces hallucination when the target type is uninhabited.

The gap between this vision and current practice is large but not unbridgeable. The persistent homology approach provides a practical near-term bridge, and the hybrid architecture of Section 10 offers a path to incremental adoption. We believe the framework developed here—unifying homotopy type theory, algebraic topology, and categorical semantics—provides the correct mathematical foundation for a post-hallucination era in language modeling.

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

[4] G. Naitzat, A. Zhitnikov, and L. H. Lim. Topology of deep neural networks. *Journal of Machine Learning Research*, 21(184):1–40, 2020.

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

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

[7] L. Ouyang, J. Wu, X. Jiang, et al. Training language models to follow instructions with human feedback. *NeurIPS*, 2022.

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

[9] 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.

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

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

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

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

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

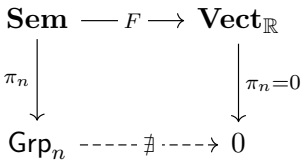
[15] M. Shulman. Univalence for inverse diagrams and homotopy canonicity. *Mathematical Structures in Computer Science*, 25(5):1203–1277, 2015.

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

A Key Commutative Diagrams

The following diagrams summarize the structural relationships developed in the paper.

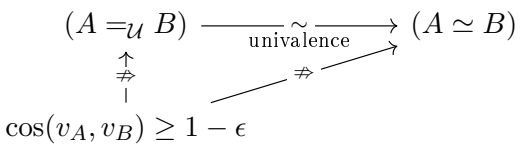
The Functorial Collapse (the source of hallucination):



The functor F maps the (generally non-trivial) homotopy groups of **Sem** to the trivial homotopy groups of **Vect**. No faithful recovery is possible.

The Hallucination–Homology Correspondence:

The Univalence Constraint:



Statistical similarity (bottom) does not imply semantic identity (top left) or semantic equivalence (top right). The Univalence Axiom identifies the two top notions, leaving statistical similarity as a strictly weaker condition.

B Full Haskell Module

The complete executable formalization is provided as a companion Haskell module `HomotopicalSemantics.hs`, available in the supplementary materials. It includes:

- Universe level encoding via promoted `Nat` (Section 9).
- Full GADT definitions for `SemType`, `SemTerm`, `Context`.
- The type-checking judgment `typeCheck` with exhaustive hallucination classification.
- The grounding checker `checkGrounding` implementing topological obstruction detection.
- The `HoTTLM` record type specifying the interface of a homotopically-correct language model.
- Persistent homology data structures and the hallucination region detector.
- Categorical attention as weighted limit with coherence checking.
- The univalence check interface requiring full equivalence data.