Contents

Higher Category Theory: A Primer

Layer 1: Simplicial Sets — The Combinatorial Language of Homotopy

Motivation

In classical mathematics, we often work with **sets** and **functions**, or with **categories** where objects are connected by morphisms that compose strictly. But many mathematical structures—especially those arising in topology, geometry, and homotopy theory—require a more flexible language. We need to describe not just objects and morphisms, but **spaces of morphisms**, and **spaces of ways those morphisms relate**, and **spaces of spaces of those relationships**, and so on.

This leads us to the idea of an ∞-**category**, where morphisms exist at all levels:

- 0-morphisms: objects,
- 1-morphisms: arrows between objects,
- 2-morphisms: homotopies between arrows,
- 3-morphisms: homotopies between homotopies,
- and so on, ad infinitum.

To encode this structure, we need a combinatorial framework that can represent **higher-dimensional coherence**. That framework is the **simplicial set**.

Formal Definition

A **simplicial set** is a functor:

$$X: \Delta^{\mathrm{op}} \to \mathrm{Set}$$

Here, Δ is the **simplex category**, whose objects are finite nonempty linearly ordered sets [n] = {0 < 1 < ... < n}, and whose morphisms are order-preserving maps. The opposite category Δ^{op} reverses the direction of these morphisms.

This means that a simplicial set assigns:

- To each [n], a set X_n of **n-simplices**,
- To each morphism [m] → [n], a **face map** or **degeneracy map** between simplices.

The data of a simplicial set is a hierarchy of sets X_0, X_1, X_2, ..., together with maps between them that encode how lower-dimensional pieces fit into higher-dimensional ones.

Intuitive Structure

Think of a simplicial set as a **combinatorial scaffold** for building shapes—but not physical shapes, rather the abstract pattern of how pieces fit together:

- A 0-simplex is a point.
- A 1-simplex is a directed edge between two points.
- A 2-simplex is a triangle, with three edges and three vertices.
- A 3-simplex is a tetrahedron, with four triangular faces.
- Higher simplices generalize this pattern.

Each simplex is not just a geometric figure—it's a **data structure** that encodes how its faces and edges relate to one another.

Imagine a **stop-motion animation** of relationships:

- Each frame (simplex) shows a configuration of objects and morphisms.
- The face maps remove elements from a configuration.
- The degeneracy maps duplicate elements.
- The entire animation encodes how structure evolves and coheres across dimensions.

The analogy captures the dynamic aspect, though mathematical simplicial sets are more rigid—the face and degeneracy maps satisfy precise identities that constrain how frames relate.

Or picture an **origami manual**:

- A 0-simplex is a dot on the paper.
- A 1-simplex is a crease connecting two dots.
- A 2-simplex is a fold that brings three creases into a triangle.

- A 3-simplex is a twist that connects four triangular folds into a tetrahedron.
- The simplicial set is the manual that tells you how to fold the paper into a coherent higher-dimensional shape.

But unlike origami, where physical constraints matter, simplicial sets are purely combinatorial—they encode relationships without geometric realization.

Simplicial Sets as Models for ∞-Categories

In ordinary categories:

- Objects are nodes.
- Morphisms are arrows.
- Composition is strictly defined.

In ∞-categories:

- Objects are 0-simplices.
- Morphisms are 1-simplices.
- Homotopies between morphisms are 2-simplices.
- Homotopies between homotopies are 3-simplices.
- And so on.

But composition is not strict—it is defined **up to homotopy**, and the homotopies themselves must satisfy **higher coherence conditions**.

This is encoded by the **inner horn-filling condition**.

Horns and Composition

A **horn** Λ^n_k is a partial simplex: an n-simplex missing one face. For example:

- Λ^2_1 is a triangle missing the edge from vertex 0 to 2.

If a simplicial set X allows every inner horn to be filled—i.e., every partial composition to be completed—then X behaves like an ∞-category. This horn-filling ensures that composition exists, but not necessarily uniquely—just **up to coherent deformation**.

Imagine drawing a triangle with only two sides. If you can infer the third side in a way that fits smoothly, you've filled the horn. This is composition in an ∞-category: not rigid, but **flexibly inferred**.

Or picture a jazz trio: the pianist and bassist play a phrase. The drummer hears it and fills in the rhythm that completes the musical idea. This captures something of the spirit—**coherent improvisation** that completes a structure. But the mathematical version is both more flexible and more rigid: flexible because the completion only needs to exist up to homotopy, rigid because once you fix the boundary, the space of possible completions is contractible. Jazz allows creative freedom; horn-filling yields essential uniqueness within homotopical flexibility.

The key insight: horn-filling replaces the strict equations of ordinary categories with topological conditions. Where a category demands f∘g = h exactly, a quasi-category only requires a 2-simplex witnessing their relationship. This shift from algebra to topology is what allows ∞-categories to capture phenomena that strict categories cannot.

Definition: ∞-Category (Quasi-Category)

A **quasi-category** is a simplicial set C that satisfies the inner horn-filling condition. That is:

- For every $n \geq 2$ and $0 < k < n$, every map $\Lambda^n_k \to C$ extends to $\Delta^n \to C$.

This definition replaces strict composition with **homotopy-coherent composition**.

In a strict category, you always take the same route from A to C via B. In a quasi-category, imagine navigation software: the route adapts to traffic, construction, and detours. You still get from A to C, but the path is **flexible and context-sensitive**. Importantly, different routes aren't arbitrary—they're related by explicit transformations that track how one path deforms into another.

Duality and Degeneracy

Simplicial sets encode not only composition but also **degeneracy**—ways in which higher-dimensional structure collapses into lower-dimensional forms. This is dual to the idea of extension: just as we can fill in missing faces (horns), we can also identify when a simplex is **degenerate**, meaning it arises from duplicating lower-dimensional data.

Degeneracy maps are the dual of face maps. Where face maps remove structure, degeneracy maps add redundancy. This duality is built into the definition of a simplicial set, and it reflects the deeper duality between **extension and restriction**, **composition and identity**, **structure and collapse**.

Final Worked Examples

Sudoku

- Each cell is a 0-simplex.
- A row or column is a 1-simplex: a constraint between cells.
- A 3×3 box is a 2-simplex: a higher-order constraint.
- The entire board is a simplicial complex: a structure built from interlocking constraints.

Solving Sudoku is like filling horns: you infer missing values from surrounding structure. The rules of Sudoku are the horn-filling conditions: they ensure coherence.

Go

- Each stone is a 0-simplex.
- A chain of stones is a 1-simplex.
- A group of stones with shared liberties is a 2-simplex.
- The evolution of the board is a growing simplicial set.

In Go, the meaning of a move depends on context—what surrounds it, what it connects to, and what it threatens. This is exactly how horns work in simplicial sets: a move (simplex) is only valid if it completes a coherent structure.

Layer 2: Left Fibrations

Encoding Homotopy-Coherent Presheaves of Spaces

Why This Layer, and Why Left?

This layer is the first structural refinement of the simplicial set language introduced in Layer 1. Simplicial sets gave us a way to encode ∞-categories as combinatorial objects. Now, we begin to ask: how do ∞-categories **organize data**? How do they **see other categories**, or **assign structure to objects**?

In classical category theory, this is the domain of **presheaves**—functors from a category to sets. In ∞-category theory, we generalize this to functors from an ∞-category to **spaces** (∞-groupoids). These are not just assignments of discrete data, but of **homotopy-coherent families of data**.

Left fibrations are the geometric encoding of such functors. They are the simplest kind of fibration in the ∞-categorical world, and they form the base case for understanding more general fibrations (like inner and Cartesian fibrations, which will follow in later layers). The reason we start with **left** fibrations is that they correspond to **covariant functors to spaces**, which are conceptually and technically simpler than their contravariant or more structured counterparts.

Right fibrations, which encode **contravariant** functors to spaces, are dual to left fibrations. They are equally fundamental, but their natural place in the tower comes after left fibrations, once the basic machinery of lifting and fibered structure is in place. Their omission here is not neglect, but sequencing.

Classical Presheaves and Their Generalization

In ordinary category theory, a **presheaf** on a category C is a functor:

$$F: C^{op} \to \text{Set}$$

This assigns to each object in C a set, and to each morphism $f: c \to d$, a restriction map $F(d) \to F(c)$. Presheaves model how data varies across a space or structure, and they underpin the theory of sheaves, topoi, and cohomology.

In ∞-category theory, we generalize this by replacing **sets** with **spaces**—that is, ∞-groupoids. The target of the functor becomes:

$$F: C \to S$$

where S is the ∞-category of spaces. This shift reflects the need to track not just discrete data, but **continuous families of data**, including paths, homotopies, and higher coherence.

Encoding Functors Geometrically

Rather than describing such functors directly, we encode them **geometrically** using **left fibrations**. A left fibration is a map of simplicial sets:

$$p: X \to S$$

satisfying a lifting condition: for every **left horn** $\Lambda^n_0 \hookrightarrow \Delta^n$, and every commutative diagram:

$$\Lambda^n_0 \longrightarrow X$$

$$\downarrow \qquad \downarrow p$$
$$\Delta^n \;\longrightarrow\; S$$

there exists a **lift** $\Delta^n \to X$ making the diagram commute.

This lifting condition ensures that for every morphism in the base S, and every point in the fiber over its source, there is a **coherent way to lift** that morphism into the total space X.

Intuition and Structure

Think of S as a **landscape** of locations, and X as a **cloud layer** hovering above it. Each location has its own cloud—its fiber—and the map p tells you which cloud belongs to which location. A left fibration ensures that if you draw a path between locations on the ground, you can lift that path into the clouds, starting from any point in the source cloud. The lift is not arbitrary—it's **coherent**, meaning it respects the structure of the landscape. Though note: unlike physical clouds, these lifts are essentially unique once you pick a starting point.

Or imagine S as a **library of books**, and X as a collection of **annotations**. Each book has its own annotations, and the map p tells you which annotations belong to which book. A left fibration ensures that if a book references another, the annotations follow that reference in a way that preserves interpretive structure—though the analogy is imperfect since mathematical coherence is more rigid than interpretive consistency.

Formal Consequence

A left fibration $p: X \to S$ encodes a functor:

$$F: C \to S$$

where C is the ∞-category modeled by S, and S is the ∞-category of spaces. The fiber over each object $s \in S$ is the space $F(s)$, and the lifting condition ensures that morphisms in C induce homotopy-coherent maps between these spaces.

This is formalized by the **straightening/unstraightening equivalence**, which will be developed in Layer 5.

Worked Example: Representable Presheaves

Let C be an ordinary category, and let $c \in C$ be an object. The **slice category** $C_{/c}$ consists of all objects mapping to c. This slice category corresponds to a right fibration over C, and encodes the **representable functor** $\text{Hom}(\text{-}, c)$.

In the ∞-categorical setting, this becomes a right fibration encoding the functor:

$$\text{Map}(\text{-}, c): C^{\text{op}} \to S$$

which assigns to each object d the space of morphisms from d into c.

Summary

- Left fibrations encode functors from ∞-categories to spaces.
- They generalize presheaves from sets to ∞-groupoids.
- The lifting condition ensures homotopy-coherent structure.
- They are the first step in modeling sheaves, descent, and ∞-topoi.
- Right fibrations, which encode contravariant functors to spaces, are dual and will be introduced once the machinery of lifting and fibered structure is fully developed.

Layer 3: Inner Fibrations

Encoding Homotopy-Coherent Functors Between ∞-Categories

Purpose and Position in the Tower

This layer generalizes the concept of left fibrations (Layer 2), which encode functors from an ∞-category to the ∞-category of spaces, by introducing **inner fibrations**—the structure that encodes **functors between arbitrary ∞-categories**. Where left fibrations model variation over a base with values in spaces, inner fibrations allow us to model variation where the values themselves carry higher categorical structure.

This is the natural next step in the Postnikov tower: having built the language of simplicial sets and understood how to encode presheaves of spaces, we now ask how to encode functors between ∞-categories themselves.

Formal Definition

Let $p: X \to S$ be a map of simplicial sets. We say that p is an **inner fibration** if for every **inner horn** $\Lambda^n_k \hookrightarrow \Delta^n$ with $0 < k < n$, and every commutative diagram:

$$\Lambda^n_k \longrightarrow X$$
$$\downarrow \qquad \downarrow p$$
$$\Delta^n \longrightarrow S$$

there exists a **lift** $\Delta^n \to X$ making the diagram commute.

This lifting condition ensures that **composable morphisms in the base** can be lifted to composable morphisms in the total space, in a way that preserves the homotopy-coherent structure of composition.

Intuition and Structure

Imagine S as a **network of workflows**, and X as a **layered system of implementations**. Each workflow in S corresponds to a process, and the map p tells you which implementation in X realizes that process. An inner fibration ensures that if two steps in a workflow are composable, their implementations in X can be composed in a way that respects the logic of the workflow. But crucially, the composition isn't unique—there might be multiple valid ways to compose implementations, all related by higher morphisms that track their equivalence.

Or think of S as a **library of abstract theories**, and X as a **catalog of concrete models**. Each theory has a model, and the map p tells you which model corresponds to which theory. An inner fibration ensures that if two theories are connected by a logical implication, their models are connected by a coherent transformation.

Why Inner?

The term "inner" refers to the fact that the horn-filling condition applies only to **inner horns**—those missing a face that is neither the first nor the last. This reflects the fact that inner fibrations encode **composition**, not just initial or terminal behavior. Left fibrations (Layer 2) fill horns missing the initial face; right fibrations fill horns missing the final face. Inner fibrations sit between them, encoding the full internal structure of functorial composition.

Encoding Functors Between ∞-Categories

Given an inner fibration p: X → S , we interpret X as a **family of ∞-categories** varying over the base S . The fiber over each object s ∈ S is an ∞-category, and morphisms in S induce **functors between these fibers**, encoded by the lifting condition.

This is the ∞-categorical analog of a **Grothendieck fibration** in classical category theory, which encodes a functor F: C → Cat by a category E fibered over C .

Duality and Directionality

Inner fibrations are **direction-neutral**: they encode functors between ∞-categories without privileging initial or terminal behavior. This makes them the natural generalization of both left and right fibrations. In later layers, we will refine inner fibrations into **Cartesian** and **coCartesian fibrations**, which encode functors to and from ∞-categories with additional structure—preserving limits, colimits, or adjunctions.

Summary

- Inner fibrations encode functors between ∞-categories.
- They generalize left and right fibrations by allowing composition to be lifted homotopy-coherently.
- The lifting condition applies to inner horns, reflecting the internal structure of composition.
- They are the foundation for modeling structured families of ∞-categories, and for defining Cartesian and coCartesian fibrations.

Layer 4: Cartesian Fibrations

Encoding Functors Valued in ∞-Categories via Structure-Preserving Lifts

Purpose and Position in the Tower

This layer builds directly on **inner fibrations** (Layer 3), which encode functors between ∞-categories in a homotopy-coherent way. Cartesian fibrations refine this idea: they encode **functors valued in ∞-categories**, not just in spaces, and they do so by enforcing a condition that ensures morphisms in the base category lift to morphisms in the total space in a way that **preserves internal structure**.

Where left fibrations encode functors to the ∞-category of spaces S, Cartesian fibrations encode functors to the ∞-category of ∞-categories Cat_∞. This is a step up in complexity and expressive power: the fibers are no longer just spaces, but full ∞-categories, and the transitions between them must respect their internal categorical structure.

Formal Definition

Let $p: X \to S$ be a map of simplicial sets. Then p is a **Cartesian fibration** if:

1. p is an **inner fibration**—i.e., it satisfies the lifting condition for all inner horns $\Lambda^n_k \hookrightarrow \Delta^n$ with $0 < k < n$,
2. For every edge $f: s \to t$ in S and every vertex $y \in X$ lying over t, there exists a **Cartesian edge** $\tilde{f}: x \to y$ in X lying over f with the following universal property: for any edge $h: w \to y$ in X and any factorization $p(h) = f \circ g$ in S, there exists a unique edge $h': w \to x$ in X such that $p(h') = g$ and $h = \tilde{f} \circ h'$.

This Cartesian lifting condition ensures that morphisms in the base category S can be lifted to morphisms in the total space X in a way that **preserves the internal structure of the fibers**.

Think of it like a camera tracking system following multiple subjects. When the camera pans from one subject to another, the Cartesian lift is the unique way to maintain focus that preserves depth relationships. Any other camera movement that reaches the same final position must factor through this canonical pan - you might add artistic flourishes, but the fundamental tracking motion is determined by the geometry of the scene. The lift is terminal among lifts: all variations are homotopic to this canonical one.

Conceptual Interpretation

Imagine S as a **timeline of evolving systems**, and over each point in time there is a **local universe**—an ∞-category of objects, morphisms, and higher morphisms. A Cartesian fibration ensures that as you move from one timepoint to another, there is a coherent way to **transport entire categorical structures**, not just individual objects. The Cartesian condition is what makes this transport "natural"—it respects the internal logic of each universe rather than imposing external choices.

Or think of S as a **catalog of mathematical theories**, and over each theory there is a **category of models**. A Cartesian fibration ensures that if one theory maps to another (say, via interpretation or extension), then the models of the first theory can be translated into models of the second in a way that respects their internal relationships.

This is not just about moving data—it's about moving **structured data**, where the structure itself is preserved and respected.

Encoding Functors to ∞-Categories

A Cartesian fibration p: X → S encodes a functor:

$$F: C \to Cat_\infty$$

where:

- C is the ∞-category modeled by S ,
- Cat_∞ is the ∞-category of ∞-categories,
- F assigns to each object in C an ∞-category,
- And to each morphism, a functor between ∞-categories that preserves structure via Cartesian lifts.

This is formalized by the **straightening/unstraightening equivalence**, which will be developed in Layer 5.

Duality: CoCartesian Fibrations

Just as Cartesian fibrations encode **structure-preserving pullbacks**, **coCartesian fibrations** encode **structure-preserving pushforwards**. Cartesian edges lift from the target (pulling back structure), while coCartesian edges lift from the source (pushing forward structure). This duality is not decorative —it reflects the fundamental symmetry of ∞-category theory. Every construction that pulls back has a counterpart that pushes forward.

CoCartesian fibrations encode functors:

$$F: C \to Cat_\infty$$

and are essential for modeling colimit-preserving functors, left Kan extensions, and descent.

Summary

- Cartesian fibrations encode functors from ∞-categories to ∞-categories.
- They generalize left fibrations, which encode functors to spaces.
- The lifting condition ensures structure-preserving transitions between fibers.

- CoCartesian fibrations are the dual notion, encoding forward-moving structure.
- These fibrations are foundational for modeling Kan extensions, adjunctions, and ∞-topoi.

Layer 5: Straightening and Unstraightening

Equivalence Between Fibrations and Functors in ∞-Category Theory

Purpose and Position in the Tower

This layer formalizes the deep equivalence between **Cartesian fibrations** (Layer 4) and **functors valued in ∞-categories**. It provides the machinery that allows one to move between a **geometric encoding** of a functor (as a fibration) and its **algebraic description** (as a mapping into Cat_∞). This equivalence is not a convenience—it is a structural bridge that allows ∞-category theory to be both combinatorially precise and conceptually flexible.

Straightening and unstraightening are the tools that make this bridge explicit. They allow us to treat fibrations as functors and vice versa, preserving all homotopy-coherent structure.

Statement of the Equivalence

Let S be a simplicial set modeling an ∞-category. Then there is an equivalence of ∞-categories:

$$\text{CartFib}_S \simeq \text{Fun}(S, \text{Cat}_\infty)$$

This means that:

- Every **Cartesian fibration** p: X → S corresponds to a **functor** F: S → Cat_∞ ,
- Every such functor can be **unstraightened** into a Cartesian fibration,
- The correspondence is **natural**, **homotopy-coherent**, and **invertible up to equivalence**.

This equivalence is the formal justification for treating fibrations as functors and functors as fibrations. It is the backbone of how ∞-categories organize structured variation.

Straightening: From Geometry to Algebra

Given a Cartesian fibration p: X → S , the **straightening** process constructs a functor:

$$St_S(p): S \to Cat_\infty$$

This functor assigns to each object $s \in S$ the fiber X_s , and to each morphism $f: s \to t$, a functor between fibers determined by the **Cartesian lift** of f .

Straightening is like converting a **folded map** into a **GPS routing system**. The folded map shows terrain and paths; straightening turns it into a digital system that tells you, for each location, what the local structure is and how to move between them. The key subtlety: unlike a real GPS that might lose detail, mathematical straightening preserves everything—every path, every alternative route, every way routes relate to each other.

Or think of it as extracting the **script** from a performance: the fibration is the enacted drama, and straightening reveals the underlying narrative logic.

Unstraightening: From Algebra to Geometry

Given a functor $F: S \to Cat\_\infty$, the **unstraightening** process constructs a Cartesian fibration $p: X \to S$ whose fiber over each object $s \in S$ is the ∞-category $F(s)$, and whose morphisms are organized by the functorial action of F .

Unstraightening is like taking a **blueprint** and constructing a **building**: the functor gives the design, and the fibration is the realized structure. But the construction is perfect—every detail of the blueprint is faithfully realized in the fibration, and conversely, straightening can recover the exact blueprint from the building.

Or imagine a **recipe book** being turned into a **kitchen**: each recipe becomes a station, and the layout reflects the structure of the book. The functor is the recipe logic; the fibration is the physical instantiation. Though the mathematical version is more faithful—unstraightening preserves all information exactly, while a real kitchen might lose the nuances of the cookbook's organization.

Why This Matters

This equivalence allows ∞-category theory to treat **structured families of ∞-categories** as geometric objects. It enables the definition of:

- **Limits and colimits** via fibrations,
- **Kan extensions** as universal constructions over fibrations,
- **Adjunctions** as relationships between fibrations,
- **Descent and gluing** in ∞-topoi.

It also allows one to **compute** with functors by manipulating their fibrational models, and vice versa.

Summary

The straightening/unstraightening equivalence is the bridge between geometric and algebraic perspectives in ∞-category theory. Rather than choosing one viewpoint, we gain power by moving fluidly between them: fibrations when we need geometric intuition, functors when we need algebraic computation.

Layer 6: Limits and Colimits in ∞-Categories

← Back to Contents

Universal Constructions in a Homotopy-Coherent Framework

Purpose and Position in the Tower

This layer builds on the machinery of fibrations and the straightening/unstraightening equivalence (Layer 5), and introduces the central universal constructions of ∞-category theory: **limits** and **colimits**. These are not merely generalizations of their classical counterparts—they are redefined to accommodate the **homotopy-coherent structure** of ∞-categories. Limits and colimits are the backbone of internal logic, gluing, descent, and extension in higher category theory.

They are defined not by strict commutativity, but by **contractibility of mapping spaces**—a shift from rigid diagrams to flexible, coherent ones.

Classical Background

In ordinary category theory, a **limit** of a diagram D: J → C is an object L ∈ C equipped with a cone over D such that any other cone factors uniquely through it. Dually, a **colimit** is an object C ∈ C receiving a cocone from D , universal among such cocones.

These definitions rely on **strict commutativity**: diagrams must commute exactly. In ∞-categories, this rigidity is replaced by **homotopy coherence**: diagrams commute up to higher morphisms, and those morphisms themselves satisfy coherence conditions.

∞-Categorical Reformulation

Let C be an ∞-category, and let D: K → C be a diagram, where K is a simplicial set modeling the shape of the diagram. A **limit** of D is an object L ∈ C together with a cone $\Delta^0 \to C_{/D}$ that is **terminal** in the ∞-category of cones over D . A **colimit** is defined dually, using the under-category $C_{D/}$, and is **initial** in the ∞-category of cocones under D .

The universal property is encoded not by uniqueness, but by **contractibility of mapping spaces**: the space of maps from any other cone to the universal one is homotopically trivial.

Conceptual Interpretation

Imagine a group of people trying to agree on a meeting point. In classical category theory, they must all agree on a single location, and the paths to that location must align perfectly. In ∞-category theory, they can each propose a location, and the paths may bend and twist, but as long as there is a coherent way to deform all proposals into a common space, a limit exists. The mathematical precision: the limit is characterized by having a contractible space of maps from any other cone, ensuring essential uniqueness.

Or picture a set of musical phrases being resolved into a final chord. In classical theory, the resolution must be exact—every voice lands on its prescribed note. In ∞-categories, the resolution can be flexible, as long as all phrases can be coherently transformed into the final chord. The colimit is the chord that absorbs and harmonizes all variations. But here's the subtlety: while there are many ways to voice the chord, they're all equivalent via a contractible space of re-voicings. The universality manifests not as rigid uniqueness but as essential uniqueness up to coherent deformation.

Limits are like **intersections**: they capture what is common across a diagram. Colimits are like **unions**: they synthesize the diagram into a coherent whole. But in ∞-categories, these constructions are not static—they are **spaces of solutions**, not single points. The intersection/union analogy works because limits preserve truth (if something is true in all components, it's true in the limit) while colimits detect existence (if something exists somewhere, it exists in the colimit).

Fibrational Encoding

Using the straightening/unstraightening equivalence, limits and colimits can be defined via **fibrations**:

- A **limit** is a terminal object in the ∞-category of sections of a Cartesian fibration.
- A **colimit** is an initial object in the ∞-category of sections of a coCartesian fibration.

This perspective allows one to compute limits and colimits by analyzing the geometry of fibrations, rather than manipulating diagrams directly.

Duality and Balance

Limits and colimits are dual constructions. Every definition, every analogy, every computational method has a mirror image. This duality is not a decoration—it is a structural principle. The ∞-category of spaces S is both complete and cocomplete: it admits all small limits and colimits. This balance is reflected in the theory itself.

Summary

- Limits and colimits in ∞-categories are defined via universal properties in the ∞-category of cones or cocones.
- These constructions rely on the machinery of fibrations and the straightening/unstraightening equivalence.
- The universal property is encoded by contractibility of mapping spaces, not uniqueness.
- Limits and colimits are central to defining adjunctions, Kan extensions, and the internal logic of ∞-topoi.

Layer 7: Kan Extensions

Universal Extension of Functors via Colimits and Cofinality

Purpose and Position in the Tower

Kan extensions are the **universal mechanism** by which functors are extended along other functors. They are the ∞-categorical analog of extending a function defined on a subset to a larger domain in a way that preserves structure. In the tower of Higher Topos Theory, Kan extensions sit atop colimits and cofinality, leveraging both to define **universal constructions** that preserve homotopy-coherent data.

They are indispensable in descent theory, sheafification, base change, and the formalization of adjunctions. Kan extensions are not merely technical—they are the **language of continuation**, of extending local data globally, and of transporting structure across ∞-categories.

Classical Background

In ordinary category theory, given functors

$$f: C \to D, \quad F: C \to E,$$

a **left Kan extension** of F along f , denoted $\mathrm{Lan}_f\, F: D \to E$, is a functor equipped with a natural transformation

$$\eta: F \Rightarrow \mathrm{Lan}_f\, F \circ f$$

that is **universal** among such transformations. That is, for any other G: D → E and α: F ⇒ G ∘ f , there exists a unique β: $\mathrm{Lan}_f\, F$ ⇒ G such that α = β ∘ η .

This universal property makes Kan extensions the **universal extension** of a functor along another (universal among such extensions).

∞-Categorical Definition

In ∞-category theory, the left Kan extension of a functor

$$F: C \to E$$

such that for each object d ∈ D , the value $\mathrm{Lan}_f\, F(d)$ is the **colimit** of the diagram

along

$$f: C \to D$$

is a functor

$$\mathrm{Lan}_f\, F: D \to E$$

equipped with a natural transformation

$$\eta: F \Rightarrow \mathrm{Lan}_f\, F \circ f$$

which is universal among such transformations. Equivalently, for each object d ∈ D the value $\mathrm{Lan}_f\, F(d)$ is the colimit of the diagram

$$C_{d/} \xrightarrow{\pi} C \xrightarrow{F} E,$$

where $C_{d/}$ is the under-category of objects of C equipped with a map to d, and π is the projection.

Suppose many sensors cover a city block and you must produce one value for the block. Form the diagram of sensor readings that map into the block and combine them (for instance by averaging in small neighborhoods) so that overlapping zones agree; the Kan extension is the formal rule that assembles those local readings into a single, coherent value and records how different legitimate assembly rules relate. The subtlety: there isn't one "correct" way to combine the readings—the Kan extension captures the entire space of reasonable combinations.

For another concrete example, given a short musical motif in a chamber group, you might assign parts to an orchestra so the original motif appears where it came from; the Kan extension records the family of consistent orchestrations and selects the universal one up to homotopy equivalence. Though of course, real orchestration involves aesthetic choices that go beyond mathematical consistency.

Using straightening/unstraightening, the functor F corresponds to a fibration over C, and $\text{Lan}_f F$ is obtained by pushing that fibration forward along f; the defining colimit is the fiberwise colimit of the pushed fibration.

Cofinality test (practical): a functor i: K → K' is cofinal if for every object k' ∈ K' the comma (undercategory) $K_{k'/}$ is weakly contractible; when this holds, colim_{K'}(F) ≃ colim_{K}(F∘ i). In practice, replace a large indexing diagram by a smaller cofinal subdiagram to simplify computation without changing the colimit.

Thus Kan extensions are computed by colimits, and cofinality is the main tool that makes those colimits tractable in concrete problems.

Duality: Right Kan Extensions

The **right Kan extension** $\text{Ran}_f F$ is defined dually via **limits** over the over-category $C_{/d}$. It represents the **universal right Kan extension** (universal among such restrictions), i.e. the terminal object among cones exhibiting the limiting property.

Right Kan extensions are crucial in **sheaf theory**, **adjunctions**, and **descent**, especially when one wants to **recover global data from local restrictions**.

Applications and Significance

- **Sheafification**: Kan extensions formalize the process of extending presheaves to sheaves.

- **Base Change**: In derived algebraic geometry, Kan extensions express how structure transforms under pullbacks and pushforwards.
- **Adjunctions**: Every adjunction arises from a Kan extension satisfying certain exactness properties.
- **Descent**: Kan extensions encode how local data glues to global data in a homotopy-coherent way.

Kan extensions are the **engine of universality** in ∞-category theory. They allow one to **extend, restrict, and transport** functors while preserving the deep structure encoded in higher morphisms.

Summary

- Kan extensions are universal constructions that extend functors along other functors.
- Left Kan extensions are defined via colimits over under-categories; right Kan extensions via limits over over-categories.
- They are computed using cofinality, making them tractable and flexible.
- They arise naturally in sheaf theory, descent, base change, and adjunctions.
- Kan extensions are the ∞-categorical language of continuation, extrapolation, and structural transport.

Layer 8: Adjunctions in ∞-Categories

← Back to Contents

The Homotopy-Coherent Dance of Left and Right Universality

Purpose and Position in the Tower

Adjunctions are the **architectural beams** of category theory—structures that support and unify vast swaths of mathematical landscape. In ∞-category theory, they retain their central role but acquire a **homotopy-coherent richness**: they are no longer mere pairs of functors satisfying a universal property, but **structured correspondences** between ∞-categories that preserve and reflect higher morphisms. The architectural metaphor is apt: just as beams carry load and transfer forces throughout a structure, adjunctions carry information and transfer structure throughout mathematics. But unlike physical beams that can bend or break, mathematical adjunctions are perfectly rigid in their coherence relations.

This layer builds directly on Kan extensions (Layer 8), since every adjunction arises from a Kan extension satisfying exactness conditions. Adjunctions are the **universal translators** between ∞-categories, enabling the transfer of structure, the definition of limits and colimits, and the construction of monads, sheaves, and derived functors.

Classical Background

In ordinary category theory, an adjunction between categories C and D consists of a pair of functors

$$F\colon C \leftrightarrows D \colon G$$

with a natural bijection

$$\mathrm{Hom}_D(F(c), d) \cong \mathrm{Hom}_C(c, G(d))$$

that is natural in both $c \in C$ and $d \in D$. The functor F is called **left adjoint** to G , and G is **right adjoint** to F .

This bijection expresses a **universal property**: F(c) is the most efficient way to map c into D , and G(d) is the best way to pull d back into C .

∞-Categorical Definition

In ∞-category theory, the hom-sets are replaced by **mapping spaces**, and the natural bijection becomes a **homotopy equivalence of spaces**:

$$\mathrm{Map}_D(F(c), d) \simeq \mathrm{Map}_C(c, G(d)).$$

This equivalence must be **natural in a homotopy-coherent sense**, meaning it respects not just objects and morphisms, but all higher simplices—commutative diagrams, homotopies between morphisms, and so on.

An adjunction in ∞-categories is thus a pair of functors

$$F\colon C \leftrightarrows D \colon G$$

together with a **unit transformation**

$$\eta\colon \mathrm{id}_C \to G \circ F$$

and a **counit transformation**

$$\varepsilon: F \circ G \to \mathrm{id}_D$$

satisfying **triangle identities up to coherent homotopy**.

Conceptual Interpretation

Adjunctions are like **translation protocols** between languages: the left adjoint translates ideas forward, the right adjoint translates them back. The mapping space equivalence ensures that meaning is preserved—not just at the level of words (objects), but at the level of grammar (morphisms) and nuance (higher homotopies). But the analogy has limits: mathematical adjunctions are more precise than linguistic translation, and the "round-trip" through both functors might transform the original in predictable ways.

Or picture a **dance**: the left adjoint leads, the right adjoint follows, and the mapping space equivalence ensures they move in perfect synchrony. The triangle identities are the choreography—ensuring that the dancers return to their original positions after a full cycle. Though unlike human dancers, mathematical adjunctions never miss a step—the coherence is perfect up to homotopy.

In a physical analogy, the left adjoint is like **pushing** a system forward (e.g., applying force), and the right adjoint is like **measuring** its response. The adjunction ensures that the push and the response are in perfect correspondence. But mathematical adjunctions are more structured than physical systems— the correspondence is exact up to coherent homotopy, with no loss or noise.

Adjunctions via Kan Extensions

Every adjunction arises from a **left Kan extension** that is **exact**—meaning it preserves colimits—and a **right adjoint** that preserves limits. The unit and counit transformations arise from the universal properties of these Kan extensions.

This connection makes adjunctions **computable** and **structural**: they are not arbitrary, but arise from the machinery of colimits, limits, and cofinality.

Fibrational Perspective

Using the straightening/unstraightening equivalence, an adjunction corresponds to a **pair of Cartesian fibrations** over a base ∞-category, with a **fiberwise equivalence of mapping spaces**. This perspective

reveals adjunctions as **structured correspondences between fibrations**, preserving homotopy-coherent data.

In this view, the unit and counit transformations are **lifts** of identity morphisms across fibrations, and the triangle identities express **homotopy coherence of these lifts**.

Adjunctions and Limits/Colimits

Adjunctions are the **engine behind the existence and preservation of limits and colimits**:

- Left adjoints preserve **colimits**.
- Right adjoints preserve **limits**.

This is not a coincidence—it is a consequence of the mapping space equivalence. The preservation properties are encoded in the **shape of the adjunction**, and they allow one to **transport universal constructions** across ∞-categories.

Applications and Significance

- **Sheaf Theory**: Adjunctions formalize the relationship between presheaves and sheaves via sheafification.
- **Monads and Comonads**: Every monad arises from an adjunction; adjunctions encode algebraic structure.
- **Descent and Gluing**: Adjunctions express how local data can be glued into global data.
- **Base Change**: In derived geometry, adjunctions express how structure transforms under pullbacks and pushforwards.
- **Homotopy Theory**: Adjunctions between model categories induce adjunctions between ∞-categories, preserving homotopical structure.

Adjunctions are the **universal bridges** of ∞-category theory. They connect, translate, and preserve structure across the vast landscape of higher categories. But these bridges have precise engineering: the left adjoint preserves colimits (it's "continuous" in the categorical sense), the right preserves limits, and the unit/counit satisfy triangle identities up to coherent homotopy.

Summary

- Adjunctions in ∞-categories are defined via homotopy equivalences of mapping spaces.

- They consist of a left and right adjoint, with unit and counit transformations satisfying triangle identities up to coherent homotopy.
- They arise from Kan extensions and are deeply connected to colimits and limits.
- They preserve and reflect structure, enabling descent, sheafification, base change, and algebraic constructions.
- Adjunctions are the universal translators of ∞-category theory.

Layer 9: Monads and Comonads in ∞-Categories

Algebraic Structure from Adjunctions in a Homotopy-Coherent Setting

Purpose and Position in the Tower

Monads and comonads arise naturally from adjunctions and encode algebraic structure internal to ∞-categories. They formalize operations that can be repeated and composed—such as taking free objects, forming completions, or extracting core data—while preserving homotopy coherence.

This layer builds directly on adjunctions (Layer 9), extracting the algebraic essence of the interaction between left and right adjoints. Monads and comonads are central to ∞-categorical algebra, descent theory, and higher sheaf theory.

Classical Background

In ordinary category theory, a **monad** on a category C is a triple (T, η, μ) consisting of:

- A functor $T: C \to C$,
- A unit $\eta: \mathrm{id}_C \to T$,
- A multiplication $\mu: T \circ T \to T$,

satisfying associativity and unit laws. Monads arise from adjunctions: if $F \dashv G$, then $T = G \circ F$ is a monad.

Dually, a **comonad** is a triple (G, ε, δ) with:

- A functor $G: C \to C$,

- A counit $\varepsilon\colon G \to \mathrm{id}_C$,
- A comultiplication $\delta\colon G \to G \circ G$,

satisfying coassociativity and counit laws. Comonads arise from the same adjunction via $G = F \circ G$.

∞-Categorical Definition

In ∞-category theory, monads and comonads are defined similarly, but the unit and multiplication (or counit and comultiplication) are **homotopy-coherent transformations**. The associativity and unit laws are satisfied **up to coherent higher homotopies**.

Given an adjunction

$$F\colon C \leftrightarrows D : G,$$

the **monad** $T = G \circ F$ is equipped with:

- A unit transformation $\eta\colon \mathrm{id}_C \to T$,
- A multiplication $\mu\colon T \circ T \to T$,

such that the diagrams expressing associativity and unitality commute up to higher homotopy.

The **comonad** $G = F \circ G$ is similarly equipped with:

- A counit $\varepsilon\colon G \to \mathrm{id}_C$,
- A comultiplication $\delta\colon G \to G \circ G$,

with coherence conditions expressed via simplicial diagrams.

Conceptual Interpretation

Monads are **machines of structure**: they take an object, freely generate structure, and allow that structure to be composed. Comonads are **machines of extraction**: they take an object, expose its internal structure, and allow that exposure to be iterated.

Monads are like **syntax generators**: given a raw input, they produce structured expressions. Comonads are like **parsers**: given a structured object, they extract its components.

In a computational analogy, monads model **effects** (e.g., state, exceptions, input/output), while comonads model **contexts** (e.g., environments, data streams).

Algebras and Coalgebras

A **T-algebra** for a monad T is an object $X \in C$ equipped with a map $T(X) \to X$ satisfying coherence conditions. These algebras encode **structured objects** governed by the monad.

A **G-coalgebra** for a comonad G is an object $X \in C$ equipped with a map $X \to G(X)$, encoding **decomposable objects** governed by the comonad.

In ∞-categories, these structures are defined via **homotopy-coherent diagrams**, and the category of algebras or coalgebras is itself an ∞-category.

Fibrational Perspective

Monads and comonads correspond to **(co)Cartesian fibrations** over the simplex category Δ, encoding the iterative structure of composition. The coherence conditions are expressed via **Segal conditions** on these fibrations.

This perspective connects monads to **operads**, **loop spaces**, and **higher algebra**, revealing their deep geometric and combinatorial structure.

Applications and Significance

- **Sheafification**: Monads encode the process of freely generating sheaves from presheaves.
- **Descent**: Comonads encode the data needed to glue local objects into global ones.
- **Algebraic Geometry**: Monads model derived functors and completion processes.
- **Homotopy Theory**: Monads arise in the study of loop spaces and structured ring spectra.
- **Higher Algebra**: Monads and comonads are foundational in the theory of ∞-operads and ∞-categories of algebras.

Monads and comonads are the **algebraic engines** of ∞-category theory, capturing structure, iteration, and coherence in a homotopical setting.

Summary

- Monads and comonads arise from adjunctions and encode algebraic structure.
- They consist of functors with unit/multiplication or counit/comultiplication transformations.
- In ∞-categories, these transformations are homotopy-coherent.
- Algebras and coalgebras for monads/comonads encode structured and decomposable objects.

- They are central to sheaf theory, descent, higher algebra, and homotopy theory.

Layer 10: Presentable and Accessible ∞-Categories

The Architecture of Manageable Infinity

Purpose and Position in the Tower

This layer introduces **presentable** and **accessible** ∞-categories—two foundational notions that allow us to **tame the vastness** of ∞-categories by identifying those that are **large but controllable**, **structured yet flexible**, and **amenable to universal constructions** like adjunctions, Kan extensions, and sheafification.

Presentable ∞-categories are the **workhorses** of Higher Topos Theory. They are the categories where most of the action happens: spaces, spectra, sheaves, stacks, and derived categories all live here. Accessibility provides the **scaffolding**: it tells us how these categories are built from smaller pieces via filtered colimits. The "workhorse" metaphor is apt—these categories do the heavy lifting of universal constructions, supporting adjoints, limits, colimits, and localizations.

This layer is the gateway to ∞-topoi, model ∞-categories, and the formalization of geometry and logic in the ∞-categorical world.

Classical Background

In ordinary category theory:

- A category is **accessible** if it is generated under filtered colimits by a small subcategory of **compact objects**.
- A category is **presentable** if it is accessible and **cocomplete** (i.e., has all small colimits).

These notions allow one to work with large categories (like the category of sets or modules) while retaining control via small generating data.

∞-Categorical Definition

Let C be an ∞-category.

- C is **accessible** if there exists a regular cardinal κ and a small ∞-category $C_0 \subseteq C$ such that C is the **Ind-completion** of C_0 under κ -filtered colimits:

$C \simeq \mathrm{Ind}\_\kappa(C_0).$

- C is **presentable** if it is accessible and admits **all small colimits**.

This means presentable ∞-categories are built from small pieces using filtered colimits and are closed under the usual universal constructions.

Example: imagine people in a line passing buckets to fill a well. If everyone passes buckets toward the well, the filling proceeds cleanly; this is like a filtered diagram. If people pass buckets randomly, there is no coherent final result. Filtered colimits require a directedness: every pair of objects has a common successor, ensuring the colimit "converges" rather than branching indefinitely. The bucket brigade works because it has this directed structure.

Concrete test: if for any two contributors there is a later contributor who incorporates both their work, the diagram is filtered; without that merging behavior the colimit may not exist.

Conceptual Interpretation

Imagine trying to understand a vast library. Accessibility tells you that the entire library can be reconstructed from a **small collection of foundational texts**, provided you allow yourself to build new books by combining and extending them in a filtered way. Presentability adds that the library is **complete**: it contains all possible compilations, anthologies, and summaries.

Or picture a sprawling city. Accessibility says the city was built from a **finite set of architectural patterns**, repeated and scaled. Presentability says the city has **every kind of building** you might need —schools, hospitals, homes—because it supports all constructions. The mathematical version is more precise: the "architectural patterns" are κ-compact objects that generate everything else via filtered colimits, and "every kind of building" means precisely that all colimits exist. The city grows not randomly but according to these generative principles.

In a biological analogy, accessibility is like a genome: a small set of genes generates the complexity of an organism via regulated expression. Presentability is the full organism: it has all the tissues, organs,

and systems that arise from those genes. But the mathematical version is cleaner—κ-compact objects generate everything via filtered colimits without the messy contingencies of biological development.

Compact Objects and Generators

An object $X \in C$ is ** κ -compact** if the functor

$$\mathrm{Map}_C(X, \text{-})$$

preserves κ -filtered colimits. These compact objects are the **building blocks** of accessible categories.

A presentable ∞-category is generated under colimits by a **small set of compact objects**. This means that every object is a **colimit of compact ones**, and the compact objects **detect equivalences**.

Adjunctions and Presentability

Presentable ∞-categories are **closed under adjunctions**:

- A functor between presentable ∞-categories is a **left adjoint** if and only if it **preserves colimits**.
- A functor is a **right adjoint** if and only if it **preserves limits** and is **accessible**.

This makes presentable ∞-categories the **natural habitat** for Kan extensions, adjunctions, and descent.

Examples

- The ∞-category of spaces S is presentable.
- The ∞-category of spectra Sp is presentable.
- The ∞-category of presheaves $P(C) = \mathrm{Fun}(C^{op}, S)$ is presentable.
- The ∞-category of sheaves on a site is presentable.

These examples show that **most categories of interest in homotopy theory, geometry, and logic are presentable**.

Fibrational and Ind-Categorical Perspective

Using the straightening/unstraightening equivalence, accessible ∞-categories correspond to **fibrations with filtered colimit structure**. The Ind-construction is a **fibrational thickening**: it adds all filtered colimits to a small category, producing a large but controlled ∞-category.

This perspective reveals accessibility as a **geometric expansion**: starting from a small seed, one grows a space of possibilities via filtered diagrams.

Summary

- Accessible ∞-categories are generated under filtered colimits by small subcategories of compact objects.
- Presentable ∞-categories are accessible and cocomplete.
- They are the natural setting for adjunctions, Kan extensions, and sheaf theory.
- Most ∞-categories of interest are presentable.
- Presentability provides a framework for working with large ∞-categories in a controlled, generative way.

Layer 11: Stable ∞-Categories

The Homotopy Theory of Additive and Triangulated Phenomena

Purpose and Position in the Tower

This layer introduces **stable ∞-categories**—∞-categories where the suspension functor is an equivalence. These are the natural home for **homological algebra**, **spectra**, and **derived categories**. Stability provides the framework for studying phenomena that are inherently **linear**, **additive**, or **periodic**.

Building on presentable ∞-categories (Layer 10), stable ∞-categories are those where every object behaves like a spectrum—it can be delooped and suspended infinitely in both directions. This layer is essential before discussing stable phenomena in ∞-topoi, derived algebraic geometry, and higher algebra.

Classical Background

In classical algebra:

- **Abelian categories** have kernels, cokernels, and exact sequences
- **Triangulated categories** have distinguished triangles and suspension functors

- **Chain complexes** can be shifted in degree

Stable ∞-categories unify and enhance these structures:

- They are the ∞-categorical enhancement of triangulated categories
- They automatically have well-behaved mapping cones and homotopy fibers
- They satisfy universal properties that triangulated categories only approximate

Formal Definition

An ∞-category C is **stable** if:

1. **Has zero object**: There exists an object that is both initial and terminal
2. **Has finite limits and colimits**: Pullbacks and pushouts exist
3. **Pullback squares are pushout squares**: A square is a pullback if and only if it is a pushout

Equivalently, C is stable if:

- The suspension functor $\Sigma: C \to C$ is an equivalence
- Every morphism sits in a fiber sequence and cofiber sequence
- The homotopy category hC is triangulated

Conceptual Interpretation

In a stable ∞-category, every phenomenon is **reversible via suspension**:

- Objects can be shifted up or down in degree
- Every morphism has a cone and a fiber
- Exact triangles encode the failure of compositions to be zero

Think of it like an **accounting system where debits and credits balance**—but with a crucial difference:

- Every credit has a corresponding debit
- Transactions can be reversed and re-reversed
- The books always balance via exact triangles

Or like **Fourier analysis**:

- Functions decompose into frequencies
- Operations commute with frequency shifts

- The transform is invertible

The Spectrum Construction

Given a pointed ∞-category C_*, its **stabilization** is:

$$Sp(C) = \lim_{n \to \infty} C_*^{\Sigma^n}$$

This process:

- Takes increasingly suspension-stable objects
- Forms the limit of the tower of suspensions
- Produces the universal stable ∞-category

The stabilization of spaces gives the ∞-category of **spectra**—the fundamental objects of stable homotopy theory.

Examples

- **Spectra**: Sp , the stabilization of pointed spaces
- **Chain complexes**: $Ch(A)$ for an abelian category A
- **Derived categories**: $D(A)$, the ∞-categorical enhancement
- **K-theory spectra**: $K(R)$ for a ring R
- **Stable module categories**: Mod_R^{stable}
- **Motivic spectra**: $SH(S)$ over a scheme S

Each encodes additive or homological phenomena in a homotopy-coherent way.

The Triangulated Structure

Every stable ∞-category has:

- **Distinguished triangles**: Fiber sequences that are also cofiber sequences
- **Suspension functor**: An autoequivalence Σ
- **Exact triangles**: For every morphism $f: X \to Y$, a triangle

$X \xrightarrow{f} Y \to cone(f) \to \Sigma X$

The triangulated structure is **canonical**—it emerges from stability, not imposed as extra data. This solves the coherence problems of classical triangulated categories.

---

t-Structures

A **t-structure** on a stable ∞-category is a way to extract abelian categories:

- Truncation functors $\tau_{\geq n}$ and $\tau_{\leq n}$
- A heart $C^{\heartsuit}$ —an abelian category
- Every object decomposes via its Postnikov tower

Examples:

- The standard t-structure on D(A) has heart A
- The perverse t-structure encodes intersection cohomology
- Motivic t-structures relate to Bloch-Kato conjectures

Stability and Presentability

A stable ∞-category is **presentable** if and only if it is a module category over spectra:

$$C \simeq \mathrm{Mod}_R(\mathrm{Sp})$$

for some ring spectrum R.

This means:

- Every presentable stable ∞-category is tensored over spectra
- Stability and presentability interact perfectly
- The ∞-category of presentable stable ∞-categories is particularly well-behaved

---

Applications and Significance

- **Derived Algebraic Geometry**: Stable ∞-categories of quasi-coherent sheaves
- **Topological Modular Forms**: Spectra with E_∞-ring structure
- **Floer Homology**: Stable ∞-categories in symplectic topology
- **Persistent Homology**: Stability in data analysis
- **Higher K-Theory**: Waldhausen S-construction

Summary

- Stable ∞-categories are those where suspension is an equivalence
- They enhance triangulated categories with full homotopy coherence
- Examples include spectra, chain complexes, and derived categories
- t-structures extract abelian categories from stable ones
- Presentable stable ∞-categories are modules over spectra
- Essential for homological algebra, K-theory, and derived geometry

---

Layer 12: Monoidal ∞-Categories

Multiplicative Structure in the Homotopy-Coherent World

Purpose and Position in the Tower

This layer introduces **monoidal ∞-categories**—∞-categories equipped with a tensor product operation that is associative and unital up to coherent homotopy. This multiplicative structure is essential for:

- Defining algebra objects and module categories
- Constructing tensor products of sheaves
- Formulating topological quantum field theories
- Understanding dualizability and traces

Building on stable ∞-categories (Layer 11), we can now discuss ring spectra, tensor products of chain complexes, and the multiplicative structure underlying much of higher algebra.

Classical Background

In ordinary category theory:

- A **monoidal category** has a tensor product ⊗ and unit object
- **Strict monoidal** means associativity holds exactly
- **Weak monoidal** means associativity holds up to isomorphism
- **Braided** and **symmetric** add commutativity structure

The ∞-categorical version:

- Associativity holds up to coherent homotopy
- Coherence data forms an A_∞-space of operations
- Braiding and symmetry become E_n-structures

Formal Definition

A **monoidal ∞-category** is an ∞-category C equipped with:

1. A tensor product functor $\otimes: C \times C \to C$
2. A unit object $1 \in C$
3. Coherent associativity: $(A \otimes B) \otimes C \simeq A \otimes (B \otimes C)$
4. Coherent unitality: $1 \otimes A \simeq A \simeq A \otimes 1$
5. Higher coherences satisfying all relations

Formally, a monoidal ∞-category is:

- A coCartesian fibration $C^\wedge\otimes \to N(Fin_*)$
- Where Fin_* is the category of pointed finite sets
- The fiber over ⟨ n ⟩ encodes n-ary tensor products

The Hierarchy of Commutativity

Monoidal ∞-categories organize by their level of commutativity:

- **A_∞**: Associative (monoidal ∞-categories)
- **$A_2 = E_1$**: Associative with no higher coherence specified
- **E_2**: Braided monoidal (one-dimensional commutativity)
- **E_3**: Sylleptic monoidal (two-dimensional commutativity)
- **E_∞**: Symmetric monoidal (all dimensional commutativity)

Each level adds coherence in higher dimensions. E_∞ is the limit where all ways of reordering tensor products are coherently equivalent.

Examples

- **Spaces with Cartesian product**: (S, ×) is symmetric monoidal
- **Spectra with smash product**: (Sp, ∧) is symmetric monoidal

- **Chain complexes**: $(Ch(R), \otimes\_R)$ is symmetric monoidal
- **Derived category**: $(D(R), \otimes^{\wedge}L_R)$ with derived tensor
- **Representations**: $(Rep(G), \otimes)$ for a group G
- **Cobordisms**: $(Cob_n, \sqcup)$ with disjoint union

Algebra Objects

In a monoidal ∞-category C , an **algebra object** is:

- An object A with multiplication $A \otimes A \to A$
- Unit $1 \to A$
- Coherent associativity and unitality

The ∞-category of algebra objects:

$$\mathrm{Alg}(C) = \mathrm{Fun}^{\mathrm{lax}}(N(\Delta^{\mathrm{op}}), C)$$

Examples:

- Ring spectra in Sp
- DG-algebras in chain complexes
- Monoids in spaces

Dualizable Objects

An object X in a monoidal ∞-category is **dualizable** if there exists:

- A dual object $X^{\wedge}\vee$
- Evaluation: $X \otimes X^{\wedge}\vee \to 1$
- Coevaluation: $1 \to X^{\wedge}\vee \otimes X$
- Triangle identities up to coherent homotopy

Dualizability characterizes:

- Finite objects in many contexts
- Objects with good finiteness properties
- The domain of traces and Euler characteristics

The Cobordism Hypothesis (1-dimensional)

For 1-dimensional TQFTs:

- A TQFT is a symmetric monoidal functor $\mathsf{Cob}_1 \to \mathsf{C}$
- The cobordism hypothesis states: TQFTs correspond to dualizable objects
- Every 1D TQFT is determined by a dualizable object (the value on the circle)

This pattern extends to all dimensions, with (∞,n)-categories replacing ∞-categories.

Tensor Products and Module Categories

Given a monoidal ∞-category C :

- **Module categories**: ∞-categories with C -action
- **Bimodule categories**: Actions on both sides
- **Relative tensor product**: M $\otimes$_{C} N

This forms the basis for:

- Morita theory
- Base change in algebra
- Categorical traces

Applications and Significance

- **Tensor Triangulated Geometry**: Reconstruction from monoidal data
- **Tannaka Duality**: Recovering groups from their representations
- **Factorization Algebras**: E_n-algebras encode n-dimensional locality
- **Chromatic Homotopy**: Monoidal structures on spectra
- **Higher Algebra**: ∞-operads and their algebras

Summary

- Monoidal ∞-categories have tensor products up to coherent homotopy
- The E_n hierarchy measures levels of commutativity
- Algebra objects generalize rings and monoids
- Dualizable objects satisfy finiteness conditions

- Essential for TQFTs, higher algebra, and tensor triangulated geometry
- Module categories and bimodules organize relative structures

---

Layer 13: ∞-Topoi

The Homotopy-Coherent Generalization of Grothendieck Topoi

Purpose and Position in the Tower

This layer introduces ∞-**topoi**, the ∞-categorical analogs of Grothendieck topoi. These are ∞-categories that behave like categories of sheaves of spaces on a site, but with full homotopical generality. They are the **natural setting** for doing geometry, logic, and cohomology in the ∞-categorical world.

∞-topoi unify and generalize classical sheaf theory, higher stacks, and homotopy type theory. They are **presentable**, **locally Cartesian closed**, and satisfy descent. This layer builds directly on presentable ∞-categories (Layer 11), Kan extensions (Layer 8), and adjunctions (Layer 9), and prepares the ground for structured notions like geometric morphisms and internal logic.

Classical Background

A **Grothendieck topos** is a category of sheaves on a site, satisfying:

- It is **presentable**.
- It has **finite limits**.
- Colimits are **universal** (commute with pullbacks).
- It is **locally Cartesian closed**.
- It satisfies **descent**: sheaves glue along covers.

These properties make topoi suitable for modeling spaces, logic, and cohomology.

∞-Categorical Definition

An ∞-category X is an ∞-**topos** if it satisfies:

1. **Presentability**: X is presentable.

2. **Finite Limits**: X admits all finite limits.
3. **Colimits are Universal**: Colimits commute with pullbacks.
4. **Local Cartesian Closure**: For every object $X \in X$, the slice category $X_{/X}$ is Cartesian closed.
5. **Descent**: X satisfies effective epimorphic descent for hypercovers.

These conditions ensure that X behaves like a category of sheaves of spaces on a site, but with full ∞-categorical structure.

Conceptual Interpretation

An ∞-topos is like a **landscape of spaces** where every local patch can be glued into a global whole, and every construction respects homotopy. It is a **homotopy-coherent sheaf theory**: not just sets glued along covers, but spaces glued along higher-dimensional overlaps.

Or think of an ∞-topos as a **semantic universe**: it supports internal logic, quantification, and type theory, but with homotopical meaning. Truth values are spaces (not just true/false but entire homotopy types), propositions are types (with proofs forming paths), and logical operations are homotopy-theoretic constructions. This isn't metaphorical—the Curry-Howard correspondence extends literally to homotopy type theory, where types ARE spaces and proofs ARE paths.

In a geometric analogy, an ∞-topos is a **space of spaces**: a category where each object is a space, and morphisms preserve spatial structure up to homotopy.

Models of ∞-Topoi

There are several equivalent models of ∞-topoi:

- **Sheaves of spaces on a site**: $Sh(C, \tau)$, where τ is a Grothendieck topology.
- **Left exact localizations of presheaf ∞-categories**: $P(C) \to X$.
- **Accessible left exact reflective subcategories** of presheaf categories.
- **Higher sheaf theory**: satisfying descent for hypercovers.

These models allow one to construct ∞-topoi from sites, localizations, and descent conditions.

Universal Properties

∞-topoi are **universal among categories satisfying descent**. They are the **free cocompletion** of a site under colimits and descent. This makes them the natural setting for:

- **Sheafification**.
- **Higher stacks**.
- **Cohomology theories**.
- **Homotopy type theory**.

They are also **stable under base change**, **closed under limits**, and **support internal logic**.

Geometric Morphisms

A **geometric morphism** between ∞-topoi is a pair of adjoint functors

$$f^* : Y \leftrightarrows X : f_*,$$

where f^* is left exact (preserves finite limits). These morphisms generalize continuous maps between spaces and allow one to **transport sheaves**, **compare topoi**, and **define base change**.

Applications and Significance

- **Derived Algebraic Geometry**: ∞-topoi model derived stacks and moduli spaces.
- **Homotopy Type Theory**: ∞-topoi provide models for type-theoretic semantics.
- **Cohomology and Descent**: ∞-topoi support higher cohomology theories and descent.
- **Logic and Semantics**: ∞-topoi internalize logical operations and quantification.

∞-topoi are the **foundational environment** for doing geometry, logic, and homotopy theory in the ∞-categorical world.

Summary

- ∞-topoi generalize Grothendieck topoi to the ∞-categorical setting.
- They are presentable, locally Cartesian closed, and satisfy descent.
- They model sheaves of spaces, higher stacks, and homotopy type theory.
- They support geometric morphisms, internal logic, and cohomology.
- ∞-topoi are the natural habitat for structured homotopy-theoretic mathematics.

Layer 14: Geometric Morphisms and Base Change

The Structural Bridges Between ∞-Topoi

Purpose and Position in the Tower

This layer introduces **geometric morphisms**—the fundamental maps between ∞-topoi—and their behavior under **base change**. These morphisms are not mere functors; they are **structured correspondences** that preserve the rich geometry and logic encoded in ∞-topoi.

Geometric morphisms are the ∞-**categorical analogs of continuous maps between topological spaces**, or morphisms of schemes in algebraic geometry. They allow one to **transport sheaves, cohomology, and descent data** across different ∞-topoi. Base change, in turn, describes how these transports behave under pullbacks—ensuring coherence and compatibility in fibered constructions.

This layer builds on presentability (Layer 11) and ∞-topoi (Layer 12), and sets the stage for descent theory, étale geometry, and the formalization of stacks and sheaves in derived settings.

Classical Background

In classical topos theory, a **geometric morphism** between topoi

$$f: X \to Y$$

consists of a pair of adjoint functors

$$f^*: Y \leftrightarrows X : f_*,$$

where:

- f^* is **left exact** (preserves finite limits),
- f_* is **right adjoint** to f^*.

This structure reflects the idea that f^* pulls sheaves back along f, while f_* pushes them forward. The left exactness of f^* ensures that the logical structure of the topos is preserved.

∞-Categorical Definition

In ∞-category theory, a **geometric morphism** between ∞-topoi

$$f: X \to Y$$

is a pair of adjoint functors

$$f^*: Y \leftrightarrows X : f_*,$$

such that:

- f^* is **left exact** (preserves finite limits),
- f_* is **accessible** and **preserves colimits**.

This definition ensures that the morphism respects both the **logical structure** (via limits) and the **homotopical structure** (via colimits) of the ∞-topoi.

Conceptual Interpretation

Think of ∞-topoi as **universes of sheaves**—each encoding a geometry, a logic, and a homotopy theory. A geometric morphism connects these universes, allowing one to **translate structure** from one to another. The "portal" metaphor captures something real: geometric morphisms preserve finite limits (logical structure) while allowing controlled change of base (geometric structure).

Or imagine two ecosystems. A geometric morphism is a **migration pathway**: species (sheaves) can move from one ecosystem to another, and the pathway preserves ecological relationships (limits) and population dynamics (colimits). But the mathematical version is more controlled—the inverse image functor f* preserves all finite limits exactly, while the direct image f_* has a left adjoint ensuring it respects the ambient logic.

In a computational analogy, ∞-topoi are **data environments**, and geometric morphisms are **structured APIs**: they allow data to be pulled and pushed between environments while preserving integrity and coherence.

Base Change and Pullbacks

Given a geometric morphism f: X → Y and a morphism g: Z → Y , one can form the **pullback** ∞-topos

$$X \times_Y Z,$$

and obtain a **base change square**:

$$
\begin{array}{ccc}
X \times_Y Z & \xrightarrow{f'} & Z \\
\downarrow & & \downarrow g \\
X & \xrightarrow{f} & Y
\end{array}
$$

This square expresses how sheaves and structure **transport coherently** across fibered diagrams. The base change formula ensures that **pulling back then pushing forward** is equivalent to **pushing forward then pulling back**—a fundamental principle in cohomology and descent.

Fibrational Perspective

Using the straightening/unstraightening equivalence, geometric morphisms correspond to **Cartesian fibrations** between ∞-topoi. The base change square becomes a **fibered diagram of fibrations**, and the coherence conditions become **homotopy pullbacks**.

This perspective reveals geometric morphisms as **structured lifts** of ∞-topoi over a base, and base change as **reindexing** these lifts along morphisms.

Applications and Significance

- **Descent Theory**: Geometric morphisms encode how local data glues into global data.
- **Étale Geometry**: Morphisms of ∞-topoi model étale maps of schemes and stacks.
- **Sheaf Theory**: Base change describes how sheaves transform under pullbacks.
- **Cohomology**: Base change formulas are essential in computing derived functors.
- **Stacks and Moduli**: Geometric morphisms formalize the structure of stacks and moduli spaces.

Geometric morphisms and base change are the **structural backbone** of ∞-topos theory. They allow one to **navigate the landscape of ∞-topoi**, transporting structure, logic, and geometry in a coherent and universal way.

Summary

- Geometric morphisms are adjoint pairs of functors between ∞-topoi, with left exact left adjoints.
- They preserve both logical and homotopical structure.
- Base change describes how geometric morphisms behave under pullbacks.

- These constructions are essential in descent, sheaf theory, cohomology, and derived geometry.
- Geometric morphisms are the structured bridges between ∞-topoi.

Layer 15: Descent and Hypercovers

Gluing Homotopy-Coherent Data Across ∞-Categorical Landscapes

Purpose and Position in the Tower

This layer introduces **descent** and **hypercovers**—the machinery that allows one to **reconstruct global data from local pieces** in a homotopy-coherent way. Descent is the ∞-categorical generalization of the sheaf condition, and hypercovers are the **refined scaffolding** that make descent possible in higher dimensions.

In the tower of Higher Topos Theory, this layer builds on ∞-topoi (Layer 12) and the theory of sheaves (Layer 13), extending them to **homotopical gluing** across complex covers. It is essential for derived algebraic geometry, stack theory, and the formalization of cohomological and geometric invariants in ∞-categories.

Classical Background

In classical sheaf theory, descent refers to the ability to **glue local sections** over a cover to obtain a global section. A presheaf F on a topological space satisfies descent if, for every open cover $\{U_i\}$, the diagram

$$F(U) \to \prod_i F(U_i) \rightrightarrows \prod_{i,j} F(U_i \cap U_j)$$

is an equalizer.

In higher category theory, this condition is **too rigid**. Instead, one replaces equalizers with **homotopy limits**, and covers with **hypercovers**—simplicial objects that encode **higher-order intersections and refinements**.

∞-Categorical Definition of Descent

Let X be an ∞-topos and $F: U^{op} \to S$ a presheaf of spaces on a site U . Then F satisfies **descent** if, for every **hypercover** $U\_\bullet \to U$, the canonical map

$$F(U) \to \lim_{[n] \in \Delta} F(U_n)$$

is an equivalence in S .

This expresses that the value of F on U can be **reconstructed from its values on the hypercover** $U\_\bullet$, using a homotopy limit over the simplicial diagram.

Hypercovers: The Geometry of Gluing

A **hypercover** is a simplicial object $U\_\bullet$ in a site U such that:

- $U_0 \to U$ is a cover,
- $U_1 \to U_0 \times_U U_0$ is a cover,
- $U_2 \to$ matching object is a cover,
- and so on.

Each level of the hypercover refines the previous one, encoding **higher-order overlaps and compatibility conditions**. Hypercovers generalize Čech nerves and allow for **homotopy-coherent descent**.

They are the **scaffolding** on which ∞-categorical sheaves are built.

Conceptual Interpretation

Imagine trying to reconstruct a 3D object from 2D slices. A Čech cover gives you the slices, but a hypercover gives you **slices, overlaps, overlaps of overlaps**, and so on—enough data to reconstruct the full object **up to homotopy**. The key insight: unlike physical reconstruction where you might need infinitely many slices for perfect accuracy, mathematical hypercovers capture exactly the homotopy type with a simplicial object.

Or picture a conversation among many people. A Čech cover records who spoke to whom. A hypercover records **who spoke to whom about what**, **how their stories align**, and **how those alignments themselves align**—capturing the full coherence of the dialogue. But unlike human

conversation which can be inconsistent, mathematical hypercovers satisfy strict simplicial identities ensuring perfect coherence.

In a musical analogy, a Čech cover gives you the notes played by each instrument. A hypercover gives you the **harmonies**, **counterpoints**, and **resolutions**—the full structure of the composition.

Descent as a Homotopy Limit Condition

Descent is not a property of individual objects—it is a **global coherence condition**. It says that the value of a sheaf on a space is the **homotopy limit** of its values on a hypercover.

This replaces the rigid equalizer condition of classical sheaf theory with a **flexible, homotopy-invariant condition** that respects higher morphisms and coherence.

Applications and Significance

- **Derived Algebraic Geometry**: Descent allows one to glue derived schemes and stacks from local models.
- **Higher Stacks**: Hypercovers are used to define ∞-stacks and their moduli.
- **Cohomology**: Descent expresses how cohomological invariants are computed from local data.
- **Topological Field Theory**: Descent encodes how local observables determine global states.
- **Model ∞-Categories**: Hypercovers are used to define fibrant objects and weak equivalences.

Descent and hypercovers are the **language of gluing** in ∞-category theory. They allow one to reconstruct global structure from local pieces, respecting all levels of homotopy coherence.

Summary

- Descent is the condition that a presheaf can be reconstructed from its values on a hypercover via a homotopy limit.
- Hypercovers are simplicial objects that encode higher-order overlaps and refinements.
- They generalize Čech covers and enable homotopy-coherent gluing.
- Descent is essential for sheaf theory, stacks, cohomology, and derived geometry.
- It replaces equalizers with homotopy limits, making gluing flexible and robust.

Layer 16: ∞-Categorical Giraud Axioms

Characterizing ∞-Topoi via Homotopy-Coherent Logical Structure

---

Purpose and Position in the Tower

This layer crystallizes the notion of an ∞-**topos** by presenting the ∞-**categorical analogues of Giraud's axioms**—a set of structural conditions that characterize ∞-topoi as **homotopy-coherent generalizations of Grothendieck topoi**. These axioms provide a **synthetic description** of ∞-topoi, independent of sites or sheaf conditions, and reveal them as **logically structured universes** of homotopy types.

This layer builds on descent and hypercovers (Layer 14), presentability (Layer 11), and the theory of sheaves (Layer 13), and prepares the ground for internal logic, cohomology, and geometric morphisms in ∞-topoi.

Classical Background

In classical category theory, **Giraud's axioms** characterize Grothendieck topoi as categories satisfying:

1. **Presentability**: The category is cocomplete and generated under colimits by a small set of objects.
2. **Colimits commute with pullbacks**.
3. **Effective epimorphisms are stable under pullback**.
4. **Every equivalence relation has a quotient**.

These axioms abstract the behavior of sheaf categories and allow one to define topoi **without reference to a site**.

∞-Categorical Giraud Axioms

Let X be a presentable ∞-category. Then X is an ∞-**topos** if it satisfies the following ∞-categorical Giraud axioms:

1. **Presentability**: X is presentable.
2. **Colimits are universal**: Colimits in X commute with pullbacks.

3. **Effective epimorphisms are stable under pullback**: If f: X → Y is an effective epimorphism, then for any map Z → Y , the pullback X ×$_Y$ Z → Z is also an effective epimorphism.

4. **Groupoid objects are effective**: Every groupoid object in X has a colimit that realizes it as a quotient.

These axioms ensure that X behaves like a category of sheaves of spaces: it supports descent, gluing, and internal homotopy theory.

Conceptual Interpretation

Imagine an ∞-topos as a **mathematical ecosystem**: presentability ensures it has enough resources (objects and colimits), universality of colimits ensures that constructions are **stable under change of context**, and effectiveness ensures that **relations and symmetries can be resolved** into concrete objects.

Or picture a legal system: presentability is the constitution (defining the scope), universality of colimits is the consistency of laws across jurisdictions, and effectiveness is the ability to **resolve disputes** (equivalence relations) into verdicts (quotients).

In a computational analogy, an ∞-topos is like a **type-theoretic universe**: presentable, stable under substitution (pullbacks), and capable of resolving equivalence types into canonical representatives.

Effective Epimorphisms and Groupoid Objects

An **effective epimorphism** is a morphism that behaves like a surjection: it can be recovered from its **Čech nerve** via a colimit. In ∞-categories, this means the colimit of the Čech nerve of f: X → Y is equivalent to Y .

A **groupoid object** is a simplicial object satisfying Segal conditions and invertibility. Effectiveness means that such objects **present quotients**: their colimit realizes the groupoid as a single object modulo its symmetries.

These conditions ensure that **homotopy equivalence relations can be resolved**—a key feature of ∞-topoi.

Universality of Colimits

This axiom ensures that **colimits behave well under base change**. If you compute a colimit in one context and then change the base (via pullback), the result is the same as computing the colimit in the new context.

This is essential for **descent**, **sheafification**, and **internal logic**: it guarantees that constructions are **stable and coherent** across the ∞-topos.

Synthetic vs. Site-Based Definitions

The Giraud axioms provide a **synthetic definition** of ∞-topoi: they describe intrinsic properties of the category, without reference to a site or presheaf construction.

This is analogous to defining a manifold via charts (site-based) versus defining it via axioms of smoothness and local Euclidean structure (synthetic).

Lurie proves that these axioms are **equivalent** to the site-based definition: every ∞-topos is equivalent to a left exact localization of a presheaf ∞-category.

Applications and Significance

- **Sheaf Theory**: The axioms characterize categories of sheaves of spaces.
- **Homotopy Type Theory**: ∞-topoi are models of homotopy type theory.
- **Derived Geometry**: ∞-topoi serve as ambient categories for derived stacks and moduli.
- **Cohomology**: The axioms ensure that cohomological descent and gluing are valid.
- **Internal Logic**: ∞-topoi support internal languages and logical structure.

The ∞-categorical Giraud axioms are the **blueprint for homotopical geometry and logic**. They define the structural integrity of ∞-topoi and ensure that they support the full machinery of Higher Topos Theory.

Summary

- ∞-topoi are presentable ∞-categories satisfying universal colimits, effective epimorphisms, and effective groupoid objects.
- These axioms generalize Giraud's classical axioms to the homotopy-coherent setting.
- They provide a synthetic characterization of ∞-topoi, independent of sites.

- They ensure that ∞-topoi support descent, gluing, internal logic, and cohomology.
- ∞-Categorical Giraud axioms are the structural foundation of Higher Topos Theory.

Layer 17: Geometric Morphisms and Logical Structure

Bridges Between ∞-Topoi and the Flow of Homotopy-Coherent Truth

Purpose and Position in the Tower

This layer introduces **geometric morphisms**—the fundamental notion of **structure-preserving maps between ∞-topoi**. These morphisms are the **channels through which logic, geometry, and homotopy flow** between different ∞-categorical universes. They encode how one ∞-topos can be interpreted inside another, and how internal languages and constructions migrate across contexts.

Geometric morphisms are the ∞-**categorical analogues of continuous maps between spaces**, or logical interpretations between theories. They are central to the internal logic of ∞-topoi, to the semantics of homotopy type theory, and to the gluing and descent of geometric data.

This layer builds on the ∞-categorical Giraud axioms (Layer 15), descent (Layer 14), and the theory of sheaves and ∞-topoi (Layers 12–13), and prepares the ground for internal language, modalities, and higher logical structure.

Classical Background

In classical topos theory, a **geometric morphism** between topoi

$$f: X \to Y$$

consists of a pair of adjoint functors

$$f^*: Y \leftrightarrows X : f_*,$$

where:

- f^* is **left exact** (preserves finite limits),
- f_* is its **right adjoint**.

This structure models **logical interpretation**: f^* pulls back predicates, and f_* pushes forward truth values. The left exactness of f^* ensures that logical structure is preserved.

∞-Categorical Definition

In ∞-category theory, a **geometric morphism** between ∞-topoi

$$f: X \to Y$$

is a pair of adjoint functors

$$f^*: Y \leftrightarrows X : f_*,$$

such that:

- f^* is **left exact** (preserves finite limits),
- f_* is **accessible** and **preserves colimits**.

This definition ensures that f^* preserves the **logical structure** of Y when interpreted in X , and that f_* reflects the **semantic content** of X back into Y .

Conceptual Interpretation

Imagine two ∞-topoi as **languages**: one geometric morphism is a **translation protocol** that preserves meaning, structure, and inference. The left exactness of f^* ensures that **logical truths** in Y remain valid when interpreted in X .

Or picture two ecosystems: a geometric morphism is a **migration corridor** that allows species (objects) to move between habitats (∞-topoi), preserving ecological relationships (limits and colimits).

In a computational analogy, geometric morphisms are **type-theoretic embeddings**: they allow one type system to be interpreted inside another, preserving logical operations and constructions.

Internal Logic and Modalities

Geometric morphisms are the **semantic backbone** of internal logic in ∞-topoi. They allow one to define:

- **Modalities**: endofunctors on an ∞-topos that reflect logical operations (e.g., necessity, possibility).
- **Subtopoi**: full subcategories defined by reflective localizations, corresponding to modal logics.
- **Truth values**: objects in an ∞-topos that represent propositions, with logical operations defined via limits and colimits.

This internal logic is **homotopy-coherent**: it respects higher morphisms, equivalences, and descent.

Base Change and Fibered ∞-Topoi

Geometric morphisms support **base change**: given a morphism f: X → Y , one can define **fibered ∞-topoi** over Y , and study how objects in X vary over Y .

This is essential for **derived geometry**, **stack theory**, and **cohomological descent**: it allows one to track how geometric and logical data evolve across spaces.

Examples and Applications

- **Sheafification**: The inclusion of presheaves into sheaves is a geometric morphism.
- **Étale Topoi**: Morphisms between étale ∞-topoi reflect geometric maps between schemes.
- **Type Theory**: Interpretations of homotopy type theory in ∞-topoi are geometric morphisms.
- **Modal Logic**: Modalities in ∞-topoi arise from geometric morphisms and subtopoi.
- **Descent**: Geometric morphisms encode how local data glues into global structure.

Geometric morphisms are the **bridges between ∞-categorical worlds**. They preserve structure, enable interpretation, and support the flow of homotopy-coherent logic.

Summary

- A geometric morphism between ∞-topoi is a pair of adjoint functors $f^* \dashv f_*$, with f^* left exact.
- They preserve logical structure and support internal languages.
- They enable base change, descent, and interpretation of type theories.
- Geometric morphisms are the semantic and structural maps between ∞-topoi.

- They are essential for derived geometry, stack theory, and homotopy type theory.

Layer 18: Modalities and Subtopoi

The ∞-Categorical Universe of Modal Logic, Reflective Structure, and Internal Worlds

---

Purpose and Position in the Tower

This layer explores **modalities**—endofunctors on ∞-topoi that encode **logical operations**, **structural reflections**, and **internal perspectives**—and their deep connection to **subtopoi**, which represent **internal worlds** or **modal contexts** within an ∞-topos.

Modalities are not limited to classical notions like necessity and possibility. In the ∞-categorical setting, they encompass a **vast landscape** of logical operators: truncation, localization, cohesion, differential structure, and more. Each modality corresponds to a **reflective subtopos**, and together they form a **modal logic of homotopy types**.

This layer builds on geometric morphisms (Layer 16), ∞-topoi (Layer 12), and internal logic, and opens the door to **type-theoretic semantics**, **cohesive homotopy theory**, and **modal homotopy type theory**.

Classical Background

In classical topos theory, a **subtopos** is a full subcategory of a topos that is **reflective**: the inclusion functor has a left adjoint. This left adjoint is a **localization** or **modality**, and it preserves finite limits.

Modalities correspond to **logical operations**: for example, the double-negation modality reflects classical logic inside intuitionistic logic.

∞-Categorical Definition of Modalities

Let X be an ∞-topos. A **modality** is a **left exact reflective localization**

$$L: X \to X$$

with unit η: id → L , such that:

- L preserves finite limits,
- The essential image of L is a **reflective subtopos** $X_L \subseteq X$,
- The unit η exhibits each object as **approximated by its modal reflection**.

Each modality defines a **modal operator** in the internal logic of X , and the subtopos X_L represents the **world where that modality holds**.

Conceptual Interpretation

Modalities are like **filters on reality**: they allow you to view the ∞-topos through a particular lens—truncated, cohesive, infinitesimal, or discrete. Each lens reveals a different internal world, governed by its own logic. The key mathematical property is that these "filters" are idempotent monads that preserve finite limits, making them much more structured than optical filters.

Or picture a **multi-layered simulation**: each modality defines a layer of abstraction. The reflective subtopos is the simulation at that layer, and the modality is the projection from full reality to that layer.

In a philosophical analogy, modalities are **ways of knowing**: necessity, possibility, knowledge, belief, time, space, continuity, and infinitesimality. Each defines a mode of truth, and the subtopos is the realm where that mode governs. But unlike philosophical modalities which are often vague, mathematical modalities are precise operators satisfying idempotence and preservation of finite limits.

Examples of Modalities

- **n-Truncation**: Reflects objects to their n-truncated approximations (removing homotopy above level n).
- **Shape Modality**: In cohesive ∞-topoi, reflects objects to their underlying homotopy type.
- **Flat Modality**: Reflects objects to discrete types (no cohesion).
- **Infinitesimal Modality**: In differential cohesion, reflects objects to infinitesimal neighborhoods.
- **Double-Negation Modality**: Reflects intuitionistic logic into classical logic.
- **Localization Modalities**: Reflect objects with respect to a homology theory or a class of morphisms.

Each of these defines a **subtopos** where certain logical or geometric properties hold universally.

Modal Logic in ∞-Topoi

The internal logic of an ∞-topos supports **modal operators**:

- □ ("necessarily") corresponds to a modality L ,
- ◇ ("possibly") corresponds to the dual coreflection (if it exists),
- Modalities can be **composed**, **nested**, and **interacted** to form a rich modal algebra.

This logic is **homotopy-coherent**: modal operators act on types (objects), propositions (subobjects), and higher morphisms, preserving structure up to equivalence.

Subtopoi as Modal Worlds

Each reflective subtopos X_L is a **modal world**: a full subcategory where the modality L acts as identity. These subtopoi are:

- **Closed under limits and colimits**,
- **Stable under base change**,
- **Interpretable as internal universes**.

They allow one to **reason internally** about modal truths, and to **transport constructions** across modal contexts.

Applications and Significance

- **Homotopy Type Theory**: Modalities define type-theoretic operators like truncation, cohesion, and differential structure.
- **Cohesive Homotopy Theory**: Modalities encode spatial and continuous structure.
- **Synthetic Differential Geometry**: Infinitesimal modalities define tangent spaces and jets.
- **Logical Semantics**: Modalities model epistemic, temporal, and deontic logics.
- **Derived Geometry**: Modalities reflect derived structure, singularities, and formal neighborhoods.

Modalities and subtopoi are the **logical and geometric lenses** of Higher Topos Theory. They allow one to **navigate, interpret, and construct** within the vast landscape of ∞-categorical worlds.

Summary

- Modalities are left exact reflective localizations of ∞-topoi.
- Each modality defines a subtopos: a modal world where certain truths hold.
- Modalities encode logical operations: truncation, cohesion, infinitesimality, etc.
- The internal logic of ∞-topoi supports a rich modal algebra.
- Modalities are essential for type theory, geometry, logic, and semantics.

Layer 19: Internal Language and Type-Theoretic Semantics

The Syntax of Homotopy-Coherent Truth and the Semantics of ∞-Topoi

---

Purpose and Position in the Tower

This layer unveils the **internal language** of ∞-topoi—a homotopy-coherent generalization of logical syntax—and its deep connection to **type-theoretic semantics**. It reveals ∞-topoi not just as categories of spaces or sheaves, but as **semantic universes** for **homotopy type theory**, **modal logic**, and **higher-order reasoning**.

This layer builds on modalities and subtopoi (Layer 17), geometric morphisms (Layer 16), and the structural foundation of ∞-topoi (Layers 12–15). It is the bridge between **syntax and semantics**, between **proof theory and homotopy theory**, and between **logical inference and geometric construction**.

Classical Background

In classical topos theory, the **internal language** of a topos is a form of **intuitionistic higher-order logic**. It allows one to reason about objects and morphisms **as if they were sets and functions**, using logical connectives, quantifiers, and equality.

This internal language is **sound and complete** with respect to the categorical semantics: logical formulas correspond to subobjects, and proofs correspond to morphisms.

∞-Categorical Internal Language

In ∞-topoi, the internal language is a **dependent type theory** enriched with **homotopy-coherent structure**. It includes:

- **Types**: Objects of the ∞-topos.
- **Terms**: Morphisms between objects.
- **Dependent Types**: Families of objects varying over a base.
- **Identity Types**: Paths or equivalences between terms.
- **Higher Inductive Types**: Types defined by generators and relations, including paths.
- **Modal Types**: Types governed by modalities (e.g., truncated, cohesive, infinitesimal).

This language is **homotopy type theory (HoTT)**: a type theory where equality is interpreted as **paths**, and logical operations are interpreted via **limits, colimits, and modalities**.

Type-Theoretic Semantics in ∞-Topoi

An ∞-topos provides a **model of homotopy type theory**:

- **Types** are interpreted as objects.
- **Terms** are interpreted as morphisms.
- **Contexts** are interpreted as slice ∞-categories.
- **Judgments** are interpreted as diagrams.
- **Proofs** are interpreted as homotopies.

This semantics is **sound and complete**: every derivable judgment in HoTT corresponds to a construction in the ∞-topos, and vice versa.

Moreover, modalities in the ∞-topos correspond to **modal type operators** in the type theory, enabling **modal HoTT**: a type theory with operators for necessity, possibility, truncation, cohesion, and more.

Conceptual Interpretation

Imagine an ∞-topos as a **semantic universe**, and its internal language as the **grammar of that universe**. Every object is a type, every morphism a term, and every homotopy a proof of equality. The type theory is the **syntax**, and the ∞-topos is the **semantics**. This isn't just analogy—the correspondence between homotopy type theory and ∞-topoi is mathematically precise, with types interpreted as objects and identity types as path spaces.

Or picture a programming language: the internal language is the **source code**, and the ∞-topos is the **runtime environment**. Every type is a data structure, every term a function, and every homotopy a transformation. But the mathematical correspondence is tighter than in programming—the semantics is fully determined by the categorical structure, with no undefined behavior or implementation-dependent features.

In a philosophical analogy, the internal language is **thought**, and the ∞-topos is **being**. Type-theoretic semantics is the **correspondence between ideas and reality**, structured by homotopy.

Modal Type Theory and Subtopoi

Modalities in the ∞-topos correspond to **modal type operators**:

- □**A**: The reflection of type A under a modality.
- ◇**A**: The coreflection (if it exists).
- **Flat A**: The discrete version of A .
- **Sharp A**: The cohesive version of A .
- **Infinitesimal A**: The infinitesimal neighborhood of A .

These operators allow one to **reason within modal worlds**, defined by subtopoi, and to **transport logic across modalities**.

Applications and Significance

- **Homotopy Type Theory**: ∞-topoi are models of HoTT, supporting identity types, higher inductive types, and univalence.
- **Modal Logic**: Modalities define modal type theories with rich logical structure.
- **Synthetic Geometry**: Internal languages support synthetic differential geometry, cohesion, and spatial reasoning.
- **Proof Assistants**: Systems like Agda, Coq, and Lean can be interpreted in ∞-topoi.
- **Mathematical Foundations**: Type-theoretic semantics provides a foundation for mathematics based on homotopy and logic.

The internal language and type-theoretic semantics of ∞-topoi are the **syntax-semantics interface** of Higher Topos Theory. They allow one to **reason, compute, and construct** within homotopy-coherent worlds.

Summary

- ∞-topoi support an internal language: a homotopy-enriched dependent type theory.
- This language is interpreted via type-theoretic semantics: types as objects, terms as morphisms, proofs as homotopies.
- Modalities define modal type operators and subtopoi as modal worlds.
- ∞-topoi are models of homotopy type theory and modal logic.
- Internal language and semantics unify logic, geometry, and computation in ∞-categories.

Layer 20: Higher Sheaf Conditions and Stack Semantics

From Local Data to Global Geometry in the ∞-Categorical World

---

Purpose and Position in the Tower

This layer deepens the theory of sheaves and descent by introducing **higher sheaf conditions** and the semantics of **stacks** in the ∞-categorical setting. It refines the notion of sheafification to accommodate **homotopy-coherent gluing**, and generalizes the concept of a sheaf to that of a **stack**—a presheaf of ∞-groupoids satisfying descent with respect to hypercovers.

This layer builds on descent and hypercovers (Layer 14), ∞-topoi (Layer 12), and geometric morphisms (Layer 16), and is foundational for **moduli problems**, **derived algebraic geometry**, and **homotopy type theory**. It is where the machinery of Higher Topos Theory begins to **interface with geometry, deformation theory, and classification problems**.

Classical Background

In classical algebraic geometry and topology:

- A **sheaf** is a presheaf that satisfies descent for open covers.
- A **stack** is a presheaf of groupoids (or categories) that satisfies descent for **isomorphisms**, not just for sets.

- Stacks are used to model **moduli spaces** where objects have **automorphisms**—e.g., vector bundles, principal bundles, or complex structures.

In higher category theory, stacks become **presheaves of ∞-groupoids** (or ∞-categories), and descent is enforced via **homotopy limits over hypercovers**.

∞-Categorical Sheaf Conditions

Let C be a site (a category with a Grothendieck topology), and let

$$F: C^{op} \to S$$

be a presheaf of spaces. Then F is a **sheaf** if for every **covering sieve** $R \subseteq \mathrm{Hom}(-, U)$, the canonical map

$$F(U) \to \lim_{f \in R} F(\mathrm{dom}(f))$$

is an equivalence in S .

Equivalently, for every **hypercover** U_• → U , the canonical map

$$F(U) \to \lim_{[n] \in \Delta} F(U_n)$$

is an equivalence in S .

This is the **higher sheaf condition**: it replaces equalizers with homotopy limits and Čech covers with hypercovers, ensuring that gluing works **up to coherent homotopy**.

Stacks in ∞-Categories

A **stack** is a presheaf of ∞-groupoids (or ∞-categories) satisfying the higher sheaf condition. That is, it assigns to each object $U \in C$ an ∞-groupoid $F(U)$, and satisfies descent for hypercovers.

Stacks encode **moduli problems**: they classify objects (like bundles or sheaves) **up to equivalence**, and track their **automorphisms and higher symmetries**.

In the ∞-categorical setting, stacks are **objects of an ∞-topos**, and their semantics are governed by **homotopy-coherent descent**.

Conceptual Interpretation

Imagine trying to classify all possible shapes of a molecule. A sheaf would record the set of shapes in each context. A stack records not just the shapes, but the **symmetries**, **deformations**, and **equivalences** between them—capturing the full moduli space.

Or picture a social network. A sheaf might record who is in each group. A stack records **how groups relate**, **who can move between them**, and **how identities transform**—a richer, more dynamic structure. But the mathematical version is more precise: a stack tracks not just membership but all the automorphisms and higher symmetries, encoding the full moduli space of configurations.

In a musical analogy, a sheaf records the notes played by each instrument. A stack records **how those notes harmonize**, **how motifs transform**, and **how themes recur**—the full orchestration.

The Problem of Gluing in Higher Dimensions

In classical sheaf theory, the sheaf condition guarantees that local data defined on an open cover can be glued uniquely into a global section. In the ∞-categorical setting, however, uniqueness is too rigid: overlaps and higher overlaps carry nontrivial homotopy data, and gluing must be allowed **up to coherent homotopy**. This is the fundamental problem that higher sheaf conditions and stacks solve.

The practical effect is that descent becomes a statement about homotopy limits over refined indexing objects (hypercovers), and about mapping spaces of gluings that are contractible rather than single-valued.

Intuition: Gluing with Flexibility and Memory

- In topology, a sheaf assigns local measurements to open sets and ensures they reconcile on overlaps. In the higher setting, reconciliation is structured: the seams themselves can carry twist and curvature that must be recorded.

- In geometry, a stack is a moduli space that remembers how objects are glued and how many automorphisms relate them; it records not only existence but the space of equivalences between gluings.

- In physics, stacks model fields with gauge symmetry: the ``value'' over a region is a space of configurations, and transitions between configurations are governed by higher symmetries.

Hypercovers and Descent (practical test)

Hypercovers are simplicial refinements of covers that encode refinements, overlaps of overlaps, and so on. A presheaf F satisfies **hyperdescent** if for every hypercover $U_\bullet \to U$ the canonical map

$$F(U) \to \lim_{[n] \in \Delta} F(U_n)$$

is an equivalence. This criterion is the strongest practical test for higher sheaf conditions: it lets one replace large indexing diagrams by smaller cofinal ones without changing the colimit, and it reduces many computations to manageable pieces.

Stack Semantics and Moduli

Stacks are used to define **moduli spaces** in derived algebraic geometry:

- The stack of vector bundles on a scheme.
- The stack of perfect complexes.
- The stack of derived schemes or derived stacks.

These stacks are **functors of points**: they assign to each test object the ∞-groupoid of objects over it, satisfying descent.

Stack semantics allows one to **reason internally** about moduli problems, using the logic of ∞-topoi and the machinery of descent.

Stack Semantics: Logic in a Landscape of Symmetries

Stacks are not just geometric—they are **logical environments**. In the internal language of an ∞-topos, stacks represent **types with nontrivial identity structure**. They allow one to reason about objects **up to equivalence**, and to track **how equivalences themselves vary**.

This is the semantic foundation of **homotopy type theory**, **derived geometry**, and **cohesive logic**. Stack semantics replaces rigid truth with **structured coherence**, and replaces equality with **equivalence plus memory**.

Higher Sheaves vs. Classical Sheaves

| Feature | Classical Sheaf | Higher Sheaf |

|-------|----------------|-----------------|

| Values | Sets | Spaces (∞-groupoids) |

| Gluing | Equalizers | Homotopy limits |

| Covers | Čech | Hypercovers |

| Automorphisms | Ignored | Tracked |

| Descent | Strict | Homotopy-coherent |

Higher sheaves generalize classical sheaves by allowing **flexible gluing**, **higher symmetries**, and **derived structure**.

Examples That Ground the Theory

- **Moduli Stack of Vector Bundles**: Assigns to each space the ∞-groupoid of vector bundles on it, including automorphisms and isomorphisms.

- **Classifying Stack BG** : For a group G , the stack BG assigns to each space the ∞-groupoid of principal G -bundles.

- **Derived Stacks**: In derived algebraic geometry, stacks encode derived schemes, where gluing involves not just functions but **chain complexes and homotopies**.

- **Mapping Stacks**: Given stacks X and Y , the mapping stack Map(X, Y) encodes the ∞-groupoid of maps from X to Y , including higher homotopies.

Applications and Significance

- **Derived Algebraic Geometry**: Stacks model derived moduli spaces and deformation problems.
- **Homotopy Type Theory**: Higher sheaves and stacks encode types with identity and equivalence structure.
- **Cohomology**: Stack semantics supports twisted and nonabelian cohomology.
- **Topological Field Theory**: Stacks encode fields, symmetries, and gauge transformations.
- **Higher Geometry**: Stacks are the basic objects of ∞-geometric theories.

Higher sheaf conditions and stack semantics are the **language of moduli and symmetry** in ∞-category theory. They allow one to classify, glue, and reason about complex geometric and logical structures.

Summary

- Higher sheaves satisfy descent via homotopy limits over hypercovers.
- Stacks are presheaves of ∞-groupoids satisfying higher sheaf conditions.
- They encode moduli problems, automorphisms, and derived structure.
- Stack semantics supports internal reasoning, cohomology, and geometry.
- This layer generalizes sheaf theory to the full homotopy-coherent setting.

Layer 21: Shape Theory and Spatial Realization

Extracting the Homotopy Type of ∞-Topoi and Reconnecting with Geometry

---

Purpose and Position in the Tower

This layer introduces **shape theory** and **spatial realization**—the tools that allow one to **extract the underlying homotopy type** of an ∞-topos and relate it back to classical spaces. Shape theory is the **bridge between abstract ∞-categorical logic and tangible homotopical geometry**. It tells us what a topos "looks like" from the perspective of homotopy theory.

Spatial realization, in turn, allows one to **reconstruct a space from its ∞-topos of sheaves**, completing the circle between geometry and logic. This layer builds on the internal logic of ∞-topoi (Layers 15–17), and prepares the ground for comparisons between ∞-topoi and classical homotopy types, especially in étale and étale-motivic contexts.

Classical Background

In classical topology, the **shape** of a topological space refers to its **homotopy type**—the data preserved under continuous deformation. For complicated or non-Hausdorff spaces, shape theory provides a way to **approximate** the space by simpler ones, capturing its essential homotopical features.

In topos theory, the **shape** of a topos is a **pro-object** in spaces that encodes its global homotopy type. This idea is generalized in ∞-category theory to define the **shape of an ∞-topos**.

∞-Categorical Definition of Shape

Let X be an ∞-topos. Its **shape**, denoted Sh(X) , is a **pro-object in spaces** (i.e., a functor from a cofiltered category to S) that encodes the **homotopy type** of X .

Formally, the shape is defined as the **left adjoint** to the constant sheaf functor. For an ∞-topos X , the shape is the pro-space:

$$Sh(X) := \text{colim}_{U \in X} U,$$

where the colimit is taken in the ∞-category of pro-spaces. This captures the **homotopy type** underlying X .

The shape functor:

$$\text{Shape: Topoi}_\infty \rightarrow \text{Pro}(S)$$

is **left adjoint** to the **constant sheaf functor** S → Topoi_∞ , and reflects the **homotopy type** of the topos.

Conceptual Interpretation

Imagine an ∞-topos as a vast, abstract landscape of homotopy-coherent data. The shape functor captures the essential homotopy type—like seeing the fundamental topological structure while abstracting away the specific sheaf-theoretic details. It's a pro-object in spaces that remembers the "coarse geometry" but forgets the fine structure.

Or think of a novel written in a complex language. The shape is the **plot structure**—the skeleton of meaning that remains when the linguistic flourishes are stripped away. More precisely: the shape functor is left adjoint to the constant sheaf functor, extracting the "pro-homotopy type" that underlies the ∞-topos while forgetting the sheaf-theoretic refinements.

In a musical analogy, the shape of an ∞-topos is the **harmonic progression** underlying a symphony: it abstracts away orchestration and dynamics, revealing the structural flow.

Spatial Realization

Spatial realization is the **inverse process**: given a pro-space (or a homotopy type), one constructs an ∞-topos whose shape is that space. This is often done via **localization** or **étale topoi**, and is central to the

theory of **étale homotopy types**, **motivic homotopy theory**, and **topological realization of algebraic geometry**.

For example, the ∞-topos of sheaves on a scheme X with the étale topology has a shape that reflects the **étale homotopy type** of X . Spatial realization allows one to **recover this type** from the topos.

Applications and Significance

- **Étale Homotopy Theory**: The shape of the étale ∞-topos of a scheme recovers its étale homotopy type.
- **Motivic Homotopy Theory**: Shape theory connects ∞-topoi to motivic spaces and spectra.
- **Topological Realization**: Spatial realization allows one to interpret ∞-topoi as spaces.
- **Cohomology**: Shape theory informs the computation of cohomological invariants.
- **Logic and Semantics**: The shape reflects the global truth structure of an ∞-topos.

Shape theory and spatial realization are the **geometric soul** of Higher Topos Theory. They allow one to **see the space behind the logic**, and to **reconstruct logic from space**.

Summary

- The shape of an ∞-topos is a pro-space encoding its homotopy type.
- It is defined via a left adjoint to the constant sheaf functor.
- Spatial realization reconstructs an ∞-topos from a homotopy type.
- These constructions connect ∞-topoi to classical geometry and topology.
- Shape theory is essential for étale homotopy, motivic theory, and cohomological analysis.

Layer 22: Classifying ∞-Topoi and Universal Properties

The Geometry of Theories and the Moduli of Homotopy-Coherent Semantics

---

Purpose and Position in the Tower

This layer reveals how ∞-**topoi serve as semantic universes** for theories. A **classifying ∞-topos** is not merely a space where models live—it is the **universal semantic environment** that *generates* all models of a given theory via pullback. It encodes the theory's **generic model**, and every other model in any ∞-topos arises as a **geometric morphism** from this universal source.

This is extends the machinery built in previous layers: modalities (Layer 17), geometric morphisms (Layer 16), internal logic (Layer 13), and descent (Layer 14). Here, logic becomes geometry, and theories become spaces.

Formal Definition

Let T be a **geometric theory** expressible in the internal language of ∞-topoi. Then the **classifying ∞-topos** X_T is an ∞-topos equipped with a **generic model** $M_T \in X_T$, such that:

> For any ∞-topos Y , models of T in Y correspond to **geometric morphisms**

> > f: $Y \to X_T$ >

> with $f^*(M_T)$ being the model in Y .

This universal property means that X_T **represents** the functor

$$\text{Mod}_T(\text{-}): \text{Topoi}^{\text{op}} \to \text{Spaces}$$

which assigns to each ∞-topos the ∞-groupoid of models of T in that topos.

Building Intuition: What Is a Classifying ∞-Topos Really?

- **In algebra**, the free group on a set is the most general group generated by that set. Every other group with those generators is a quotient.

→ The classifying ∞-topos is the **free semantic environment** for a theory. Every other model is a **semantic quotient**.

- **In geometry**, a moduli space classifies geometric objects up to equivalence.

→ The classifying ∞-topos is the **moduli space of models** of a theory.

- **In logic**, a theory has models in various structures.

→ The classifying ∞-topos is the **universal structure** where the theory is most generically realized.

- **A lens factory**: The classifying ∞-topos doesn't just contain lenses—it **produces** them. Each geometric morphism to another ∞-topos is a lens through which the theory is viewed in that context. More precisely: pulling back along a geometric morphism instantiates the generic model in a specific ∞-topos.

- **A musical motif**: The classifying ∞-topos is the **motif**—a minimal, generative structure. Each model is a **variation** on that motif, shaped by the ambient ∞-topos. Though unlike music where variations are creative choices, here the variations are determined by geometric morphisms— pullback functors that preserve the theory's logical structure.

- **A language's grammar**: It's not a dictionary—it's the **syntax engine**. Each speaker (∞-topos) uses it to generate sentences (models), but the grammar remains invariant.

- **A genetic code**: The classifying ∞-topos is the **genotype** of a theory. Each model is a **phenotype**, expressed differently depending on the surrounding ∞-topos.

- **A universal sketchpad**: It's the **canvas** where the theory is drawn in its purest form. Every other drawing (model) is a projection onto a different surface (∞-topos).

Examples Across Domains

Let's ground this in concrete examples:

- **Spaces S **: Classifies ∞-groupoids. Every ∞-groupoid lives in S , and geometric morphisms to S encode their structure.

- **Sheaves on a Site**: Classify local models of geometric theories. For example, sheaves on the étale site of a scheme classify étale-local data.

- **Spectra Sp **: Classify stable homotopy types. The ∞-topos of spectra is the semantic home of stable phenomena.

- **Moduli Stacks**: Classify objects like vector bundles, coherent sheaves, or derived schemes. These are classifying ∞-topoi for geometric theories.

- **Synthetic Differential ∞-Topoi**: Classify smooth structures, infinitesimal neighborhoods, and differential modalities.

Applications and Significance

- **Homotopy Type Theory**: Classifying ∞-topoi model type theories and their semantics. Each type theory has a semantic universe.

- **Derived Algebraic Geometry**: Moduli stacks are classifying ∞-topoi for derived structures.

- **Cohomology and Descent**: Classifying ∞-topoi encode cohomological invariants and descent data.

- **Modal Logic**: Modalities correspond to subtopoi of classifying ∞-topoi, encoding logical operators geometrically.

- **Higher Algebra**: Classifying ∞-topoi represent algebraic theories (e.g., monoids, rings, categories) in ∞-categorical terms.

Summary

- A **classifying ∞-topos** X_T represents a theory T via a universal property.
- It contains a **generic model**, and every other model arises via pullback.
- It is a **moduli space** of models, encoding the semantics of T .
- It unifies logic, geometry, and homotopy in a single ∞-categorical framework.
- Classifying ∞-topoi are foundational for semantics, moduli, and universal constructions.

Layer 23: Structured ∞-Categorical Universes and Logical Topoi

---

Semantic Geometry in Motion

An ∞-topos is not a monolith. It can be sliced, stratified, indexed, and transported. This layer concerns the geometry of ∞-topoi not as isolated semantic spaces, but as **structured families**—varying over a base, interacting through morphisms, and forming higher sheaves of universes. These constructions allow logic, geometry, and homotopy to **flow**, **branch**, and **recombine** across ∞-categorical landscapes.

Let B be an ∞-category. A **family of ∞-topoi over B ** is a functor

$$X: B^{op} \to Topoi_\infty$$

assigning to each object $b \in B$ an ∞-topos X_b , and to each morphism f: b' → b , a geometric morphism $X_b \to X_{b'}$. This is not a mere indexing—it is a **semantic fibration**, a way for logic to vary over space, time, or structure.

Fibered ∞-Topoi: Transporting Logic Across a Base

A **fibered ∞-topos** is a Cartesian fibration

$$p: E \to B$$

where each fiber E_b is an ∞-topos, and the transition functors are geometric morphisms. This structure encodes **semantic transport**: how truth, type, and structure move across a base category.

This is the ∞-categorical analog of a vector bundle, but instead of vectors, each fiber carries a **universe of homotopy types**, and the transition functions are **logical translations**. The total space E is a **semantic manifold**, and the base B may represent time, context, or parameter space.

Stacks of ∞-Topoi: Gluing Universes with Descent

A **stack of ∞-topoi** is a presheaf

$$X: C^{op} \to Topoi_\infty$$

satisfying descent with respect to a Grothendieck topology on C . This means that for every covering sieve R ⊆ Hom(-, U) , the ∞-topos X(U) can be reconstructed from the values X(V) for V → U in R , via a homotopy limit in the ∞-category of ∞-topoi.

This is a **sheaf of semantic worlds**, where each open set carries an ∞-topos, and the gluing conditions ensure that logic is **coherently distributed**. It is the ∞-categorical analog of a sheaf of rings, but with entire logical universes as values.

Logical Topoi: Internal Languages and Modal Semantics

A **logical ∞-topos** is an ∞-topos equipped with structure sufficient to interpret a type theory. This includes:

- **Universe objects**: classifying families of types.
- **Modalities**: reflective subtopoi encoding logical operators.
- **Dependent types**: modeled by fibrations.
- **Identity types**: interpreted via path spaces.
- **Higher inductive types**: constructed via colimits and truncations.

Each logical ∞-topos is a **semantic engine**, capable of interpreting homotopy type theory, modal logic, and synthetic differential geometry. The logic is not imposed—it **emerges** from the geometry of the topos.

Examples

- **Étale ∞-topoi over a scheme**: form a fibered ∞-topos over the category of schemes, encoding local geometric variation.

- **Cohesive ∞-topoi**: fibered over a base encoding spatial or temporal variation, with modalities for shape, flatness, and infinitesimality.

- **Stacks of ∞-topoi in derived geometry**: encode families of derived stacks varying over a base moduli space.

- **Logical fibrations in type theory**: model context-dependent semantics, where each fiber is an ∞-topos interpreting a fragment of a theory.

Analogical Interweaving

- A fibered ∞-topos is like a **library system** spread across cities: each branch (fiber) has its own collection (types), but interlibrary loans (geometric morphisms) allow knowledge to flow. More precisely: the base parametrizes contexts, each fiber is a full ∞-topos, and the Cartesian structure ensures coherent variation.

- A stack of ∞-topoi resembles a **distributed system**: each node runs its own logic, but the system ensures coherence across the network. Though unlike computer systems, mathematical coherence is perfect—the descent conditions guarantee exact gluing.

- A logical ∞-topos is a **language with dialects**: the core grammar is universal, but each context (fiber) adapts it to local semantics.

- The total space of a fibered ∞-topos is a **semantic landscape**, where each point carries a universe, and paths encode transformations of meaning.

- Modalities in logical ∞-topoi are **filters of perception**: they determine what kinds of truth are visible in each subtopos.

Summary

Structured ∞-categorical universes arise as fibered, indexed, or stacked ∞-topoi. These constructions model variation, stratification, and internal semantics across ∞-categories. Logical ∞-topoi support type-theoretic and modal semantics. This layer generalizes sheaf theory to families of semantic environments, foundational for derived geometry and homotopy type theory.

Layer 24: The ∞-Category of ∞-Topoi

Universes of Universes and the Geometry of Semantic Flow

∞-Topoi as Objects in a Higher Geometry

An ∞-topos is a universe of homotopy types—a setting where spaces, logic, and structure coexist and interact. But these universes are not isolated. They can be compared, connected, and transformed. The ∞-category

$$\mathrm{Topoi}_\infty$$

is the mathematical structure that organizes all ∞-topoi and the ways they relate. Each object is an ∞-topos; each morphism is a **geometric morphism**, a pair of functors that translate logic and structure between universes.

This is not a mere catalog. It is a **semantic geometry**: a space whose points are entire logical worlds, and whose paths are structure-preserving translations between them.

Geometric Morphisms: Translating Universes

A geometric morphism

$$f: X \to Y$$

consists of two functors:

- $f^*: Y \to X$: the **inverse image**, which pulls back objects from Y into X ,
- $f_*: X \to Y$: the **direct image**, which pushes forward objects from X into Y .

These functors are **adjoint**: they satisfy a balancing condition that ensures compatibility. Crucially, f^* must preserve **finite limits**—this guarantees that logical structure (like conjunctions, implications, and identity types) remains intact when translated.

This is not just a technical requirement. It ensures that the **meaning** of constructions survives the journey from one universe to another.

Limits and Colimits: Gluing and Intersecting Universes

In ordinary category theory:

- A **limit** is a universal object satisfying a system of constraints.
- A **colimit** is a universal object obtained by gluing together a diagram.

In Topoi_∞ , these notions take on semantic meaning:

- A **limit** of ∞-topoi corresponds to a **shared semantic core**: the intersection of multiple universes, retaining only the structure they all agree on.
- A **colimit** corresponds to a **semantic amalgam**: a new universe that contains and coherently merges the logic of several others.

These constructions are essential in descent theory, stack semantics, and the construction of moduli spaces of theories.

Semantic Flow: The Geometry of Interpretation

A geometric morphism is not just a map—it is a **channel of interpretation**. It allows one ∞-topos to be understood inside another. The inverse image functor f^* translates types, propositions, and

constructions from Y into X . The direct image functor f∗ sends results, truth values, and cohomological data back.

This bidirectional flow is the **lifeblood of semantic geometry**. It enables:

- **Transport of logic** across contexts,
- **Interpretation of theories** in new settings,
- **Gluing of local semantics** into global structure.

Stratification: Layers of Logical Depth

Not all ∞-topoi are equal in expressive power. The ∞-category Topoi_∞ is **stratified** by logical and homotopical complexity:

- **n-topoi**: ∞-topoi where all objects are n -truncated. These are universes where types have no homotopy above level n . For example, a 0-topos is a classical topos; a 1-topos supports groupoids; a 2-topos supports 2-groupoids, and so on.

- **Cohesive ∞-topoi**: equipped with modalities that distinguish between discrete, continuous, and infinitesimal structure. These support synthetic differential geometry and spatial reasoning.

- **Logical ∞-topoi**: support internal type theories with universe objects, univalence, and higher inductive types. These are semantic engines for homotopy type theory.

This stratification is not arbitrary—it reflects the **dimensional architecture** of logic itself.

Moduli of ∞-Topoi: Universes as Parameters

Just as one can study moduli spaces of curves, bundles, or varieties, one can study **moduli spaces of ∞-topoi**. These are higher stacks whose points are ∞-topoi and whose paths are geometric morphisms.

Examples:

- The moduli stack of sheaf ∞-topoi over a site.
- The moduli of cohesive ∞-topoi over a spatial base.
- The moduli of logical ∞-topoi interpreting a fixed type theory.

These moduli encode **variation of semantics** across parameters—how logic changes over time, space, or structure.

Examples

- The ∞-topos of spaces S is **terminal**: every ∞-topos maps to it, reflecting its underlying homotopy type.
- The ∞-topos of spectra Sp is **initial** among stable ∞-topoi: it is the minimal universe for stable phenomena.
- Sheaf ∞-topoi on sites form a full subcategory of Topoi_∞ , closed under limits and colimits.
- The ∞-category of ∞-topoi over a base B forms a **fibered ∞-category**, encoding variation of semantics across a parameter space.

Summary

The ∞-category of ∞-topoi is the semantic geometry of universes. It organizes logical worlds into a higher structure, where morphisms are translations, limits are intersections, colimits are amalgamations, and stratification reflects logical depth. It is the infrastructure for understanding how logic, geometry, and homotopy interact across contexts.

Layer 25: Cohesive ∞-Topoi

Encoding Spatial Structure via Modalities and Adjoint Strings

Purpose and Position in the Tower

This layer introduces **cohesive ∞-topoi**—∞-topoi equipped with an adjoint quadruple of functors that encode how **discrete**, **continuous**, and **infinitesimal** structures relate. Cohesive ∞-topoi provide a synthetic framework for differential geometry, topology, and smooth structure, where spatial concepts emerge from categorical relationships rather than being imposed externally.

Building on modalities (Layer 16) and geometric morphisms (Layer 15), cohesive ∞-topoi formalize the intuition that spaces have both **discrete points** and **continuous structure**, with systematic ways to pass between these perspectives.

Classical Background

In classical mathematics, we distinguish between:

- **Discrete sets**: Collections of isolated points
- **Topological spaces**: Sets with continuous structure
- **Smooth manifolds**: Spaces with differential structure

These are typically treated as separate categories with forgetful functors between them. Cohesive ∞-topoi unify this picture by encoding all three levels—and their relationships—within a single ∞-topos.

Formal Definition

A **cohesive ∞-topos** is an ∞-topos H equipped with an adjoint quadruple of functors to the ∞-topos of spaces S :

$$\Pi \dashv \text{Disc} \dashv \Gamma \dashv \text{Codisc}: H \leftrightarrows S$$

where:

- Π : **Shape**—forgets cohesive structure, returning underlying homotopy type
- Disc : **Discrete**—regards a space as discrete within H
- Γ : **Global sections**—underlying points functor
- Codisc : **Codiscrete**—regards a space as codiscrete (indiscrete) within H

These functors satisfy:

1. Disc and Codisc are fully faithful
2. Π preserves finite products
3. Disc preserves finite products

Conceptual Interpretation

Think of a cohesive ∞-topos as encoding **layers of resolution**:

- **Discrete layer** (Disc): Individual points, like pixels on a screen
- **Continuous layer** (Π): Smooth curves and surfaces, like vector graphics
- **Codiscrete layer** (Codisc): Blurred, indistinguishable structure

The adjoint quadruple provides systematic ways to move between these layers, extracting discrete data from continuous objects or building continuous structure from discrete pieces.

Imagine a city map:

- Π gives the **connectivity graph**—which neighborhoods connect to which (the fundamental ∞-groupoid)
- Disc views individual **buildings as isolated points** (discrete objects have only identity morphisms)
- Γ extracts the **set of all locations** (global sections functor)
- Codisc sees the city as **one undifferentiated blob** (codiscrete objects have unique morphisms between any two points)

The mathematical version is exact: these four functors form an adjoint chain relating discrete and cohesive structures.

Axiomatic Consequences

From the adjoint quadruple, we derive modalities:

- **Shape modality** $\natural = \text{Disc} \circ \Pi$: Discretizes the shape
- **Flat modality** $\flat = \text{Codisc} \circ \Gamma$: Forgets cohesion
- **Sharp modality** $\sharp = \text{Codisc} \circ \text{Disc}$: Codiscretizes

These modalities satisfy:

$$\natural \dashv \flat \dashv \sharp$$

This adjoint triple of modalities encodes how cohesive structure can be **forgotten**, **flattened**, or **trivialized**.

Pieces-Have-Points Condition

A cohesive ∞-topos satisfies **pieces-have-points** if the canonical morphism

$$\Pi(X) \to \Gamma(X)$$

is an epimorphism for all $X \in H$. This ensures that every connected component contains at least one point—ruling out "ghost" components with shape but no substance.

This condition distinguishes **geometric** cohesion (where pieces have points) from more exotic forms where continuous structure can exist without discrete support.

Differential Cohesion

A **differentially cohesive ∞-topos** has an additional adjoint quadruple encoding **infinitesimal** structure:

$$\Pi_{\text{inf}} \dashv \text{Disc}_{\text{inf}} \dashv \Gamma_{\text{inf}} \dashv \text{Codisc}_{\text{inf}}$$

This gives rise to:

- **Infinitesimal shape** $\text{Im} = \text{Disc}_{\text{inf}} \circ \Pi_{\text{inf}}$
- **Reduction modality** $\text{Re} = \text{Codisc}_{\text{inf}} \circ \Gamma_{\text{inf}}$
- **Infinitesimal modality** $\backslash\& = \text{Im} \circ \text{Re}$

These encode **jets**, **tangent spaces**, and **formal neighborhoods**—the apparatus of differential geometry expressed categorically.

Examples

- **Smooth ∞-groupoids**: The ∞-topos of sheaves on the site of smooth manifolds with open covers
- **Synthetic differential ∞-groupoids**: Sheaves on the site of infinitesimally thickened smooth manifolds
- **Euclidean-topological ∞-groupoids**: Sheaves on the site of Euclidean spaces with continuous maps
- **Complex-analytic ∞-groupoids**: Sheaves on complex manifolds with holomorphic maps

Each provides a different model of cohesion, encoding different notions of spatial and smooth structure.

Applications and Significance

- **Synthetic Differential Geometry**: Cohesive ∞-topoi provide models where infinitesimals exist as actual objects
- **Higher Gauge Theory**: Gauge fields and connections are naturally formulated in differential cohesion

- **Smooth Homotopy Theory**: The shape modality extracts homotopy types while respecting smooth structure
- **Formal Moduli Problems**: Differential cohesion encodes deformation theory synthetically

---

Summary

- Cohesive ∞-topoi encode spatial structure via an adjoint quadruple to spaces
- The adjunctions give rise to shape, flat, and sharp modalities
- Pieces-have-points ensures geometric behavior
- Differential cohesion adds infinitesimal structure via another adjoint quadruple
- This framework provides synthetic approaches to topology, differential geometry, and smooth homotopy theory

---

Layer 26: (∞,2)-Categories

The First Step Beyond ∞-Categories: Non-Invertible 2-Morphisms

---

Purpose and Position in the Tower

This layer introduces **(∞,2)-categories**—∞-categories where 2-morphisms need not be invertible. This is the first step beyond ordinary ∞-categories (which are (∞,1)-categories), adding a new dimension of non-invertible structure. While ∞-categories track objects, morphisms, and their homotopies, (∞,2)-categories also track **transformations between morphisms** that cannot be undone.

This layer builds on the foundation of ∞-categories developed in Layers 1-24, preparing for (∞,2)-topoi and eventually higher (∞,n)-structures. The jump from (∞,1) to (∞,2) is conceptually the most significant—it introduces genuinely new phenomena that cannot be reduced to homotopy.

Classical Background

In ordinary category theory:

- A **category** has objects and morphisms
- A **2-category** adds 2-morphisms (transformations between morphisms)
- A **strict 2-category** requires associativity and unit laws to hold exactly

- A **weak 2-category** (bicategory) allows these laws to hold up to isomorphism

The passage to $(\infty,2)$-categories generalizes weak 2-categories by:

- Allowing all higher morphisms (3-morphisms, 4-morphisms, etc.)
- Requiring these higher morphisms to be invertible above dimension 2
- Encoding coherence via simplicial or complete Segal space models

---

Formal Definition

An **$(\infty,2)$-category** is a simplicial space $X_{\bullet,\bullet}$ satisfying:

1. **Segal conditions**: For each n , the map

$$X_{n,\bullet} \to X_{1,\bullet} \times_{X_{0,\bullet}} \cdots \times_{X_{0,\bullet}} X_{1,\bullet}$$

is an equivalence of simplicial spaces (n factors on the right)

1. **Completeness**: The map $X_{0,\bullet} \to X_{1,\bullet}^{\wedge}\{\simeq\}$ (to invertible morphisms) is an equivalence

1. **2-Segal conditions**: The vertical direction also satisfies Segal conditions

Alternatively, an $(\infty,2)$-category is a **category object** in the ∞-category of ∞-categories, or equivalently, a functor from Δ^{op} to ∞-Cat satisfying descent.

Conceptual Interpretation

Think of an $(\infty,2)$-category as having three levels of structure:

- **Objects** (0-cells): The entities being studied
- **1-morphisms** (1-cells): Transformations between objects
- **2-morphisms** (2-cells): Modifications of transformations
- **Higher morphisms** (n-cells for $n > 2$): All invertible, encoding coherence

The key innovation is that 2-morphisms are **directed**—they have a source and target 1-morphism, but may not have an inverse. This allows us to express:

- **Inequalities** between morphisms (one is "better" than another)
- **Reductions** that cannot be reversed
- **Information loss** in transformations

Imagine a revision control system:

- **Files** are objects
- **Commits** are 1-morphisms
- **Merge strategies** are 2-morphisms
- A merge might lose information (non-invertible 2-morphism)
- Higher structure tracks consistency of the revision history

---

The Homotopy Hypothesis for (∞,2)-Categories

Just as ∞-groupoids model homotopy types, (∞,2)-categories have their own spatial interpretation:

- **(∞,0)-categories** (∞-groupoids) model **homotopy types**
- **(∞,1)-categories** model **directed homotopy types**
- **(∞,2)-categories** model **doubly directed homotopy types**

An (∞,2)-category can be geometrically realized as a space with:

- Points (objects)
- Directed paths (1-morphisms)
- Directed surfaces between paths (2-morphisms)
- Higher homotopies (all invertible above dimension 2)

Examples

- ∞-**Cat**: The (∞,2)-category of ∞-categories, functors, and natural transformations
- **Bimod(R)**: The (∞,2)-category of rings, bimodules, and bimodule maps
- **Span(C)**: The (∞,2)-category of spans in an ∞-category C
- **Corr(C)**: The (∞,2)-category of correspondences
- **2-Vect**: The (∞,2)-category of 2-vector spaces (categorified linear algebra)

Each exhibits genuine 2-categorical phenomena not visible at the (∞,1)-level.

Gray Tensor Product

The **Gray tensor product** $\otimes_{\text{Gray}}$ provides a symmetric monoidal structure on (∞,2)-Cat. Unlike the Cartesian product, the Gray product encodes the non-trivial braiding of 2-morphisms, capturing how transformations in different directions can interfere.

This tensor product is essential for:

- Defining **monoidal (∞,2)-categories**
- Constructing **(∞,2)-categorical traces**
- Formulating **3-dimensional TQFTs**

Adjunctions in (∞,2)-Categories

An adjunction in an (∞,2)-category consists of:

- 1-morphisms f: C → D and g: D → C
- 2-morphisms η: id_C ⇒ g ∘ f (unit)
- 2-morphisms ε: f ∘ g ⇒ id_D (counit)
- Invertible modifications satisfying triangle identities

This enriches the notion of adjunction with 2-categorical structure, allowing for:

- **Pseudo-adjunctions** (up to isomorphism)
- **Lax adjunctions** (up to transformation)
- **Mates** (correspondence between 2-cells)

Applications and Significance

- **Categorified Algebra**: (∞,2)-categories provide the framework for categorifying algebraic structures
- **2-Dimensional TQFTs**: Extended 2D TQFTs are functors from 2Cob to (∞,2)-categories
- **Higher Stacks**: 2-stacks are fibered (∞,2)-categories, tracking higher automorphisms
- **Derived Morita Theory**: Equivalences of derived categories form an (∞,2)-category
- **String Topology**: Operations on loop spaces organize into (∞,2)-categorical structure

---

Summary

- (∞,2)-categories add non-invertible 2-morphisms to ∞-categories
- They can be modeled as simplicial spaces satisfying 2-Segal conditions
- The Gray tensor product captures non-trivial braiding of 2-morphisms
- Examples include ∞-Cat, bimodules, spans, and correspondences
- They are essential for categorified algebra, 2D TQFTs, and higher stack theory

---

Layer 27: (∞,2)-Topoi and Higher Categorical Logic

Topos Theory in Dimension Two

Purpose and Position in the Tower

This layer extends the theory of ∞-topoi (Layer 11) to the (∞,2)-categorical setting, introducing **(∞,2)-topoi**—(∞,2)-categories that behave like topoi. These are universes where we track not just objects and morphisms up to homotopy, but also **non-invertible 2-morphisms** that encode directed transformations between morphisms.

Building on (∞,2)-categories (Layer 25) and the theory of ∞-topoi (Layers 11-24), this layer explores what happens when topos-theoretic concepts are enhanced with genuine 2-categorical structure. The result is a framework for **2-categorical logic**, **directed type theory**, and **higher stack theory**.

Classical Background

The progression of topos theory through dimensions:

- **1-topoi**: Grothendieck topoi, categories of sheaves
- **2-topoi**: 2-categories of stacks, encode stacky phenomena
- **(∞,1)-topoi**: ∞-topoi, homotopy-coherent sheaf theory
- **(∞,2)-topoi**: Combine stacky and homotopical phenomena

A 2-topos traditionally satisfies 2-categorical versions of the Giraud axioms, with exactness conditions at both the 1-morphism and 2-morphism levels.

Formal Definition

An **(∞,2)-topos** is an (∞,2)-category X satisfying:

1. **2-Presentability**: X is accessible and admits all small (∞,2)-colimits
2. **2-Exactness**: (∞,2)-colimits are universal—preserved by base change
3. **2-Descent**: Satisfies descent for (∞,2)-categorical hypercovers
4. **Local (∞,1)-topos structure**: Each hom-∞-category X(A,B) forms an ∞-topos

Additionally, X has:

- **2-Grothendieck construction**: Fibrations classify functors to $(\infty,2)$-Cat
- **2-Yoneda embedding**: Fully faithful at the $(\infty,2)$-level
- **2-Kan extensions**: Exist and satisfy 2-categorical universal properties

---

The Role of Directed 2-Morphisms

The non-invertibility of 2-morphisms introduces new phenomena:

- **Directed Logic**: Implications that cannot be reversed
- **Computational Reductions**: Optimizations that lose intermediate data
- **Asymmetric Coherence**: Transformations with preferred directions

For example, in an $(\infty,2)$-topos of computational processes:

- Objects are **data types**
- 1-morphisms are **programs**
- 2-morphisms are **optimizations** (often non-invertible)
- A compiler optimization that eliminates dead code is a non-invertible 2-morphism

---

2-Sheaf Theory

A **2-sheaf** on a 2-site is a 2-functor satisfying descent for 2-categorical covers. The sheaf condition now involves:

- **1-descent**: Local sections glue to global sections
- **2-descent**: Local transformations glue to global transformations
- **Coherence**: Higher coherence data for the gluing

This enriched descent captures phenomena like:

- **Stacky quotients**: Objects with automorphisms
- **Higher gauge theory**: Connections with gauge transformations
- **Categorified cohomology**: Cohomology with coefficients in 2-groups

---

Internal Logic of $(\infty,2)$-Topoi

The internal logic extends homotopy type theory with:

- **Directed identity types**: Non-symmetric path types
- **2-truncation modality**: Projects to underlying $(\infty,1)$-structure
- **Computational modalities**: Track reduction and evaluation
- **Directed univalence**: Equivalences that respect directionality

This yields a **directed type theory** where:

- Types are objects
- Terms are 1-morphisms
- Rewrites are 2-morphisms
- Computation has direction

Stack Semantics in (∞,2)-Topoi

An $(\infty,2)$-topos naturally supports **2-stacks**—sheaves of $(\infty,1)$-categories satisfying 2-descent. These encode:

- **Moduli 2-stacks**: Moduli of objects with automorphisms and transformations
- **Higher gerbes**: 2-gerbes classify 2-group cohomology
- **Categorified vector bundles**: 2-vector bundles with connection data

The passage from stacks to 2-stacks adds a layer of categorical structure to moduli problems.

Examples of (∞,2)-Topoi

- **2-Sheaves on a 2-site**: The $(\infty,2)$-topos of 2-sheaves with values in $(\infty,1)$-Cat
- **Directed spaces**: Sheaves on directed topological spaces
- **Parameterized spectra**: Stable $(\infty,2)$-topos of parameterized spectra
- **Derived 2-stacks**: $(\infty,2)$-topos of derived stacks on a derived scheme
- **Computational universes**: $(\infty,2)$-topoi modeling reduction in programming languages

---

Universal Properties and 2-Classifying Objects

An $(\infty,2)$-topos has classifying objects for:

- **$(\infty,1)$-categories**: Object classifying internal ∞-categories
- **2-groups**: Object classifying 2-group torsors
- **Directed fibrations**: Object classifying directed families

These classifying objects organize into an internal **(∞,2)-category of (∞,1)-categories**, providing a universe for 2-categorical logic.

Monoidal Structure and Enrichment

Many (∞,2)-topoi carry additional structure:

- **Monoidal structure**: Tensor product compatible with colimits
- **Enrichment**: Over other (∞,2)-topoi or monoidal (∞,2)-categories
- **Duality**: Compact objects and perfect pairings

This structure is essential for:

- **Categorified algebra**: Modules over algebra objects
- **2-categorical traces**: Hochschild homology in dimension 2
- **Extended field theories**: 3D TQFTs valued in (∞,2)-topoi

---

Applications and Significance

- **Directed Homotopy Theory**: Models of concurrent computation
- **Higher Gauge Theory**: 2-connections and 2-holonomy
- **Categorified Logic**: Proof theory with computational content
- **Derived Symplectic Geometry**: Lagrangian correspondences form an (∞,2)-topos
- **Higher Topos Theory**: Foundation for (∞,n)-topoi

Summary

- (∞,2)-topoi combine ∞-topos theory with genuine 2-categorical structure
- Non-invertible 2-morphisms encode directed transformations and computations
- 2-descent and 2-sheaf theory capture stacky phenomena
- Internal logic extends to directed type theory
- Examples include 2-sheaves, directed spaces, and computational universes
- They provide the foundation for higher topos theory and categorified logic