Foundations: The Representation Stack and the Realization Pipeline

Part I of *A Math→Physics Representation Library*

Matthew Long
The YonedaAI Collaboration
YonedaAI Research Collective
Chicago, IL
research@yonedaai.com · https://yonedaai.com

4 July 2026

Abstract

We initiate a seven-part modular program that formalizes each mathematical domain as a *physical-representation module*. This first part supplies the common substrate on which the remaining six build: a disciplined, domain-indexed formalism for translating mathematical structure into physical representation. We introduce the category Dom of mathematical domains, the fibered assignment $d \mapsto \mathsf{Math}_d$ of structures in each domain, and a target category Phys of physical representation objects (state spaces, observable algebras, process categories, gauge moduli, quantum code spaces, measurement outcomes). The central data structure is the *representation entry* $E = (M, P, \tau, \sigma)$: a mathematical object M, a proposed physical representation P, a translation datum τ, and an *epistemic status label* $\sigma \in \{\mathsf{S}, \mathsf{H}, \mathsf{P}\}$ (standard / heuristic / speculative). Entries organize into a *representation prestack* $\mathrm{Rep}_{\mathrm{phys}} : \mathsf{Dom}^{\mathrm{op}} \to \mathsf{Gpd}$, which becomes a *representation stack* exactly when local entries glue up to coherent gauge equivalence. Structure becomes number through the *realization pipeline* $\mathrm{Obs}_\alpha(M) = \mathrm{Obs}\big(\mathrm{Real}_\alpha(\Phi(M))\big)$, governed by four axioms: Realization, Equivalence, Locality/descent, and Decomposition. Our principal contributions are: (T1) functoriality of the realization pipeline; (T2) a descent criterion characterizing when the representation prestack is a stack; (T3) a *status calculus* making $\{\mathsf{S}, \mathsf{H}, \mathsf{P}\}$ an ordered commutative monoid under which epistemic reliability is monotone non-increasing along chains of translations; and (T4) an obstruction theorem showing that the coarse (gauge-quotiented) representation library fails descent precisely when nontrivial automorphisms are present. We provide a runnable Haskell encoding of the stack and pipeline and an idiomatic Lean sketch of the core types. The framework is deliberately *modular*, not unified: it fixes the notation, axioms, and status discipline that Parts II–VI instantiate for motives/periods, algebraic geometry, algebraic topology, category theory/HoTT, and sheaves/stacks/gauge, closing the loop in the synthesis Part.

Contents

1 Introduction

1.1 Motivation

A recurring experience in mathematical physics is that a physical quantity is not merely *described by* some piece of mathematics but is literally *obtained* from it by a structured act of realization. A scattering amplitude is obtained as a period integral of an algebraic form over a chain; a partition function is obtained as a trace; a topological quantum field theory (TQFT) assigns a vector space to a manifold and a linear map to a cobordism; a logical qubit is obtained by encoding into a larger physical Hilbert space. In each case a mathematical object M is passed through a pipeline

$$\begin{aligned} M \xrightarrow{\Phi} \text{abstract physical information} &\xrightarrow{\mathrm{Real}_\alpha} \text{physical representation} \\ &\xrightarrow{\mathrm{Obs}} \text{measurement data}, \end{aligned}$$

and what we call an "observable" is the endpoint of this pipeline. The guiding slogan of the entire program is compressed as

$$\boxed{\text{observable} \;=\; \mathrm{Real}(\text{abstract structure}).}$$

This paper does *not* claim that all of physics is a functor out of all of mathematics. Such a single universal functor is almost certainly not well-defined. Instead we take the opposite, deliberately modest stance: the translation is *local* and *domain-indexed*, and every translation carries an explicit *epistemic status label* recording whether it is a standard mathematical fact, a strong heuristic, or a speculative ontological proposal. The status label is part of the formalism, not decoration. It is the device by which the framework remains honest about what it has actually established versus what it merely suggests.

1.2 The modular series

This is Part I of a seven-part modular series, *A Math→Physics Representation Library*. We stress the word *modular*: this is not a single monolithic theory but a ladder of standalone modules, each readable and defensible on its own, whose composition is exhibited as an explicit hierarchical building process rather than asserted as an already-unified whole. The ladder is summarized in Table 1.

The ladder is, by design, *circular*. Part I *posits* that representation entries behave like a stack: local data glue not on the nose but up to equivalence. Part VI *proves* what a stack actually is (a fibered category satisfying descent) and shows that $\mathrm{Rep}_{\mathrm{phys}}$ is stack-like exactly when compatible local entries glue up to gauge equivalence. Thus Part VI supplies the rigorous foundation for Part I's opening formalism; the synthesis Part records this as a "closure theorem." We therefore write Part I with explicit forward references, flagging every promissory note that a later module discharges.

1.3 Contributions

The concrete contributions of this paper are:

Part	Topic	Structure introduced / emergent property
I	Foundations (this paper)	$\mathsf{Dom}, \mathsf{Math}_d, \mathsf{Phys}$; representation entries (M, P, τ, σ); prestack/stack; realization pipeline $\mathrm{Obs} \circ \mathrm{Real}_\alpha \circ \Phi$; status system $\mathsf{S}/\mathsf{H}/\mathsf{P}$. *Disciplined translation as data.*
II	Motives, periods, amplitudes	Motives $\mathrm{Mot}(X)$, period data $\Pi = (X, D, \omega, \Gamma)$, motivic amplitude objects $\mathcal{A}^{\mathfrak{m}}$, coproduct Δ, cobracket δ. Real_α *made concrete and numeric.*
III	Algebraic geometry	Schemes, moduli stacks, positive geometries, Hodge structures, Gauss–Manin. *Possibility spaces as geometry.*
IV	Algebraic topology	(Co)homology, characteristic classes, bordism, TQFT as $\mathsf{Bord}_n \to \mathsf{Vect}$. *Functorial realization; conserved global data.*
V	Category theory & HoTT	Categories, functors, monoidal/dagger-compact structure, univalence. *The universal grammar; equivalence becomes identity.*
VI	Sheaves, stacks, gauge, quantum HA	Stacks as pseudofunctors, gauge redundancy $[X/G]$, quantum high-availability codes, Hopf-algebraic renormalization. *Many representatives, one logical content.*
Synth.	Recomposition	Reassembly of I–VI into one conjectural ontology plus limitations. *The ladder closes on Part I.*

Table 1: The modular composition ladder. Each part introduces new structure and adds an emergent property that the composition makes available.

1. A precise definition of the objects Dom, Math_d, Phys and the bracket notation $[\![M]\!]_{\text{phys}}$ for the physical representation of M (Section 2).
2. The four core definitions of the representation-stack formalism: *representation entry*, *representation library*, *representation prestack*, and *representation stack* (Section 3), together with the master diagram (Figure 1).
3. The *realization pipeline* $\mathrm{Obs}_\alpha = \mathrm{Obs} \circ \mathrm{Real}_\alpha \circ \Phi$ and its four governing axioms (Section 4), plus the parallel six-axiom *ontology* presentation and its central conjecture (Section 5).
4. Four results with proofs: functoriality of the pipeline (Theorem 6.1, a proposition), the descent characterization of stackhood (Theorem 6.3), the status calculus (Theorem 6.5), and the coarse-quotient obstruction (Theorem 6.8), collected in Section 6.
5. The core translation library and two cross-cutting tables (operations $\leftrightarrow$ physical interpretation; domains $\leftrightarrow$ physical faculties) that every later module reuses (Section 7).
6. A runnable Haskell encoding and an idiomatic Lean sketch of the stack and pipeline (Section 9).

1.4 Epistemic status system

Throughout, every dictionary entry and every proposed correspondence carries one of three labels, used *verbatim* across all seven parts.

Label	**Meaning**	**Canonical example**
S	standard use in mathematics or mathematical physics	TQFT as a functor $\mathsf{Bord}_n \to \mathsf{Vect}$
H	strong heuristic representation-dictionary entry	motive as "functional essence"
P	speculative philosophical ontology	reality as motivic-categorical realization

Composite labels (S/H, H/P) appear where an entry is standard as pure mathematics but heuristic or speculative in its physical reading; we preserve these composites verbatim. Theorem 6.5 promotes this informal system to a genuine algebraic calculus.

1.5 Outline

Section 2 fixes the categorical framework. Section 3 defines the representation stack. Section 4 defines the realization pipeline and its axioms. Section 5 presents the ontology axioms and the closure conjecture. Section 6 proves the four main theorems. Section 7 reproduces the core translation library. Section 8 sets up Parts II–VI as concrete instantiations. Section 9 presents the Haskell and Lean formalizations. Section 10 discusses limitations, and Section 11 concludes with the four slogans of the program.

2 Mathematical Framework

We work throughout with (possibly higher) categories. For definiteness a reader may take all categories to be ordinary 1-categories on first reading and reinterpret Gpd, Cat as $(2,1)$- and 2-categories where the text says "pseudofunctor" or "up to coherent isomorphism." Part V makes the higher structure precise; here we need only the following ingredients.

2.1 Domains, structures, and physical targets

Definition 2.1 (Domain category)**.** The *domain category* Dom has as objects mathematical domains—algebraic topology, algebraic geometry, category theory, homotopy type theory, sheaf/stack theory, motivic period theory, and so on—and as morphisms $u : e \to d$ the *refinements*: inclusions of a subdomain, restrictions of scope, or specialization functors that express "e is a more local or more special context inside d." Composition is composition of refinements and each domain has an identity refinement.

Definition 2.2 (Structures in a domain)**.** For each $d \in \mathsf{Dom}$ let Math_d be a category (or higher category) whose objects are the mathematical structures native to d and whose morphisms are the structure-preserving maps of d. A refinement $u : e \to d$ induces a *restriction functor* $u^* : \mathsf{Math}_d \to \mathsf{Math}_e$ (read: view a d-structure through the more special context e). We do not require the Math_d to be glued into a single category; the whole point of modularity is that they are indexed by Dom and related only by the refinement functors.

Definition 2.3 (Physical target category)**.** Phys is a category (in general a bicategory or ∞-category) of *physical representation targets*. Representative objects include: state spaces (Hilbert spaces, phase spaces), observable algebras (C^*- or von Neumann algebras), process categories (categories of histories or channels), gauge moduli (quotient stacks of field configurations), quantum code spaces (logical subsystems of a physical Hilbert space), and measurement-outcome sets. Morphisms are the physically meaningful maps between such targets: isometries, $*$-homomorphisms, completely positive maps, gauge-equivariant maps, and so on.

Definition 2.4 (Physical-representation bracket)**.** Given a chosen interpretation, we write

$$[\![M]\!]_{\mathrm{phys}} \in \mathsf{Phys}$$

for the physical representation assigned to a mathematical structure $M \in \mathsf{Math}_d$. The bracket is a partial, interpretation-dependent operation: $[\![M]\!]_{\mathrm{phys}}$ is defined only when an entry supplying it has been declared, and its epistemic reliability is exactly the status of that entry.

2.2 Realization channels

The passage from abstract information to a concrete physical representation is not unique: the same abstract object can be realized through different *channels*. We index channels by a symbol α.

Definition 2.5 (Realization channel)**.** A *realization channel* is an index α drawn from

$$\begin{aligned}\mathsf{Chan} = \{\, &\text{Betti},\ \text{de Rham},\ \text{Hodge},\ \text{étale},\ p\text{-adic},\ \text{quantum},\\ &\text{thermodynamic},\ \text{computational},\ \text{experimental},\ \text{operational}\,\},\end{aligned}$$

together with a functor $\mathrm{Real}_\alpha : \mathsf{Info}_d \to \mathsf{Rep}_d$ from a category of abstract-information objects to a category of physical representations. Different α realize the same abstract object into different but comparable physical data; comparison morphisms between channels are the *comparison isomorphisms* (e.g. the period pairing relating Betti and de Rham channels in Part II).

2.3 The groupoid-valued assignment

Definition 2.6 (Groupoid of entries)**.** For $d \in \mathsf{Dom}$, let $\mathrm{Rep}_{\mathrm{phys}}(d)$ be the *groupoid* whose objects are the representation entries native to d (Theorem 3.1 below) and whose morphisms are the *equivalences* of entries—isomorphisms of the underlying mathematical structures that are compatible with the translation data and that represent physically irrelevant changes (dualities, gauge changes, code stabilizers, changes of coordinates). We take a groupoid rather than a category because, at the level of representation *content*, only invertible identifications are physically meaningful; non-invertible processes live in Phys, not among the entries.

The assignment $d \mapsto \mathrm{Rep}_{\mathrm{phys}}(d)$, together with the restriction functors induced by refinements, is the pseudofunctor $\mathrm{Rep}_{\mathrm{phys}} : \mathsf{Dom}^{\mathrm{op}} \to \mathsf{Gpd}$ that we make precise in Section 3. The final ingredient we shall need is a notion of covering.

Definition 2.7 (Coverage on Dom)**.** A *Grothendieck topology* (or coverage) J on Dom assigns to each domain d a collection of *covering families* $\{u_i : e_i \to d\}_{i\in I}$ of refinements, closed under composition, base change, and containing all isomorphisms, and interpreted physically as: "the local contexts e_i jointly exhaust the content of d." We call the pair (Dom, J) a *site of domains*. Coverage encodes "contextual coverage": a family of special or local mathematical viewpoints that together determine the global one.

3 The Representation Stack

We now give the four central definitions of the formalism. They are stated so as to be independent of the higher-categorical subtleties; Section 6 supplies the theorems that make them useful, and Part VI supplies the full descent-theoretic underpinning.

Definition 3.1 (Representation entry)**.** A *representation entry* is a quadruple

$$E = (M, P, \tau, \sigma),$$

where $M \in \mathsf{Math}_d$ is a mathematical structure, $P \in \mathsf{Phys}$ is a proposed physical representation, τ is a *semantic translation datum*, and $\sigma \in \{\mathsf{S}, \mathsf{H}, \mathsf{P}\}$ is the epistemic status label. The translation datum τ may be a functor $\mathsf{Math}_d \to \mathsf{Phys}$, a natural transformation between such, an equivalence class of models, a physical interpretation map, or a weaker relation; the strength of τ is exactly what σ records.

We formalize the graded strength of τ explicitly, because it is what the status calculus (Theorem 6.5) tracks.

Definition 3.2 (Translation strength)**.** A translation datum τ from $M \in \mathsf{Math}_d$ to $P \in \mathsf{Phys}$ is one of the following, listed in decreasing strength:

1. a *functorial translation*: an actual functor $F : \mathsf{Math}_d \to \mathsf{Phys}$ with $F(M) \cong P$ (candidate status S);
2. a *natural translation*: a natural transformation between declared functors witnessing $M \rightsquigarrow P$ (candidate status S/H);
3. an *interpretive rule*: a partial assignment $M \mapsto P$ justified by analogy or dictionary (candidate status H);
4. a *speculative map*: a proposed but unconstructed assignment (candidate status P).

The word "candidate" is deliberate: the actual label σ of an entry is assigned by the author and may be weaker than the nominal strength of τ (e.g. a genuine functor whose *physical interpretation* is only heuristic carries a composite label S/H).

Definition 3.3 (Representation library)**.** A *mathematics-to-physics representation library* is a collection $\mathcal{L}$ of representation entries that is closed, whenever the operations are defined, under:

1. *restriction* to subdomains along refinements $u : e \to d$;
2. *transport* along equivalences of mathematical structures;
3. *composition* of compatible translations;
4. *decomposition* through boundary, coproduct, or localization operations;
5. *realization* into observables via the pipeline of Section 4.

Definition 3.4 (Representation prestack)**.** A *representation prestack* is a pseudofunctor

$$\mathrm{Rep}_{\mathrm{phys}} : \mathsf{Dom}^{\mathrm{op}} \longrightarrow \mathsf{Gpd}$$

assigning to each domain d the groupoid $\mathrm{Rep}_{\mathrm{phys}}(d)$ of representation entries (Theorem 2.6) and to each refinement $u : e \to d$ a restriction functor $u^* : \mathrm{Rep}_{\mathrm{phys}}(d) \to \mathrm{Rep}_{\mathrm{phys}}(e)$, together with coherent composition isomorphisms $\gamma_{u,v} : (uv)^* \Rightarrow v^*u^*$ satisfying the pseudofunctor cocycle conditions.

Definition 3.5 (Representation stack)**.** Let (Dom, J) be a site of domains. A representation prestack $\mathrm{Rep}_{\mathrm{phys}}$ is a *representation stack* if, for every covering family $\{u_i : e_i \to d\}_{i \in I} \in J(d)$, compatible families of local representation entries glue to a global entry uniquely up to coherent isomorphism; equivalently (Theorem 6.3), the comparison functor from $\mathrm{Rep}_{\mathrm{phys}}(d)$ to the groupoid of J-descent data is an equivalence.

Remark 3.6. The word "stack" is used broadly. The groupoids $\mathrm{Rep}_{\mathrm{phys}}(d)$ record physically irrelevant equivalences: gauge changes, duality transformations, code stabilizers, or changes of coordinates. Part I *posits* this stack-like behavior by analogy; Part VI supplies its rigorous descent-theoretic meaning. The reader should treat Theorem 3.5 as a promissory note whose payment is scheduled for Part VI, and Theorem 6.3 as the interface contract between them.

3.1 The master diagram

The canonical picture of the entire program is the following five-node pipeline, with a feedback loop expressing that gauge/equivalence/duality-related structures enter the pipeline as the *same* abstract information.

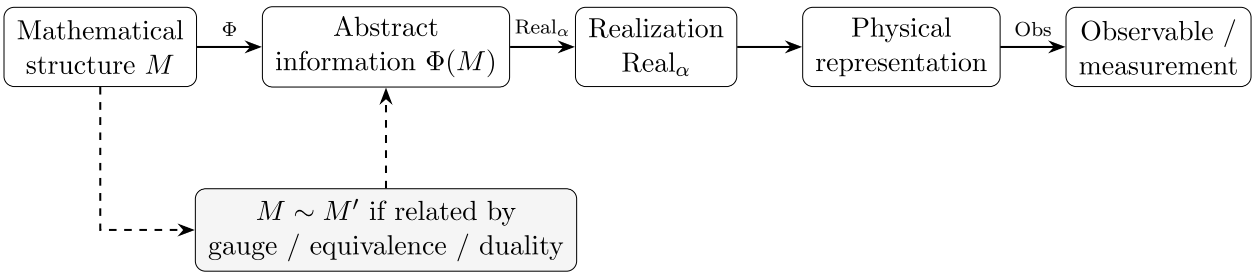

Figure 1: The master diagram of the realization pipeline. A mathematical structure M is sent by Φ to abstract physical information, realized through a channel Real_α into a physical representation, from which Obs extracts measurement data. The dashed feedback expresses the Equivalence axiom: structures related by gauge, equivalence, or duality feed identical abstract information into the pipeline.

The same pipeline can be drawn as a commuting diagram of categories, which is the form used in Theorem 6.1:

$$\begin{tikzcd} \mathsf{Math}_d \arrow[r, "\Phi"] \arrow[rrr, bend right, "\mathrm{Obs}_\alpha"'] & \mathsf{Info}_d \arrow[r, "\mathrm{Real}_\alpha"] & \mathsf{Rep}_d \arrow[r, "\mathrm{Obs}"] & \mathsf{Meas}_d \end{tikzcd}$$

4 The Realization Pipeline

Definition 4.1 (Realization pipeline)**.** Fix a domain d and a channel α. The *realization pipeline* is the composite

$$\mathrm{Obs}_\alpha \;=\; \mathrm{Obs} \circ \mathrm{Real}_\alpha \circ \Phi \;:\; \mathsf{Math}_d \xrightarrow{\Phi} \mathsf{Info}_d \xrightarrow{\mathrm{Real}_\alpha} \mathsf{Rep}_d \xrightarrow{\mathrm{Obs}} \mathsf{Meas}_d,$$

where Φ extracts the abstract physical-information content of a structure, Real_α realizes it into a physical representation through channel α, and Obs extracts measurement data. On objects,

$$\mathrm{Obs}_\alpha(M) \;=\; \mathrm{Obs}\big(\mathrm{Real}_\alpha(\Phi(M))\big).$$

The pipeline is governed by four axioms. These are the axioms every later module instantiates concretely.

Axiom 1 (Realization)**.** Every observable is the output of the pipeline: observable $= \mathrm{Obs} \circ \mathrm{Real}_\alpha$(abstract structure). There is no observable content except through a declared channel α.

Axiom 2 (Equivalence)**.** Mathematical descriptions related by an equivalence lying in the automorphism/gauge groupoid of a representation entry determine the *same* physical content at the chosen observational level. Formally, if $g : M \xrightarrow{\sim} M'$ is a morphism in the groupoid $\mathrm{Rep}_{\mathrm{phys}}(d)$ then there is a canonical isomorphism $\mathrm{Obs}_\alpha(M) \cong \mathrm{Obs}_\alpha(M')$ in Meas_d; when Meas_d is skeletal (e.g. a bare set of numerical measurement outcomes) this canonical isomorphism is a genuine equality $\mathrm{Obs}_\alpha(M) = \mathrm{Obs}_\alpha(M')$.

Axiom 3 (Locality / descent)**.** Physical representations must be locally assignable and globally coherent: for a covering $\{e_i \to d\}$, a global representation is determined by compatible local ones. Failure of descent is not a defect of the formalism but a physical phenomenon—an obstruction, anomaly, or missing boundary datum.

Axiom 4 (Decomposition)**.** Compositional or period-like observables admit a decomposition operation (boundary ∂, coproduct Δ, coaction, spectral sequence, or factorization) revealing lower-level informational pieces. The decomposition is functorial: it commutes with compatible realization channels.

Example 4.2 (TQFT as a status-S pipeline)**.** Let d be algebraic topology and take Φ to send an n-manifold to its bordism data, Real_α to be a fixed TQFT functor $Z : \mathsf{Bord}_n \to \mathsf{Vect}$, and Obs to read off partition functions and correlation numbers. Then Obs_α is an honest functor and the entry carries status S: the realization is an actually-constructed functor. This is the paradigm the other modules aim to emulate. It is developed fully in Part IV.

Example 4.3 (A status-H pipeline)**.** Let d be motivic period theory, Φ send a variety X to its motive $\mathrm{Mot}(X)$ read as "functional essence," Real_α be a period realization, and Obs take numerical periods. The mathematics of realization functors on motives is standard (status S *as mathematics*), but the reading of the motive as the physical "functional essence" of X is heuristic, so the *entry* carries the composite label S/H. This is developed in Part II, where the caution that a motive is not literally "the functions of" X is stated prominently.

5 Axioms for a Representation Ontology

The four pipeline axioms of Section 4 are operational. The source program also presents a parallel, more philosophical set of six *ontology* axioms, which we reproduce because the synthesis Part treats them as the target statement the whole ladder discharges. We then note their relationship to the pipeline axioms.

Ontology Axiom 1 (Syntax)**.** Mathematics is the syntax of physical representation: physical systems, states, and processes are carried by mathematical structures $M \in \mathsf{Math}_d$.

Ontology Axiom 2 (Semantics)**.** The functional essence of a structure—its invariant prenumerical content—is its semantics; in Part II this is made precise as a motive.

Ontology Axiom 3 (Realization (ontological))**.** Observables arise by realizing semantic content through a channel: $\mathrm{Obs}_\alpha(M) = \mathrm{Obs}(\mathrm{Real}_\alpha(\Phi(M)))$. This is Axiom 1 read ontologically.

Ontology Axiom 4 (Equivalence / gauge)**.** Descriptions differing by gauge, equivalence, or duality determine the same physical content; this is Axiom 2 read ontologically and is discharged type-theoretically (univalence, Part V) and geometrically (descent, Part VI).

Ontology Axiom 5 (Decomposition (ontological))**.** Observable content decomposes into primitive informational pieces via a coalgebraic operation; this is Axiom 4 read ontologically and is made concrete by the Goncharov coproduct (Part II) and Connes–Kreimer coproduct (Part VI).

Ontology Axiom 6 (Motivic pre-numericality)**.** Numbers extracted from physics are realizations of pre-numerical motivic data: $\mathcal{A} = \mathrm{per}(\mathcal{A}^{\mathfrak{m}})$. This is the amplitude-level refinement of Axiom 1; it drives Part II.

Conjecture 5.1 (Motivic-categorical information ontology)**.** A sufficiently broad class of physical theories admits a representation-stack description in which states, processes, observables, amplitudes, gauge redundancies, and measurement values arise as realizations of structured categorical, homotopical, geometric, and motivic information. Equivalently, for such theories the prestack $\mathrm{Rep}_{\mathrm{phys}}$ built from the modules of Parts II–VI is a representation stack in the sense of Theorem 3.5.

Remark 5.2 (Relationship of the two axiom sets). The pipeline axioms (Axiom 1–Axiom 4) are what one *uses* to build entries; the ontology axioms (Ontology Axiom 1–Ontology Axiom 6) are what one *claims* the entries collectively mean. Axiom 1 and Ontology Axiom 3 coincide; Axiom 2 and Ontology Axiom 4 coincide; Axiom 4 and Ontology Axiom 5 coincide; the Locality axiom (Axiom 3) is the operational shadow of Theorem 5.1's demand that $\mathrm{Rep}_{\mathrm{phys}}$ be a *stack*, i.e. that descent hold. Ontology Axiom 1, Ontology Axiom 2, and Ontology Axiom 6 have no operational counterpart in Section 4; they are the semantic layer Part II supplies.

6 Results

We prove the four main results. Theorem 6.1 and Theorem 6.3 concern the pipeline and the stack; Theorem 6.5 turns the status system into a calculus; Theorem 6.8 is the first appearance of the "stacks retain gauge information" phenomenon that recurs across the whole series.

6.1 Functoriality of the realization pipeline

Proposition 6.1 (Functoriality of the realization pipeline)**.** *Fix a domain d and a channel α. Suppose the three stages*

$$\Phi : \mathsf{Math}_d \to \mathsf{Info}_d, \qquad \mathrm{Real}_\alpha : \mathsf{Info}_d \to \mathsf{Rep}_d, \qquad \mathrm{Obs} : \mathsf{Rep}_d \to \mathsf{Meas}_d$$

are each functors. Then the assignment $M \mapsto \mathrm{Obs}_\alpha(M) = \mathrm{Obs}(\mathrm{Real}_\alpha(\Phi(M)))$ extends to a functor $\mathrm{Obs}_\alpha : \mathsf{Math}_d \to \mathsf{Meas}_d$. Moreover the assignment $\alpha \mapsto \mathrm{Obs}_\alpha$ is functorial in the channel: a comparison morphism $\theta : \alpha \Rightarrow \beta$ (a natural transformation $\mathrm{Real}_\alpha \Rightarrow \mathrm{Real}_\beta$) induces a natural transformation $\mathrm{Obs}_\alpha \Rightarrow \mathrm{Obs}_\beta$.

Proof. The composite of functors is a functor: on objects, $(\mathrm{Obs} \circ \mathrm{Real}_\alpha \circ \Phi)(M)$ is defined as above; on a morphism $f : M \to M'$ set $\mathrm{Obs}_\alpha(f) = \mathrm{Obs}(\mathrm{Real}_\alpha(\Phi(f)))$. Functoriality is inherited stagewise. Identities: $\mathrm{Obs}_\alpha(\mathrm{id}_M) = \mathrm{Obs}(\mathrm{Real}_\alpha(\Phi(\mathrm{id}_M))) = \mathrm{Obs}(\mathrm{Real}_\alpha(\mathrm{id}_{\Phi M})) = \mathrm{Obs}(\mathrm{id}_{\mathrm{Real}_\alpha \Phi M}) = \mathrm{id}_{\mathrm{Obs}_\alpha M}$, using that each stage preserves identities. Composition: for $f : M \to M'$, $g : M' \to M''$,

$$\begin{aligned}\mathrm{Obs}_\alpha(g \circ f) &= \mathrm{Obs}\mathrm{Real}_\alpha\Phi(g \circ f) = \mathrm{Obs}\mathrm{Real}_\alpha\big(\Phi(g) \circ \Phi(f)\big)\\ &= \mathrm{Obs}\big(\mathrm{Real}_\alpha\Phi(g) \circ \mathrm{Real}_\alpha\Phi(f)\big) = \mathrm{Obs}_\alpha(g) \circ \mathrm{Obs}_\alpha(f),\end{aligned}$$

again stagewise. Hence Obs_α is a functor. For the second claim, let $\theta : \mathrm{Real}_\alpha \Rightarrow \mathrm{Real}_\beta$ be natural. Define $\Theta_M := \mathrm{Obs}(\theta_{\Phi M}) : \mathrm{Obs}_\alpha(M) \to \mathrm{Obs}_\beta(M)$. For $f : M \to M'$ naturality of θ at $\Phi(f)$ gives $\theta_{\Phi M'} \circ \mathrm{Real}_\alpha\Phi(f) = \mathrm{Real}_\beta\Phi(f) \circ \theta_{\Phi M}$; applying the functor Obs and using its functoriality yields $\Theta_{M'} \circ \mathrm{Obs}_\alpha(f) = \mathrm{Obs}_\beta(f) \circ \Theta_M$, i.e. Θ is natural. Concretely $\Theta = \mathrm{Obs} * \theta * \Phi$ is the horizontal composition (whiskering / Godement product) of θ with the functors Obs and Φ, so its naturality is the standard 2-categorical fact that whiskering a natural transformation by functors yields a natural transformation. This is exactly the naturality that Part II's period comparison (Betti vs. de Rham) instantiates. □

Remark 6.2 (Where the status labels bite). Theorem 6.1 is a triviality about functor composition; its *force* is entirely in the hypothesis that each stage is a functor. An entry earns status S precisely when this hypothesis is discharged by an actual construction (as in Theorem 4.2). An H- or P-entry only posits the stages, so the same formula holds *formally* but is not backed by a construction. The theorem thus makes explicit that "Obs_α is a functor" is a claim whose reliability equals the status of its weakest stage—which is precisely what Theorem 6.5 quantifies.

6.2 When is the prestack a stack?

Theorem 6.3 (Descent characterization of stackhood)**.** *Let* (Dom, J) *be a site of domains and* $\mathrm{Rep}_{\mathrm{phys}} : \mathsf{Dom}^{\mathrm{op}} \to \mathsf{Gpd}$ *a representation prestack. Then* $\mathrm{Rep}_{\mathrm{phys}}$ *is a representation stack (Theorem 3.5) if and only if for every* $d \in \mathsf{Dom}$ *and every covering family* $\{u_i : e_i \to d\}_{i\in I} \in J(d)$ *the natural comparison functor*

$$\Lambda_d : \mathrm{Rep}_{\mathrm{phys}}(d) \longrightarrow \mathrm{Desc}\big(J; \{u_i\}; \mathrm{Rep}_{\mathrm{phys}}\big)$$

into the groupoid of J-descent data is an equivalence of groupoids. Here a descent datum is a family $(E_i)_i$ *with* $E_i \in \mathrm{Rep}_{\mathrm{phys}}(e_i)$ *together with gluing isomorphisms* $\phi_{ij} : \mathrm{pr}_i^* E_i \xrightarrow{\sim} \mathrm{pr}_j^* E_j$ *on the pairwise overlaps satisfying the cocycle condition* $\phi_{jk} \circ \phi_{ij} = \phi_{ik}$ *on triple overlaps (suppressing, as is standard, the pullbacks of each* ϕ *to the common triple fiber product* $e_i \times_d e_j \times_d e_k$ *along the evident projections). Here* $\mathrm{pr}_i : e_i \times_d e_j \to e_i$ *and* $\mathrm{pr}_j : e_i \times_d e_j \to e_j$ *are the two projections out of the fiber product (intersection)* $e_i \times_d e_j$*, which exists by the base-change closure of the coverage (Theorem 2.7); we use projection subscripts (not transition-map subscripts) throughout, so* ϕ_{ij} *merely indexes the ordered overlap* (i, j)*. The triple overlaps are the fiber products* $e_i \times_d e_j \times_d e_k$*.*

Proof. ($\Rightarrow$) Suppose $\mathrm{Rep}_{\mathrm{phys}}$ is a stack. By Theorem 3.5, compatible families of local entries glue to a global entry uniquely up to coherent isomorphism. "Compatible family" is exactly a descent datum; "glues to a global entry" says Λ_d is essentially surjective; "uniquely up to coherent isomorphism" says Λ_d is fully faithful. Hence Λ_d is an equivalence.

($\Leftarrow$) Suppose each Λ_d is an equivalence. Essential surjectivity gives existence of a gluing (every descent datum comes from a global entry); fullness and faithfulness give uniqueness up to unique compatible isomorphism. This is Theorem 3.5.

We record that the equivalence, when it fails, fails by a controlled amount: the prestack always admits a *stackification* $\mathrm{Rep}_{\mathrm{phys}} \to \mathrm{Rep}^{+}_{\mathrm{phys}}$, the universal map to a stack, obtained by the plus-construction that formally adjoins gluings of descent data. $\mathrm{Rep}_{\mathrm{phys}}$ is already a stack iff the unit $\mathrm{Rep}_{\mathrm{phys}} \to \mathrm{Rep}^{+}_{\mathrm{phys}}$ is an equivalence, iff each Λ_d is an equivalence. The construction of $\mathrm{Rep}^{+}_{\mathrm{phys}}$ and the verification that it is a stack are exactly the descent-theoretic content discharged rigorously in Part VI; here we use only its universal property, which is what makes the "if and only if" meaningful without circularity. □

Remark 6.4 (Forward reference to Part VI). Theorem 6.3 is stated as an interface. Its right-hand side—the groupoid $\mathrm{Desc}(J; \{u_i\}; \mathrm{Rep}_{\mathrm{phys}})$ and the stackification $\mathrm{Rep}^{+}_{\mathrm{phys}}$—is constructed in full in Part VI, where descent for a Grothendieck topology is developed properly (following Giraud and Vistoli). Part I is entitled to *state* the criterion and to reason with it, because both directions are formal once "descent datum" and "stackification" are granted their standard meanings. This is the precise sense in which Part VI "pays" Part I's promissory note.

6.3 The status calculus

We now promote the informal $\mathsf{S}/\mathsf{H}/\mathsf{P}$ system to an algebraic structure, so that the status of a *composite* translation is computed, not guessed. Order the labels by reliability,

$$\mathsf{S} \preceq \mathsf{H} \preceq \mathsf{P},$$

read "S is at least as reliable as H, which is at least as reliable as P."

Theorem 6.5 (Status calculus). *Let $\Sigma = \{\mathsf{S}, \mathsf{H}, \mathsf{P}\}$ with the total order $\mathsf{S} \preceq \mathsf{H} \preceq \mathsf{P}$.*

1. *$(\Sigma, \vee, \mathsf{S})$ is a commutative idempotent monoid, where $a \vee b := \max_{\preceq}(a, b)$ is the* less reliable *of the two labels — their* join *(supremum) in the reliability order — and S, the $\preceq$-least (most reliable) element, is the unit.*

2. *If representation entries $E_1 = (M_1, P_1, \tau_1, \sigma_1)$ and $E_2 = (M_2, P_2, \tau_2, \sigma_2)$ compose (i.e. $\tau_2 \circ \tau_1$ is defined, so $P_1 = M_2$ in the evident sense), then the composite entry $E_2 \circ E_1$ has status*

$$\sigma(E_2 \circ E_1) \;=\; \sigma_1 \vee \sigma_2.$$

   In particular the status value *is* monotone non-decreasing *under composition in the $\preceq$ order—equivalently,* reliability *is monotone non-increasing: $\sigma(E_2 \circ E_1) \preceq \sigma_i$ fails in general, while $\sigma_i \preceq \sigma(E_2 \circ E_1)$ always holds—a chain of translations is only as reliable as its weakest link.*

3. *Regard a representation library $\mathcal{L}$ as a category whose objects are* all *the carriers appearing in its entries—the union* $\mathsf{Math}_d \cup \mathsf{Info}_d \cup \mathsf{Rep}_d \cup \mathsf{Phys} \cup \mathsf{Meas}_d$ *of mathematical, informational, and physical targets—and whose morphisms are entries under composition (so a pipeline stage* $\mathsf{Math}_d \to \mathsf{Info}_d$ *and an entry* $\mathsf{Info}_d \to \mathsf{Rep}_d$ *compose as morphisms with matching object types), and let $B(\Sigma, \vee)$ be the one-object category obtained by delooping the status monoid (its unique hom-monoid is* $(\Sigma, \vee)$*). Then the assignment $\sigma : \mathcal{L} \to B(\Sigma, \vee)$ sending each entry to its status label is a* functor*; it is the universal composition-respecting,* S*-normalized status invariant.*

Proof. (1) $\vee = \max_{\preceq}$ on a totally ordered set is associative, commutative, and idempotent, and the $\preceq$-least element S is its identity: $\mathsf{S} \vee a = \max_{\preceq}(\mathsf{S}, a) = a$. Hence $(\Sigma, \vee, \mathsf{S})$ is a commutative idempotent monoid (a bounded join-semilattice with bottom S and top P).

(2) By the intended semantics of the labels (Section 1.4): S asserts that an actual functorial/natural construction exists; H asserts a heuristic interpretive rule; P asserts a speculative claim. A composite $\tau_2 \circ \tau_1$ inherits the constructions of *both* factors, so it can be no more reliable than the less reliable factor: if either τ_i is merely speculative, the composite passes through a speculative step and is speculative; if the weaker of the two is heuristic and the other standard, the composite is heuristic; only if both are standard is the composite standard. This is exactly $\sigma_1 \vee \sigma_2 = \max_{\preceq}(\sigma_1, \sigma_2)$. Monotonicity is then $\sigma_i \preceq \max_{\preceq}(\sigma_1, \sigma_2)$, which holds by definition of max.

(3) Composition of entries is associative with units the identity entries $(M, M, \mathrm{id}, \mathsf{S})$, so $\mathcal{L}$ is a category. Delooping the commutative monoid $(\Sigma, \vee, \mathsf{S})$ yields the one-object category $B(\Sigma, \vee)$ whose single hom-set is Σ with composition $\vee$ and identity S. The assignment σ sends every object of $\mathcal{L}$ to the unique object of $B(\Sigma, \vee)$ and every entry to its label; it preserves identities, $\sigma(\mathrm{id}) = \mathsf{S}$, and composition, $\sigma(E_2 \circ E_1) = \sigma(E_2) \vee \sigma(E_1)$ by (2). Hence σ is a functor $\mathcal{L} \to B(\Sigma, \vee)$. (We emphasize that $\mathcal{L}$ is a genuine category, not a monoid, so the target is the *delooping* $B(\Sigma, \vee)$ and no monoidal structure on $\mathcal{L}$ is assumed or needed.) Universality: any composition-respecting, identity-normalized invariant valued in a totally ordered monoid factors through σ, because σ is determined on generators (single entries) and extended by $\vee$, which is forced by (2). □

Corollary 6.6 (Reliability of a pipeline)**.** *The status of a realization pipeline* $\mathrm{Obs}_\alpha = \mathrm{Obs} \circ \mathrm{Real}_\alpha \circ \Phi$ *viewed as a composite of three entries is* $\sigma(\Phi) \vee \sigma(\mathrm{Real}_\alpha) \vee \sigma(\mathrm{Obs})$*. Consequently an* S*-labeled pipeline requires* all three *stages to be* S*; a single heuristic stage downgrades the whole pipeline to* H.

Proof. Immediate from Theorem 6.5(2) applied twice and associativity of $\vee$. This is the formal content of Theorem 6.2. □

Remark 6.7. Theorem 6.5 is original to this program: it operationalizes the paper's status system into a checkable calculus, valuable for automated (Lean-checked) status propagation. The Haskell and Lean encodings of Section 9 implement $\vee$ directly, so that composing entries *computes* the composite status rather than requiring the author to assign it by hand.

6.4 Coarse quotients lose gauge information

The last theorem is the Part I shadow of a phenomenon that recurs, at increasing rigor, throughout the series: quotienting away automorphisms destroys descent.

Theorem 6.8 (Coarse-quotient obstruction)**.** *Suppose* $\mathrm{Rep}_{\mathrm{phys}}$ *is a representation stack and let* $\pi(d) := \pi_0(\mathrm{Rep}_{\mathrm{phys}}(d))$ *be the set of isomorphism classes of entries (the "coarse" representation library, obtained by quotienting each groupoid by its gauge equivalences). Then* $d \mapsto \pi(d)$ *is a presheaf of sets, but in general it* can fail *to be a sheaf. Concretely, suppose some entry* $E \in \mathrm{Rep}_{\mathrm{phys}}(d)$ *has a nontrivial automorphism group* $\mathrm{Aut}(E) \neq 1$ *and the site is rich enough to admit a covering* $\{e_i \to d\}$ *supporting a nontrivial* $\mathrm{Aut}(E)$*-torsor (equivalently, a nontrivial Čech* 1*-cocycle class valued in* $\mathrm{Aut}(E)$*, so that* $\check{H}^1(\{e_i\}; \mathrm{Aut}(E)) \neq 0$*). Then there is a covering on which local classes agree on overlaps yet fail to glue to a unique global class, so* π *is not a sheaf.*

Proof. Presheaf: π_0 is functorial (a functor of groupoids induces a map on isomorphism-class sets), and restriction functors u^* induce restriction maps $\pi(d) \to \pi(e)$; the pseudofunctor coherence of $\mathrm{Rep}_{\mathrm{phys}}$ makes these strictly functorial after passing to π_0 (all 2-cells become identities on π_0). Hence π is a presheaf.

Failure of the sheaf condition: pick $E \in \mathrm{Rep}_{\mathrm{phys}}(d)$ with a nontrivial automorphism $g \in \mathrm{Aut}(E)$, and choose a covering $\{u_i : e_i \to d\}$ that is "fine enough" that g becomes locally trivializable, i.e. each restriction $u_i^* g$ is isotopic to the identity in $\mathrm{Rep}_{\mathrm{phys}}(e_i)$ while g itself is not the identity globally. (Such a covering exists whenever the automorphism is supported by the gluing data—this is the generic situation for gauge automorphisms; in Part VI this is realized concretely for $[X/G]$-type stacks with G acting with stabilizers.) Let E^g be the distinct global entry obtained by regluing the *same* local data $(u_i^* E)_i$ along the descent datum twisted by the nontrivial 1-cocycle valued in $\mathrm{Aut}(E)$ determined by g (a "twisted sector"). Then E and E^g have equal restrictions to every e_i and equal overlap comparisons in π_0, hence define the *same* element of the sheaf of local sections of π; but $E \not\cong E^g$ as global entries, since the nontrivial cocycle class obstructs a global identification when $g \neq \mathrm{id}$. Thus the section over $\{e_i\}$ has (at least) two preimages in $\pi(d)$, violating the uniqueness half of the sheaf axiom. Existence can fail dually. Therefore π is not a sheaf. □

Corollary 6.9. *The stack* $\mathrm{Rep}_{\mathrm{phys}}$ *is strictly more informative than its coarse library* π*: passing to* π *forgets exactly the automorphism (gauge/stabilizer) data, and this data is what descent requires. Retaining it is the entire reason the formalism is built on groupoid-valued (stack) rather than set-valued (sheaf) assignments.*

Proof. By Theorem 6.8, π fails descent exactly where $\mathrm{Rep}_{\mathrm{phys}}$ has nontrivial automorphisms; since $\mathrm{Rep}_{\mathrm{phys}}$ satisfies descent by hypothesis, the difference between them is precisely the retained automorphism data. This is the Part I statement of the recurring theorem "stacks retain gauge information," proved algebro-geometrically in Part III, type-theoretically in Part V, and descent-theoretically in Part VI. □

7 The Core Translation Library

We reproduce the foundational dictionary that Parts II–VI extend. Every row carries a status label per Section 1.4. The library is presented as three tables: the core translation grammar (Table 2), the operations dictionary (Table 3), and the domains-as-faculties dictionary (Table 4). These are shared across all seven parts; later modules cite them rather than restating them.

Table 2: Core translation library (foundations grammar).

Mathematics	**Physical representation**	**Semantic interpretation**	**Status**
Object X	Physical system or world-structure	Structured carrier of physical information	H
Representation $\rho : X \to \mathrm{End}(V)$	Physical realization on states	How structure acts on observables/states	S
Realization functor	Translation into measurable form	Abstract information to readable data	S/H
Invariant	Conservation law / observable	What is stable under allowed transformations	S
Equivalence	Same physics, different coordinates	Dual descriptions, gauge equivalence	S
Symmetry	Redundancy or physical transformation	Law-preserving structure	S
Duality	Two theories, same content	Equivalence of representation languages	S
Moduli space	Space of distinct configurations	Parameter space of possible structures	S
Quotient	Identification of redundant descriptions	Gauge reduction or physical state space	S
Boundary	Physical interface / asymptotic region	Where factorization/measurement appears	S/H
Singularity	Threshold / phase transition	Breakdown or special physical regime	S/H
Decomposition	Factorization, cut, measurement split	Observable broken into primitive pieces	S
Primitive	Irreducible informational unit	Indecomposable in the chosen grammar	H
Functor	Physical theory as translation	States to spaces, processes to histories	S
Natural transformation	Equivalence/deformation of theories	Consistent translation between descriptions	S/H

continued on next page

Table 2, continued

Mathematics	**Physical representation**	**Semantic interpretation**	**Status**
Category	Universe of systems and processes	Objects are systems; morphisms are transitions	S/H

Table 3: Operations $\leftrightarrow$ physical interpretation (shared appendix).

Operation	**Physical interpretation**
Boundary ∂	Edge / flux / factorization channel
Differential d	Infinitesimal change
Coboundary d^*	Gauge / exact shift
Cobracket δ	Primitive antisymmetric split
Coproduct Δ	Full decomposition split
Residue	Pole / singular channel
Monodromy	Analytic continuation
Filtration	Scale hierarchy
Grading	Charge / degree / loop / weight
Spectral sequence	Multi-scale obstruction
Localization	Effective physics at a scale
Completion	Perturbation expansion
Blowup	Regularize a divergence
Quotient	Gauge reduction
Pullback	Constraint matching
Pushout	Couple systems
Trace	Partition function / loop amplitude
Dual	Measurement / conjugate
Tensor product $\otimes$	Parallel composition
Direct sum $\oplus$	Superselection
Hom	Space of processes
Ext	Bound state / anomaly / deformation
Tor	Hidden compatibility failure

Table 4: Mathematical domains as physical faculties (shared appendix).

Domain	**Physical faculty**
Algebra	Composition of observables / transformations

continued on next page

Table 4, continued

Domain	Physical faculty
Coalgebra	Decomposition of observables / processes
Topology	Global invariants under deformation
Algebraic topology	Conserved charges / defects / anomalies / sectors
Differential geometry	Smooth fields / curvature / spacetime dynamics
Algebraic geometry	Solution spaces / moduli / singularities / periods
Arithmetic geometry	Hidden number-theoretic structure
Category theory	Composition / translation / systems / processes
Higher category theory	Defects / interfaces / higher processes
Homotopy type theory	Internal logic of spaces / equivalences
Sheaf theory	Local-to-global information
Stack theory	Gauge-correct local-to-global information
Motive theory	Universal functional essence
Homological algebra	Obstructions / resolutions / derived consistency
K-theory	Stable charge / vector-bundle classification
Operator algebra	Quantum observables / noncommutative spaces
Representation theory	Symmetry becomes physical action
Symplectic geometry	Classical phase space
Poisson geometry	Bracket structure of observables
Noncommutative geometry	Geometry from noncommuting observables
Tropical geometry	Degeneration / asymptotics
Cluster algebra	Mutating amplitude coordinates
Operad theory	Interaction vertices / compositional laws

8 Setting up Parts II–VI

The upshot of Part I is that Parts II–VI are, formally, successive concrete instantiations of the *same* three slots: the domain-indexed structure category Math_d, the realization channel Real_α, and the decomposition operation of Axiom 4. We record the instantiations explicitly so that each later module can open by citing this section.

Part II (Motives, periods, amplitudes).
Here d is motivic period theory; Math_d is the category of motives $\mathrm{Mot}(X)$ and motivic amplitude objects $\mathcal{A}^{\mathfrak{m}}$; the channels $\alpha \in \{\mathrm{Betti}, \mathrm{de\ Rham}, \mathrm{Hodge}, \text{étale}\}$ become concrete realizations $H_\alpha(X) \simeq \mathrm{Real}_\alpha(\mathrm{Mot}(X))$; the decomposition operation is the Goncharov

coproduct Δ and cobracket δ. Ontology Axiom 6 ($\mathcal{A} = \mathrm{per}(\mathcal{A}^{\mathfrak{m}})$) is the running formula. This is where Part I's abstract Real_α first acquires numerically checkable content.

Part III (Algebraic geometry).
Here Math_d is schemes, moduli spaces/stacks, and variations of Hodge structure; Part II's motives and periods are now understood as living over and varying across geometric moduli, with the Gauss–Manin connection as the family version of Real_α. The decomposition operation is the residue/boundary structure of positive geometries. Theorem 6.8 is upgraded here to the moduli-stack versus coarse-moduli distinction.

Part IV (Algebraic topology).
Here Math_d is (co)homology, characteristic classes, and bordism; Real_α becomes the honest functor $Z : \mathsf{Bord}_n \to \mathsf{Vect}$ of Theorem 4.2. This is the first fully rigorous, status-S instantiation of the entire pipeline, and the first appearance of functoriality as the ambient grammar. The decomposition operation is the boundary ∂ and the long exact sequences it generates.

Part V (Category theory & HoTT).
Here the implicit functoriality of Part IV is made the explicit universal grammar: Math_d is categories, functors, natural transformations, and monoidal/dagger-compact structure, with HoTT's univalence formalizing Axiom 2. Part V discharges the Equivalence axiom type-theoretically: equivalent representations *are* identified, not merely related, which is the univalent reading of Theorem 6.9.

Part VI (Sheaves, stacks, gauge, quantum HA).
Here Math_d is presheaves/sheaves/stacks as pseudofunctors $\mathsf{Dom}^{\mathrm{op}} \to \mathsf{Gpd}$ (built from Part V's categories), gauge redundancy $[X/G]$, and quantum high-availability codes. Part VI proves Theorem 6.3's right-hand side rigorously, constructs the stackification $\mathrm{Rep}^+_{\mathrm{phys}}$, and thereby discharges Part I's central promissory note: $\mathrm{Rep}_{\mathrm{phys}}$ *is* a stack exactly when local entries glue up to gauge equivalence. The recurring identity

$$\mathrm{Aut}_{[X/G]}(x) \simeq \mathrm{Stab}_G(x)$$

and the high-availability decomposition

$$H_{\mathrm{phys}} \simeq H_{\mathrm{logical}} \otimes H_{\mathrm{gauge}} \oplus H_{\mathrm{error}}$$

live here.

Remark 8.1 (The closure of the ladder)*.* The dependency is genuinely circular and this is a feature, not a bug. Part I *posits* Theorem 3.5 and states Theorem 6.3 as an interface; Part VI *constructs* the descent groupoid and stackification that make Theorem 6.3 a theorem with a full proof; the synthesis Part then records that Parts II–VI assemble a concrete $\mathrm{Rep}_{\mathrm{phys}}$ satisfying Theorem 3.5, thereby turning Theorem 5.1 into a proved statement for the class of theories the modules cover. Figure 2 depicts the closure.

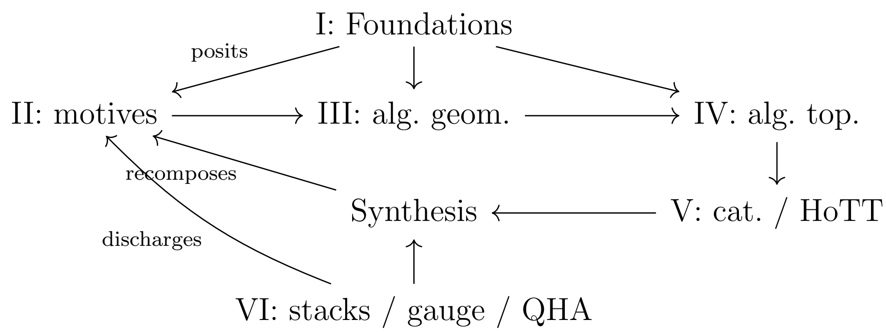

Figure 2: The modular ladder closes: Part I posits the stack; Parts II–VI instantiate it; Part VI discharges Part I's descent promissory note; the synthesis Part recomposes I–VI and verifies Theorem 5.1.

9 Formalization: Haskell and Lean

To honor the "math is code, code is math" bridge (Curry–Howard–Lambek), we provide machine-readable encodings of the core structures. The complete, compiling sources accompany this paper; we display the load-bearing fragments.

Throughout we adopt a *type-theoretic* encoding: a category is represented by a type of objects, an object by a term of that type, and a functor/translation by a function between such types. This is why a localized entry for a structure M carries a function `m -> p`: the field holds the local restriction of a translation functor, and M itself is supplied separately as the term `mathStruct`. A classical 1-category theorist may read `m`, `p` as the object-types of the source and target categories and `translation` as (the object-map of) the functor; this propositions-as-types style deliberately anticipates the HoTT perspective of Part V (objects as terms, functors as functions, equivalences as identities).

9.1 Haskell encoding

The status monoid of Theorem 6.5, the representation entry of Theorem 3.1, and the pipeline of Theorem 4.1 are encoded directly. The key point is that composing entries *computes* the composite status via $\vee$, realizing Theorem 6.6 at the type level.

```
-- Epistemic status, ordered S < H < P  (Theorem: status calculus)
data Status = Std | Heur | Spec deriving (Eq, Ord, Show, Enum, Bounded)

-- Composite status: the LESS reliable of two labels (the monoid join \/),
-- i.e. the supremum in the reliability order Std < Heur < Spec.
joinStatus :: Status -> Status -> Status
joinStatus = max          -- max under Std < Heur < Spec

instance Semigroup Status where (<>) = joinStatus
instance Monoid    Status where mempty = Std   -- Std (bottom) is the unit

-- Graded translation witnesses (Definition: translation strength)
data Translation m p
  = FunctorialTranslation (m -> p)      -- candidate S
```

```
  | NaturalTranslation    (m -> p)      -- candidate S/H
  | InterpretiveRule      (m -> Maybe p)-- candidate H
  | SpeculativeMap        (m -> p)      -- candidate P

-- A representation entry E = (M, P, tau, sigma)
data RepEntry m p = RepEntry
  { mathStruct  :: m
  , physRep     :: p
  , translation :: Translation m p
  , status      :: Status }

-- Realization pipeline  Obs_alpha = Obs . Real_alpha . Phi
data Channel = Betti | DeRham | Hodge | Etale | PAdic | Quantum
             | Thermodynamic | Computational | Experimental | Operational
  deriving (Eq, Show)

realizationPipeline
  :: (i -> r) -> (r -> o)      -- Real_alpha, Obs
  -> (m -> i)                  -- Phi
  -> m -> o
realizationPipeline realA obs phi = obs . realA . phi
```

The composition of entries computes the composite status by `status e2 <> status e1`, which is the join of Theorem 6.5(2). The accompanying `Main.hs` demonstrates: (i) that composing an S entry with an H entry yields H; (ii) that the pipeline of Theorem 4.2 composes to S; and (iii) that a single speculative stage collapses a pipeline to P, exactly Theorem 6.6.

9.2 Lean sketch

The Lean file records the core types, definition-signatures, and a small composition lemma. It is an idiomatic best-effort sketch that elaborates standalone against a recent Lean 4 toolchain (no Mathlib dependency); its purpose is to fix the intended formal statements for later mechanization.

```
-- Derived Ord orders by declaration order: std < heur < spec.
inductive Status where
  | std | heur | spec
  deriving DecidableEq, Repr, Ord

-- composite status = the LESS reliable label = the join (greater) in that order
def Status.join (a b : Status) : Status :=
  match compare a b with
  | .lt => b
  | _   => a

-- Graded translation witnesses, mirroring the Haskell sum type so that the
-- Lean sketch preserves the same semantic nuance (Definition: translation
-- strength).
inductive Translation (M P : Type) where
  | functorial   : (M -> P) -> Translation M P        -- candidate S
  | natural      : (M -> P) -> Translation M P        -- candidate S/H
  | interpretive : (M -> Option P) -> Translation M P -- candidate H
```

```
  | speculative  : (M -> P) -> Translation M P        -- candidate P

structure RepEntry (M P : Type) where
  mathStruct  : M
  physRep     : P
  translation : Translation M P   -- tau, graded by strength (not a bare map)
  status      : Status

-- Translation.comp (full 16-case definition in the accompanying source)
-- composes the underlying maps and degrades strength to the weaker leg, so the
-- composite's candidate strength is never stronger than either factor.
-- Composition of entries joins their statuses (Theorem: status calculus, 2):
-- the composite is only as reliable as its weakest link.
def RepEntry.comp {A B C : Type}
    (e2 : RepEntry B C) (e1 : RepEntry A B) : RepEntry A C :=
  { mathStruct  := e1.mathStruct
  , physRep     := e2.physRep
  , translation := Translation.comp e2.translation e1.translation
  , status      := Status.join e1.status e2.status }

-- The composite of two standard entries is standard.
theorem RepEntry.comp_std {A B C : Type}
    (e2 : RepEntry B C) (e1 : RepEntry A B)
    (h1 : e1.status = Status.std) (h2 : e2.status = Status.std) :
    (RepEntry.comp e2 e1).status = Status.std := by
  have hstat : (RepEntry.comp e2 e1).status
      = Status.join e1.status e2.status := rfl
  rw [hstat, h1, h2]; decide

-- Realization pipeline as a composite of three maps
structure RealizationPipeline (M I R O : Type) where
  phi   : M -> I
  realA : I -> R
  obs   : R -> O

def RealizationPipeline.run {M I R O}
    (P : RealizationPipeline M I R O) : M -> O :=
  fun m => P.obs (P.realA (P.phi m))
```

Part V and Part VI develop the categorical and descent-theoretic Lean structures (pseudofunctors, Grothendieck topologies, descent data) that a full mechanization of Theorem 6.3 would require.

10 Discussion

10.1 Limitations

We reproduce, and commit every later module to respecting, the five explicit limitations of the program.

1. *Not every analogy is a theory.* Status labels are part of the formalism, not decoration;

an H or P entry is a proposal, not a result. Theorem 6.5 makes this quantitative: reliability cannot be manufactured by composing analogies.

2. *Motives are not literally "the functions of" an object.* The functional-essence reading is this program's proposed semantic layer (status H), to be flagged wherever used. Part II states this prominently.

3. *No single universal functor.* A functor from all of mathematics to all of physics is unlikely to be well-defined; the formalism is deliberately local and domain-indexed (Dom-fibered), which is exactly why we work with a prestack rather than a single functor.

4. *Not every observable is a period.* Some amplitudes require elliptic, modular, or non-Tate structures; this constrains Part II.

5. *Gauge redundancy is not physical symmetry.* Gauge identifies descriptions of the same content; physical symmetry maps a state to a genuinely different one. Theorem 6.8 and Theorem 6.9 formalize the first half; Part VI treats the distinction in full.

10.2 Connections

The formalism is a bookkeeping discipline first and a mathematical object second. Its value is that it makes three things explicit that are usually left implicit: (i) *which* channel α a numerical prediction came through (Theorem 2.5); (ii) *how reliable* a chain of translations is (Theorem 6.5); and (iii) *what is lost* when one passes to a coarse, gauge-fixed description (Theorem 6.8). Each later module supplies a genuine mathematical object where Part I supplies only a slot, and the synthesis Part verifies that the slots compose.

10.3 Relation to prior frameworks

The realization-as-functor viewpoint is the Atiyah–Segal axiomatics of TQFT generalized: Theorem 4.2 is literally Atiyah's definition, and Lurie's cobordism-hypothesis program is the paradigm for "realization is a functor." The prestack/stack language follows Giraud and Vistoli; the descent criterion of Theorem 6.3 is their stackification theorem specialized to $\mathrm{Rep}_{\mathrm{phys}}$. The Curry–Howard–Lambek correspondence underwrites the Haskell/Lean bridge of Section 9. What is new here is not any single one of these ingredients but their organization into a status-labeled, domain-indexed prestack with an explicit realization pipeline, together with the status calculus of Theorem 6.5, which is original to this program.

11 Conclusion

We have built the foundational layer of a seven-part modular program. The layer consists of: the category Dom of domains and its fibered structure categories Math_d; the physical target category Phys and the bracket $[\![M]\!]_{\mathrm{phys}}$; the four definitions of the representation-stack formalism (entry, library, prestack, stack); the realization pipeline $\mathrm{Obs}_\alpha = \mathrm{Obs} \circ \mathrm{Real}_\alpha \circ \Phi$ with its four axioms; and the $\mathsf{S}/\mathsf{H}/\mathsf{P}$ status discipline promoted to a genuine calculus. We proved functoriality of the pipeline (Theorem 6.1), the descent characterization of stackhood (Theorem 6.3), the status calculus (Theorem 6.5), and the coarse-quotient obstruction (Theorem 6.8). Parts II–VI instantiate the three open slots Math_d, Real_α, and the decomposition operation; Part VI discharges the descent promissory note; the synthesis Part closes the loop.

We close, as the program does, with its four compressed slogans, each of which a later module makes precise:

mathematics is the syntax of physical representation;
motives are functional semantics;
periods are numerical measurements;
coalgebras are decomposition laws.

Part I has made the first slogan precise: the syntax is Math_d, the semantics is realized through Real_α, the measurement is Obs, and the honesty is the status label σ. The remaining three slogans are the subject of Parts II and beyond.

References

[1] M. F. Atiyah, *Topological quantum field theories*, Publications Mathématiques de l'IHÉS **68** (1988), 175–186.

[2] J. Lurie, *On the classification of topological field theories*, arXiv:0905.0465 (2009); in *Current Developments in Mathematics 2008*, Int. Press, 129–280.

[3] J. C. Baez and M. Stay, *Physics, topology, logic and computation: a Rosetta stone*, arXiv:0903.0340 (2009); *Lecture Notes in Physics* **813** (2010), 95–172.

[4] S. Abramsky and B. Coecke, *A categorical semantics of quantum protocols*, arXiv:quant-ph/0402130 (2004).

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

[6] U. Schreiber, *Differential cohomology in a cohesive ∞-topos*, arXiv:1310.7930 (2013).

[7] J. Lambek and P. J. Scott, *Introduction to Higher Order Categorical Logic*, Cambridge Studies in Advanced Mathematics **7**, Cambridge University Press, 1986.

[8] A. Vistoli, *Notes on Grothendieck topologies, fibered categories and descent theory*, arXiv:math/0412512 (2004).

[9] J. Giraud, *Cohomologie non abélienne*, Grundlehren der mathematischen Wissenschaften **179**, Springer, 1971.

[10] M. Kontsevich and D. Zagier, *Periods*, in *Mathematics Unlimited — 2001 and Beyond*, Springer, 2001, 771–808.

[11] A. B. Goncharov, *Multiple polylogarithms and mixed Tate motives*, arXiv:math/0103059 (2001).

[12] A. Connes and D. Kreimer, *Renormalization in quantum field theory and the Riemann–Hilbert problem I*, arXiv:hep-th/9912092 (1999).

[13] F. Brown, *Feynman amplitudes, coaction principle, and cosmic Galois group*, arXiv:1512.06409 (2015).

[14] R. Dijkgraaf and E. Witten, *Topological gauge theories and group cohomology*, Comm. Math. Phys. **129** (1990), 393–429.

[15] A. Yu. Kitaev, *Fault-tolerant quantum computation by anyons*, arXiv:quant-ph/9707021 (1997); Ann. Phys. **303** (2003), 2–30.