The Yoneda Constraint: A Universal Axiom for Embedded Systems

Unifying Gödel, the Measurement Problem, Compiler Bootstrap, and AI Alignment Under One Categorical Principle

Matthew Long*
The YonedaAI Collaboration
YonedaAI Research Collective
Chicago, IL

March 2026

Abstract

The Yoneda lemma, when applied to embedded subsystems, yields a universal constraint on self-knowledge: no subsystem can fully characterize the system containing it. We show that this single principle—the *Yoneda Constraint*—unifies fundamental limitation results across physics (the measurement boundary problem), logic (Gödel incompleteness), compiler theory (the bootstrap paradox), type theory (the minimal runtime axiom), epistemology (the embedded observer constraint), and artificial intelligence (the alignment problem). We formalize the constraint as a deficit in the left Kan extension of a restricted Yoneda presheaf, prove its universality via a generalization of Lawvere's fixed-point theorem, and demonstrate it empirically through the JAPL programming language's self-hosting compiler. The Yoneda Constraint is not a deficiency of any particular formalism but a structural necessity of any subsystem–supersystem relationship possessing sufficient categorical richness. We argue it constitutes a universal axiom for all informational-physical systems: *any subsystem S embedded within a total system T can characterize T only up to isomorphism of its accessible fragment—never the whole.*

Contents

*Corresponding author: `matthew@yonedaai.com`

1 Introduction — One Axiom, Many Faces

Throughout the history of mathematics, physics, and computer science, a recurring theme has emerged: systems embedded within larger systems face fundamental limitations on what they can know, prove, or compute about their containing environment. These limitations have been discovered independently, formalized in domain-specific languages, and studied as if they were distinct phenomena. We argue they are all manifestations of a single categorical principle.

1.1 The Recurring Pattern

In 1931, Gödel showed that any sufficiently powerful formal system cannot prove all true statements about itself [3]. In 1961, Lawvere recast this as a fixed-point theorem in cartesian closed categories [4]. In physics, the measurement problem—the impossibility of an observer fully characterizing the pre-geometric substrate from within emergent spacetime—has resisted resolution for a century. In compiler theory, every self-hosting compiler requires an external bootstrap. In AI, the alignment problem asks whether an agent embedded in an environment can verify its own alignment—and the answer appears to be structurally negative.

These are not coincidences. They are the *same theorem*, expressed in different categories.

1.2 The Universal Mapping

Domain	**System T**	**Subsystem S**	**What S Can't Access**	**Name**
Physics	Universe	Observer	Pre-geometric substrate	MBP
Logic	Arithmetic	Formal system	Gödel sentences	Incompleteness
Category Theory	CCC	Point-surjective map	Fixed-point-free endo.	Lawvere/SRIP
Epistemology	Reality	Embedded observer	Epistemic remainder	EOC
Compiler Theory	Full language	Self-hosted compiler	Unsupported features	Bootstrap Paradox
Type Theory	Decision space	Type system	ω-dependent decisions	MRA
AI	Environment	Agent	Unmodeled dynamics	Alignment Problem

Table 1: The Yoneda Constraint across seven domains. MBP = Measurement Boundary Problem; SRIP = Self-Reference Incompleteness Principle; EOC = Embedded Observer Constraint; MRA = Minimal Runtime Axiom.

1.3 Thesis Statement

We claim:

> *Any subsystem S embedded within a total system T can characterize T only up to isomorphism of its accessible fragment—never the whole.*

Formally, S knows T only through $\mathrm{Hom}(S, T|_S, -)$, which determines $(S, T|_S)$ up to isomorphism but not T itself. The deficit between what S can characterize and what T actually is constitutes the *Yoneda Constraint*. This paper formalizes this constraint, proves its universality, and demonstrates its manifestation across all seven domains listed in Table 1.

1.4 Overview of Contributions

1. **Formal definition** of the Yoneda Constraint as a Kan extension deficit (Section 3).
2. **Universality proof** via a generalization of Lawvere's fixed-point theorem (Section 3).
3. **Seven instantiations** showing the constraint manifests identically across physics, logic, category theory, epistemology, compiler theory, type theory, and AI (Section 4–Section 10).
4. **Empirical evidence** from the JAPL programming language's self-hosting compiler (Section 8).
5. **The impossibility of self-knowledge theorem**, a strict generalization of Gödel's incompleteness (Section 12).
6. **Structural analysis** of why the constraint is inescapable and what it implies for each domain (Section 11, Section 13).

1.5 Relation to Prior Work in the YonedaAI Program

This paper serves as the capstone of the YonedaAI research program. Papers 1–2 established the measurement boundary problem and its categorical formulation. Paper 3 proved the Self-Reference Incompleteness Principle via Lawvere's fixed-point theorem. Papers 4–7 developed the embedded observer framework, epistemic horizons, and the connection to emergent spacetime. The present work unifies all of these under a single axiom and extends it to compiler theory, type theory, and AI alignment.

2 Mathematical Foundation — The Yoneda Lemma

We begin with the mathematical bedrock upon which the entire paper rests.

2.1 Categories and Functors

Definition 2.1. A *category* $\mathcal{C}$ consists of a collection of objects $\mathrm{Ob}(\mathcal{C})$, a collection of morphisms $\mathrm{Mor}(\mathcal{C})$ with source and target maps, an identity morphism id_A for each object A, and a composition operation $\circ$ satisfying associativity and identity laws.

Definition 2.2. A *functor* $F : \mathcal{C} \to \mathcal{D}$ is a mapping that sends objects to objects and morphisms to morphisms, preserving identity and composition.

Definition 2.3. A *natural transformation* $\alpha : F \Rightarrow G$ between functors $F, G : \mathcal{C} \to \mathcal{D}$ is a family of morphisms $\alpha_A : F(A) \to G(A)$ indexed by objects $A \in \mathcal{C}$, satisfying the naturality condition: for every morphism $f : A \to B$ in $\mathcal{C}$, we have $G(f) \circ \alpha_A = \alpha_B \circ F(f)$.

2.2 The Yoneda Lemma

Theorem 2.4 (Yoneda Lemma [1])**.** *Let $\mathcal{C}$ be a locally small category, $A \in \mathrm{Ob}(\mathcal{C})$, and $F : \mathcal{C}^{\mathrm{op}} \to \mathbf{Set}$ a presheaf. Then there is a bijection, natural in both A and F:*

$$\mathrm{Nat}(\mathrm{Hom}_{\mathcal{C}}(-, A), F) \cong F(A).$$

Proof. Given a natural transformation $\alpha : \mathrm{Hom}_{\mathcal{C}}(-, A) \Rightarrow F$, we extract the element $\alpha_A(\mathrm{id}_A) \in F(A)$. Conversely, given $x \in F(A)$, we define $\alpha_B(f) = F(f)(x)$ for each $f : B \to A$. Naturality follows from functoriality of F. The two constructions are mutually inverse. $\square$

2.3 The Yoneda Embedding

Corollary 2.5 (Yoneda Embedding)**.** *The functor* $\mathsf{y} : \mathcal{C} \to [\mathcal{C}^{\mathrm{op}}, \mathbf{Set}]$ *defined by* $\mathsf{y}(A) = \mathrm{Hom}_{\mathcal{C}}(-, A)$ *is fully faithful.*

This means: *an object A is completely determined, up to isomorphism, by the totality of morphisms into it from all other objects.* The representable presheaf $\mathrm{Hom}(-, A)$ encodes all the information about A that is accessible through the category's morphism structure.

2.4 The Critical Insight: Restriction Loses Information

The Yoneda embedding is fully faithful on all of $\mathcal{C}$. But what happens when we restrict it?

Proposition 2.6. *Let $\mathcal{S}$ be a proper full subcategory of $\mathcal{C}$, and let $i : \mathcal{S} \hookrightarrow \mathcal{C}$ be the inclusion functor. The restricted Yoneda embedding*

$$\mathsf{y}|_{\mathcal{S}} : \mathcal{C} \to [\mathcal{S}^{\mathrm{op}}, \mathbf{Set}], \quad A \mapsto \mathrm{Hom}_{\mathcal{C}}(-, A)|_{\mathcal{S}} = \mathrm{Hom}_{\mathcal{C}}(i(-), A)$$

is ***not*** *necessarily fully faithful.*

Proof. Consider a morphism $f : T_1 \to T_2$ in $\mathcal{C}$ where $T_1, T_2 \notin \mathrm{Ob}(\mathcal{S})$. If no object of $\mathcal{S}$ admits a morphism to T_1, then $\mathrm{Hom}(i(S), T_1) = \emptyset$ for all $S \in \mathcal{S}$, and $\mathsf{y}|_{\mathcal{S}}(T_1) = \mathsf{y}|_{\mathcal{S}}(T_2) = \emptyset$ regardless of the structure of T_1 and T_2. More generally, $\mathsf{y}|_{\mathcal{S}}$ identifies objects that are indistinguishable from the perspective of $\mathcal{S}$. □

This is the mathematical seed of the Yoneda Constraint: *restriction of the Yoneda embedding to a subcategory loses information about the ambient category.*

2.5 Presheaves and Representability

Definition 2.7. A presheaf $P : \mathcal{C}^{\mathrm{op}} \to \mathbf{Set}$ is *representable* if $P \cong \mathrm{Hom}_{\mathcal{C}}(-, A)$ for some object A. The Yoneda lemma tells us that representable presheaves correspond bijectively to objects of $\mathcal{C}$.

The presheaf category $[\mathcal{C}^{\mathrm{op}}, \mathbf{Set}]$ is the "free cocompletion" of $\mathcal{C}$—it contains all the "ideal" objects that $\mathcal{C}$-morphisms could point to. The image of the Yoneda embedding identifies the "real" objects within this larger universe. The gap between the restricted image and the full presheaf category is where the Yoneda Constraint lives.

3 The Yoneda Constraint — Formal Statement

We now state and prove the central result of this paper.

3.1 Setup and Definitions

Definition 3.1 (Embedded Subsystem)**.** An *embedded subsystem* is a pair $(\mathcal{S}, i)$ where $\mathcal{S}$ is a category and $i : \mathcal{S} \hookrightarrow \mathcal{C}$ is a fully faithful functor (the "inclusion") into a larger category $\mathcal{C}$ (the "total system"). We say the embedding is *proper* if i is not essentially surjective—there exist objects of $\mathcal{C}$ not isomorphic to any object in the image of i.

Definition 3.2 (Accessible Fragment)**.** Given an embedded subsystem $(\mathcal{S}, i)$ and an object $T \in \mathrm{Ob}(\mathcal{C})$, the *accessible fragment* of T relative to $\mathcal{S}$ is the restricted presheaf:

$$T|_{\mathcal{S}} := \mathrm{Hom}_{\mathcal{C}}(i(-), T) : \mathcal{S}^{\mathrm{op}} \to \mathbf{Set}.$$

This is the information about T that $\mathcal{S}$ can "see" through its morphisms.

Definition 3.3 (Yoneda Deficit)**.** The *Yoneda deficit* of the subsystem $\mathcal{S}$ with respect to T is the cokernel (in a suitable 2-categorical or enriched sense) of the unit of the Kan extension adjunction:

$$\Delta(\mathcal{S}, T) := \mathrm{coker}(\eta_T : \mathsf{y}(T) \longrightarrow \mathrm{Lan}_i(\mathsf{y}(T)|_{\mathcal{S}}))$$

where Lan_i denotes the left Kan extension along $i : \mathcal{S} \hookrightarrow \mathcal{C}$.

Intuitively, the Yoneda deficit measures how much information about T is lost when we observe T only through the lens of $\mathcal{S}$, and then try to reconstruct T from those observations.

3.2 The Yoneda Constraint Theorem

Theorem 3.4 (The Yoneda Constraint)**.** *Let $(\mathcal{S}, i)$ be a proper embedded subsystem of a locally small category $\mathcal{C}$. If there exists an object $T \in \mathrm{Ob}(\mathcal{C}) \setminus \mathrm{Ob}(i(\mathcal{S}))$ such that T admits morphisms from objects not in $i(\mathcal{S})$ that are not determined by morphisms from $i(\mathcal{S})$, then:*

(i) The restricted presheaf $T|_{\mathcal{S}} = \mathrm{Hom}_{\mathcal{C}}(i(-), T)$ does not determine T up to isomorphism in $\mathcal{C}$.

(ii) The Yoneda deficit $\Delta(\mathcal{S}, T)$ is non-trivial.

(iii) The left Kan extension $\mathrm{Lan}_i(T|_{\mathcal{S}})$ does not recover $\mathsf{y}(T)$.

Proof. We prove each part.

(i) By hypothesis, there exist objects $X \notin i(\mathcal{S})$ and a morphism $f : X \to T$ such that f does not factor through any object of $i(\mathcal{S})$. The restricted presheaf $T|_{\mathcal{S}}$ records only morphisms $i(S) \to T$ for $S \in \mathcal{S}$, so it contains no information about f. We construct a witness: suppose $\mathcal{C}$ has sufficient colimits (or work in a suitable completion). Let T' be any object of $\mathcal{C}$ satisfying $\mathrm{Hom}(i(S), T') \cong \mathrm{Hom}(i(S), T)$ naturally in S but $\mathrm{Hom}(X, T') \not\cong \mathrm{Hom}(X, T)$. Such T' exists precisely because $X \notin i(\mathcal{S})$: the restricted presheaf does not constrain morphisms from X. For a concrete construction, when $\mathcal{C}$ admits pushouts, let $T' = T \sqcup_X (X \sqcup \{*\})$ where the pushout modifies the fiber over X while preserving fibers over $i(\mathcal{S})$. Then $T|_{\mathcal{S}} \cong T'|_{\mathcal{S}}$ but $T \not\cong T'$ in $\mathcal{C}$.

(ii) The unit $\eta_T : \mathsf{y}(T) \to \mathrm{Lan}_i(\mathsf{y}(T)|_{\mathcal{S}})$ is a natural transformation. By the universal property of the left Kan extension,

$$\mathrm{Lan}_i(\mathsf{y}(T)|_{\mathcal{S}})(X) = \mathrm{colim}_{(S, i(S) \to X)} \mathrm{Hom}_{\mathcal{C}}(i(S), T)$$

for each $X \in \mathcal{C}$. For $X \notin i(\mathcal{S})$ with morphisms to T not factoring through $i(\mathcal{S})$, the colimit does not capture all of $\mathrm{Hom}_{\mathcal{C}}(X, T)$. Thus η_T is not an isomorphism at X, making $\Delta(\mathcal{S}, T)$ non-trivial.

(iii) This follows directly from (ii): if η_T is not an isomorphism, then $\mathrm{Lan}_i(\mathsf{y}(T)|_{\mathcal{S}}) \neq \mathsf{y}(T)$. $\square$

3.3 Universality via Lawvere's Fixed-Point Theorem

The Yoneda Constraint gains its universal character through its connection to Lawvere's fixed-point theorem [4].

Theorem 3.5 (Lawvere's Fixed-Point Theorem)**.** *Let $\mathcal{C}$ be a cartesian closed category and let $e : A \to \Omega^A$ be a point-surjective morphism (i.e., for every $g : A \to \Omega$, there exists $a : 1 \to A$ with $e \circ a = \hat{g}$ where $\hat{g}$ is the transpose). Then every endomorphism $t : \Omega \to \Omega$ has a fixed point.*

Corollary 3.6 (Contrapositive — The Diagonal Obstruction)**.** *If Ω has a fixed-point-free endomorphism $t : \Omega \to \Omega$ (i.e., $t \circ x \neq x$ for all $x : 1 \to \Omega$), then no morphism $A \to \Omega^A$ is point-surjective.*

Theorem 3.7 (Yoneda Constraint via Lawvere)**.** *Let $\mathcal{C}$ be a cartesian closed category containing a subsystem $\mathcal{S}$ modeled as an object A (the "internal representation" of $\mathcal{S}$). The self-representation map*

$$\rho : A \to \Omega^A$$

(where Ω is a truth-value or state-space object with a fixed-point-free endomorphism) cannot be point-surjective. Therefore, there exist "states of affairs" $g : A \to \Omega$ that A cannot name through ρ—these constitute the Yoneda deficit.

Proof. By Theorem 3.6, the existence of a fixed-point-free endomorphism on Ω prevents ρ from being surjective. The elements $g \in \Omega^A$ not in the image of ρ are precisely the aspects of $\mathcal{C}$ that $\mathcal{S}$ (represented by A) cannot access. In the notation of Theorem 3.3, these unreachable elements constitute $\Delta(\mathcal{S}, T)$ where T is the "total state" object. □

3.4 The Constraint is Inescapable

Proposition 3.8 (Monotonicity of the Constraint)**.** *Enlarging the subsystem $\mathcal{S}$ to $\mathcal{S}'$ with $\mathcal{S} \subsetneq \mathcal{S}' \subsetneq \mathcal{C}$ reduces but does not eliminate the Yoneda deficit:*

$$\Delta(\mathcal{S}', T) \leq \Delta(\mathcal{S}, T), \quad \text{but } \Delta(\mathcal{S}', T) \neq 0 \text{ whenever } \mathcal{S}' \neq \mathcal{C}.$$

Proof. The inclusion $\mathcal{S} \hookrightarrow \mathcal{S}'$ means that $\mathcal{S}'$ has access to strictly more morphisms, so the Kan extension from $\mathcal{S}'$ is a better approximation. However, as long as $\mathcal{S}' \neq \mathcal{C}$, there remain objects and morphisms outside $\mathcal{S}'$, and the argument of Theorem 3.4 applies with $\mathcal{S}'$ in place of $\mathcal{S}$. □

Remark 3.9. The only way to eliminate the Yoneda deficit is $\mathcal{S} = \mathcal{C}$, which contradicts the assumption that $\mathcal{S}$ is a *subsystem*. The constraint is thus a structural necessity of any proper embedding, not a deficiency of any particular subsystem.

4 Instance 1: Physics — The Measurement Boundary Problem

4.1 The Physical Setup

In the YonedaAI framework (Papers 1–2), spacetime and its geometric structure are not fundamental but *emergent* from a pre-geometric substrate. The emergence is modeled by a functor:

$$\Phi : \mathcal{C}_{\text{Pre}} \longrightarrow \mathcal{C}_{\text{Emer}}$$

where $\mathcal{C}_{\text{Pre}}$ is the category of pre-geometric configurations and $\mathcal{C}_{\text{Emer}}$ is the category of emergent spacetime structures.

An observer $\mathcal{O}$ is an object in $\mathcal{C}_{\text{Emer}}$. The observer's measurements are morphisms in $\mathcal{C}_{\text{Emer}}$—they probe the emergent structure but not the pre-geometric substrate directly.

4.2 The Emergence Functor is Not Faithful

Proposition 4.1. *The emergence functor Φ is neither faithful nor full in general.*

This means:

- **Not faithful**: distinct pre-geometric configurations can give rise to identical emergent structures. The observer cannot distinguish them.
- **Not full**: there exist relationships between emergent structures that do not lift to the pre-geometric level. The observer may perceive patterns that are artifacts of emergence.

4.3 MBP as Yoneda Constraint

Let $\mathcal{S} = \mathcal{C}_{\text{Emer}}$ (the observer's accessible category) and $\mathcal{C} = \mathcal{C}_{\text{Pre}}$ (the full pre-geometric category). The observer's knowledge of the universe is:

$$T|_{\mathcal{S}} = \text{Hom}_{\mathcal{C}_{\text{Pre}}}(\Phi(-), T) : \mathcal{C}_{\text{Emer}}^{\text{op}} \to \mathbf{Set}$$

for any pre-geometric state T. By Theorem 3.4, this restricted presheaf does not determine T up to isomorphism. The Yoneda deficit $\Delta(\mathcal{C}_{\text{Emer}}, T)$ is exactly the *emergence kernel*—the pre-geometric information that is structurally inaccessible to any emergent observer.

Theorem 4.2 (MBP is the Yoneda Constraint)**.** *The Measurement Boundary Problem—the structural impossibility of an emergent observer probing the pre-geometric substrate—is an instance of the Yoneda Constraint with:*

$$\begin{aligned}
\mathcal{C} &= \mathcal{C}_{\text{Pre}} && \textit{(total system)} \\
\mathcal{S} &= \mathcal{C}_{\text{Emer}} && \textit{(subsystem)} \\
i &= \Phi^{\text{op}} && \textit{(inclusion via emergence)} \\
\Delta &= \ker(\Phi) && \textit{(emergence kernel)}
\end{aligned}$$

4.4 Physical Consequences

The Yoneda Constraint applied to physics suggests several *speculative* directions, which we offer as conjectures rather than established results:

1. **Quantum mechanics** (speculative): If the Born rule and state collapse are features of the restricted presheaf rather than the pre-geometric dynamics, then different interpretations of quantum mechanics might correspond to different choices of Kan extension from the observer's data to the full category. This remains a conjecture requiring rigorous formalization.
2. **The black hole information paradox** (speculative): If the black hole interior is modeled as part of $\mathcal{C} \setminus \mathcal{S}$ for an exterior observer, the Yoneda deficit would naturally account for the apparent information loss. This analogy is suggestive but has not been made rigorous.
3. **Dark sector phenomena** (highly speculative): It is conceivable that dark energy and dark matter are gravitational manifestations of pre-geometric structure visible only through the restricted presheaf. We note this possibility without endorsing it; it requires substantial additional work to evaluate.

5 Instance 2: Logic — Gödel's Incompleteness

5.1 Arithmetic as a CCC

Following Lawvere [4] and Yanofsky [5], we model Peano arithmetic within a cartesian closed category $\mathcal{C}$ where:

- Objects are types (natural numbers, sets of numbers, functions on numbers).
- Morphisms are computable functions.
- The exponential Ω^A represents the space of predicates on A.
- $\Omega = \{0, 1\}$ is the truth-value object with the fixed-point-free endomorphism $\neg : \Omega \to \Omega$.

5.2 Gödel Numbering as Self-Representation

Gödel numbering provides a morphism $\rho : \mathbb{N} \to \Omega^{\mathbb{N}}$ that encodes formulas as numbers and evaluates them. The claim that arithmetic can "talk about itself" is precisely the claim that ρ exists. The claim that arithmetic is *complete* would require ρ to be point-surjective.

5.3 The Gödel Sentence as Yoneda Deficit

Theorem 5.1 (Gödel via Yoneda)**.** *The Gödel sentence G such that $G \leftrightarrow \neg\mathrm{Prov}(\ulcorner G \urcorner)$ is an element of the Yoneda deficit $\Delta(\mathcal{S}, T)$ where:*

$$\begin{aligned} \mathcal{C} &= \textit{the category of all arithmetical truths} \\ \mathcal{S} &= \textit{the subcategory of provable statements} \\ T &= \textit{the truth-value object } \Omega \end{aligned}$$

Proof. By Theorem 3.7, the negation $\neg : \Omega \to \Omega$ is fixed-point-free, so ρ is not point-surjective. The Gödel sentence G is precisely the predicate $g : \mathbb{N} \to \Omega$ that ρ fails to hit. In our framework:

$$G \in \mathrm{Hom}(\mathbb{N}, \Omega) \setminus \mathrm{im}(\rho) = \Delta(\mathcal{S}, \Omega).$$

The sentence G is true (it belongs to $\mathcal{C}$) but not provable (it does not belong to $\mathcal{S}$). This is the Yoneda deficit made concrete. □

5.4 First and Second Incompleteness

Corollary 5.2 (First Incompleteness as Yoneda Constraint)**.** *For any consistent formal system $\mathcal{S}$ containing arithmetic, the Yoneda deficit $\Delta(\mathcal{S}, \Omega) \neq 0$. There exist true but unprovable sentences.*

Corollary 5.3 (Second Incompleteness as Meta-Constraint)**.** *The formal system $\mathcal{S}$ cannot verify that its own Yoneda embedding is faithful—i.e., $\mathcal{S}$ cannot prove* $\mathrm{Con}(\mathcal{S})$*. This is because the faithfulness of $\mathsf{y}|_{\mathcal{S}}$ is itself a statement in $\Delta(\mathcal{S}, \Omega)$.*

6 Instance 3: The SRIP — Categorical Unification

6.1 The Self-Reference Incompleteness Principle

In Paper 3 of the YonedaAI program, we proved the Self-Reference Incompleteness Principle (SRIP): in any cartesian closed category with sufficient structure, self-referential maps cannot capture all endomorphisms.

Theorem 6.1 (SRIP, Paper 3)**.** *Let $\mathcal{C}$ be a CCC with a truth-value object Ω admitting a fixed-point-free endomorphism $t : \Omega \to \Omega$. Then for any object A and morphism $e : A \to \Omega^A$, there exists $g : A \to \Omega$ not in the image of e.*

6.2 SRIP as Yoneda Constraint

The SRIP is a special case of the Yoneda Constraint where the subsystem is the "self-referential fragment" of the category:

Theorem 6.2 (SRIP is Yoneda Constraint)**.** *The SRIP is the Yoneda Constraint applied to the diagonal embedding $\Delta_A : \mathcal{C} \to \mathcal{C}^{\mathcal{C}}$ restricted to the self-representable fragment.*

Proof. The diagonal map $\delta : A \to A^A$ (the curried identity) embeds A into its own function space. The self-representation $\rho : A \to \Omega^A$ is the composition of this embedding with evaluation. By the Yoneda Constraint, the image of ρ (the "self-representable fragment") does not exhaust Ω^A. The SRIP's "unreachable predicate" g is precisely an element of the Yoneda deficit. □

6.3 The Diagonal Argument Generalized

Every classical diagonal argument—Cantor's, Russell's, Turing's, Gödel's—is an instance of the same categorical obstruction: the map $A \to \Omega^A$ cannot be surjective when Ω has a fixed-point-free endomorphism. The Yoneda Constraint provides the unifying framework:

$$\begin{tikzcd} A \arrow[r, "\rho"] \arrow[dr, "g"'] & \Omega^A \arrow[d, "\mathrm{eval}"] \\ & \Omega \arrow[loop right, "t"] \end{tikzcd}$$

The diagonal element $g = t \circ \mathrm{eval} \circ \langle \mathrm{id}_A, \rho \rangle$ satisfies $g \neq \rho(a)$ for all $a : 1 \to A$, placing it in the Yoneda deficit.

7 Instance 4: Epistemology — The Embedded Observer

7.1 The Measurement Algebra

Papers 4–7 of the YonedaAI program developed the theory of embedded observers. An observer S embedded in reality R has access to a *measurement algebra* $\mathcal{M}(S)$—the set of all observations S can make about R.

Definition 7.1. The *measurement algebra* of an embedded observer $S \subset R$ is:

$$\mathcal{M}(S) = \{m : S \to \Omega \mid m \text{ is a valid measurement outcome}\}$$

where Ω is the space of possible measurement values.

7.2 The Epistemic Horizon

Definition 7.2. The *epistemic horizon* of S in R is the boundary of the Yoneda image:

$$\partial \mathsf{y}(S) = \{T \in \mathrm{Ob}(\mathcal{C}) \mid T|_{\mathcal{S}} \neq \emptyset \text{ but } \Delta(\mathcal{S}, T) \neq 0\}$$

These are objects partially but not fully accessible to S.

The epistemic horizon separates what the observer can know (the interior of $\mathsf{y}(\mathcal{S})$) from what is structurally inaccessible (the exterior). Objects on the horizon are partially visible—the observer can detect their "shadow" but cannot fully characterize them.

7.3 The Epistemic Remainder

Theorem 7.3 (Epistemic Remainder). *For any embedded observer $S \subsetneq R$, there exists an* epistemic remainder*:*

$$\mathcal{E}(S) = \bigcup_{T \in \mathrm{Ob}(\mathcal{C})} \Delta(\mathcal{S}, T) \neq \emptyset$$

consisting of aspects of reality that S cannot access through any measurement.

Proof. Since $S \subsetneq R$, the embedding is proper. By Theorem 3.4, for any $T \notin \mathcal{S}$ with the required morphism structure, $\Delta(\mathcal{S}, T) \neq 0$. The union over all such T is non-empty. □

7.4 Science as Internal Cartography

This framework reframes the scientific enterprise: science is not the discovery of "reality as it is" but the systematic mapping of the restricted Yoneda presheaf $\mathrm{Hom}(S, R|_S, -)$. The map is faithful to the accessible fragment, but the territory (R) always exceeds the map.

> *The scientist-observer builds an ever more detailed chart of the accessible fragment. The Yoneda Constraint guarantees the chart can never become the territory.*

The Kan extension from $\mathcal{S}$ to $\mathcal{C}$ represents the observer's *best guess* about reality beyond the epistemic horizon—this is theoretical physics, metaphysics, and speculation. The Yoneda deficit measures the irreducible gap between this best guess and actual reality.

8 Instance 5: Compiler Theory — The Bootstrap Paradox

8.1 The Problem of Self-Hosting

A *self-hosting compiler* is a compiler for a language L that is itself written in L. The fundamental question is: can such a compiler compile *all* programs in L, including itself?

Consider the JAPL programming language, developed as part of the YonedaAI research program. JAPL's self-hosting compiler C_{JAPL} is written in JAPL and compiles JAPL programs to executable form. However, the original JAPL compiler was written in TypeScript—an *external* language.

8.2 Categorical Formulation

Let $\mathcal{L}$ be the category of all JAPL programs (objects) with compilation maps (morphisms). Let $\mathcal{S} \subset \mathcal{L}$ be the subcategory of programs that the self-hosting compiler can compile (the "supported fragment").

Definition 8.1. The *compiler-accessible fragment* of $\mathcal{L}$ is:

$$\mathcal{L}|_{\mathcal{S}} = \{P \in \mathrm{Ob}(\mathcal{L}) \mid C_{\mathrm{JAPL}}(P) \text{ terminates successfully}\}$$

8.3 The Bootstrap Deficit

Theorem 8.2 (Compiler Bootstrap as Yoneda Constraint)**.** *For any self-hosting compiler C_L of a sufficiently expressive language L, the compiler-accessible fragment $\mathcal{S}$ is a proper subcategory of $\mathcal{L}$. The Yoneda deficit $\Delta(\mathcal{S}, \mathcal{L})$ is non-empty and consists of programs that use language features beyond what C_L can compile.*

In the case of JAPL, the concrete Yoneda deficit includes:

1. **Nested constructor patterns**: Pattern matching on deeply nested algebraic data types.
2. **List spread operations**: Variadic destructuring in pattern contexts.
3. **Full module system**: Circular imports and re-exports.
4. **Self-referential type inference**: Types whose inference requires solving the compiler's own type equations.

These are not accidental limitations—they are structural consequences of the compiler being a *subsystem* of the language it compiles.

8.4 The Compiler-Theoretic Gödel Sentence

Theorem 8.3 (Compiler Incompleteness)**.** *No self-hosting compiler for a Turing-complete language L can compile all programs in L using only language features that it itself can compile.*

Proof. The argument proceeds on two levels. First, concretely: the compiler C_L is itself a program in L, written using a specific subset $\mathcal{S}$ of language features. Its code-generation and semantic analysis phases can only handle patterns that were anticipated during its construction. Any language feature whose compilation requires logic not present in C_L (and not expressible through the features C_L uses) lies outside $\mathcal{S}$. Since L is Turing-complete, it can express constructs of unbounded complexity—including programs whose compilation strategy requires arbitrarily sophisticated analysis. The compiler, being a fixed finite program, handles only a bounded subset.

Second, categorically: by the Yoneda Constraint with $\mathcal{C} = \mathcal{L}$ (the full language category) and $\mathcal{S}$ the self-compilable fragment, the restricted presheaf $\mathcal{L}|_{\mathcal{S}}$ does not determine $\mathcal{L}$. Programs whose compilation requires features or analysis techniques outside $\mathcal{S}$ constitute the Yoneda deficit. The self-compilable fragment is a proper subset, and its complement includes programs whose compilation requires capabilities beyond those the compiler itself can express. $\square$

8.5 The Bootstrap Chain

The resolution is the *bootstrap chain*: an external system (the TypeScript compiler) provides the initial compilation, after which the self-hosting compiler can maintain itself within its supported fragment. Each link in the chain is a Kan extension from a smaller subcategory:

$$\mathcal{S}_0\mathcal{S}_1 \cdots \mathcal{S}_n \subsetneq \mathcal{L}$$

The chain converges to a fixed point $\mathcal{S}_\infty \subsetneq \mathcal{L}$—the maximal self-compilable fragment. The Yoneda deficit $\Delta(\mathcal{S}_\infty, \mathcal{L})$ is the irreducible "bootstrap residue" that always requires external assistance.

Remark 8.4. This parallels the situation in other domains: even C, the "lingua franca" of systems programming, was originally compiled by hand from assembly language. Every language ultimately bootstraps from something outside itself. This is the compiler-theoretic manifestation of the Yoneda Constraint.

9 Instance 6: Type Theory — The Minimal Runtime Axiom

9.1 The Decision Transfer Hierarchy

The Minimal Runtime Axiom (MRA) formalizes the observation that no type system can eliminate all runtime decisions. For a program P, define:

$$\begin{aligned} D(P) &= \text{the total set of decisions required to execute } P \\ C(P) &= \text{the subset of decisions resolved at compile time} \\ R(P) &= D(P) \setminus C(P) = \text{the runtime residual} \end{aligned}$$

The MRA states: $R(P) \neq \emptyset$ for any sufficiently complex program P.

9.2 MRA as Yoneda Constraint

Theorem 9.1 (MRA is Yoneda Constraint)**.** *The Minimal Runtime Axiom is an instance of the Yoneda Constraint with:*

$$\begin{aligned} \mathcal{C} &= \textit{the category of all decisions } D(P) \\ \mathcal{S} &= \textit{the subcategory of compile-time decisions } C(P) \\ \Delta(\mathcal{S}, T) = R(P) &= \textit{the runtime residual} \end{aligned}$$

Proof. The type system performs static analysis, which is a functor $\Phi_{\text{static}} : \mathcal{D} \to \mathcal{T}$ from the decision category to the type category. The image of Φ_{static} is $C(P)$—the compile-time knowable fragment.

For decisions involving ω-*data* (external input, user behavior, network responses), no amount of static analysis can determine the outcome. These ω-dependent decisions are morphisms from objects outside the type system's category—they are the Yoneda deficit.

The restricted presheaf $D(P)|_{C(P)}$ (what the type system knows about the full decision space) does not determine $D(P)$. The deficit $R(P) = D(P) \setminus C(P)$ is non-empty precisely because ω-data introduces objects and morphisms that the static analysis cannot access. $\square$

9.3 The Decision Transfer Hierarchy as Inclusion Functors

The MRA identifies a hierarchy of decision transfer:

$$\text{Proof time} \hookrightarrow \text{Compile time} \hookrightarrow \text{Link time} \hookrightarrow \text{Load time} \hookrightarrow \text{Runtime}$$

Each level is a subcategory of the next, with an inclusion functor. The Yoneda image grows at each stage—more decisions become accessible—but the final Yoneda deficit (truly runtime decisions depending on ω-data) remains irreducible.

Proposition 9.2. *The decision transfer hierarchy is a filtered diagram of inclusions:*

$$\mathcal{S}_{\text{proof}} \hookrightarrow \mathcal{S}_{\text{compile}} \hookrightarrow \mathcal{S}_{\text{link}} \hookrightarrow \mathcal{S}_{\text{load}} \hookrightarrow \mathcal{S}_{\text{run}} = \mathcal{C}$$

The Yoneda deficit decreases monotonically:

$$\Delta_{\text{proof}} \geq \Delta_{\text{compile}} \geq \Delta_{\text{link}} \geq \Delta_{\text{load}} \geq \Delta_{\text{run}} = 0$$

Only at the final stage (actual runtime, when all ω-data has been received) does the deficit vanish—but this is trivially $\mathcal{S} = \mathcal{C}$, confirming Theorem 3.8.

10 Instance 7: AI — The Alignment Problem

10.1 The Agent-Environment Category

An AI agent $\mathcal{A}$ is embedded in an environment $\mathcal{E}$. The agent maintains an internal *world model*:

$$W_{\mathcal{A}} = \text{Hom}(\mathcal{A}, \mathcal{E}|_{\mathcal{A}}, -) : \mathcal{A}^{\text{op}} \to \mathbf{Set}$$

This is precisely the restricted Yoneda presheaf—the agent's "view" of the environment through its sensors and actuators.

10.2 Alignment as Presheaf Agreement

Definition 10.1. An agent $\mathcal{A}$ is *aligned* with a specification σ if the agent's behavior in $\mathcal{E}$ satisfies σ for all reachable states. Formally, alignment requires:

$$\forall T \in \mathcal{E},\ \forall f \in \text{Hom}(\mathcal{A}, T) : \sigma(f) = \text{true}$$

10.3 The Alignment Deficit

Theorem 10.2 (Alignment Problem as Yoneda Constraint)**.** *For any agent $\mathcal{A} \subsetneq \mathcal{E}$, the agent cannot verify its own alignment with respect to properties that depend on the Yoneda deficit $\Delta(\mathcal{A}, \mathcal{E})$.*

Proof. Alignment verification requires the agent to check $\sigma(f)$ for all $f \in \text{Hom}(\mathcal{A}, T)$ and all relevant T. But by the Yoneda Constraint, the agent's world model $W_{\mathcal{A}}$ does not capture all of $\mathcal{E}$. Properties that depend on the behavior of $T \in \mathcal{E} \setminus \mathcal{A}$ (objects the agent cannot model) are outside the agent's verification capacity.

More specifically, verifying alignment is a self-referential task: the agent must verify properties of its *own behavior* within $\mathcal{E}$. By Theorem 6.1 (the SRIP), self-referential verification is subject to the diagonal obstruction. The agent cannot construct a complete model of its own behavior in $\mathcal{E}$ because this would require a surjective self-representation map $\rho : \mathcal{A} \to \Omega^{\mathcal{A}}$, which the Yoneda Constraint forbids. □

10.4 Implications for AI Safety

Corollary 10.3 (Structural Limits of Alignment)**.** *(i)* ***Perfect alignment is impossible*** *for any embedded agent: the Yoneda deficit guarantees unverifiable behaviors.*

(ii) ***Alignment of the accessible fragment is achievable****: the agent can verify σ on $\mathcal{E}|_{\mathcal{A}}$, its restricted presheaf.*

(iii) ***External oversight is structurally necessary****: just as the bootstrap compiler needs TypeScript, the AI agent needs an external verifier to check properties in $\Delta(\mathcal{A}, \mathcal{E})$.*

Remark 10.4. This does not counsel despair about AI alignment. Rather, it provides a precise structural characterization of what alignment can and cannot achieve, and it identifies the irreducible role of external oversight. The Yoneda Constraint tells us not that alignment is hopeless, but that it requires a *multi-level* approach: internal alignment of the accessible fragment, plus external verification of the deficit.

11 The Universal Structure

11.1 The Common Categorical Skeleton

All seven instances share the same categorical structure. We summarize:

Definition 11.1 (Yoneda Constraint Instance)**.** A *Yoneda Constraint instance* is a tuple $(\mathcal{C}, \mathcal{S}, i, T, \Delta)$ where:

(i) $\mathcal{C}$ is a locally small category (the "universe" or "total system").

(ii) $\mathcal{S}$ is a proper full subcategory of $\mathcal{C}$ (the "subsystem").

(iii) $i : \mathcal{S} \hookrightarrow \mathcal{C}$ is the inclusion functor.

(iv) $T \in \mathrm{Ob}(\mathcal{C}) \setminus \mathrm{Ob}(\mathcal{S})$ is a "target object" with non-trivial morphism structure from outside $\mathcal{S}$.

(v) $\Delta(\mathcal{S}, T) = \mathrm{coker}(\eta_T : \mathsf{y}(T) \to \mathrm{Lan}_i(\mathsf{y}(T)|_{\mathcal{S}}))$ is the Yoneda deficit, which is non-trivial.

Instance	$\mathcal{C}$	$\mathcal{S}$	i	T	Δ
Physics	$\mathcal{C}_{\mathrm{Pre}}$	$\mathcal{C}_{\mathrm{Emer}}$	Φ^{op}	Pre-geom. state	Emergence kernel
Gödel	Arith. truths	Provable stmts	Proof $\hookrightarrow$ Truth	Ω	Gödel sentence
SRIP	CCC	Self-repr. frag.	Diagonal	Ω^A	Unreachable pred.
Epistemology	Reality R	Observer S	Embedding	State of R	Epistemic remainder
Compiler	Language $\mathcal{L}$	Supported frag.	Compilation	Full program	Bootstrap residue
Type Theory	Decisions D	Compile-time C	Static analysis	ω-decision	Runtime residual
AI	Environment $\mathcal{E}$	Agent $\mathcal{A}$	Sensors/actuators	Env. state	Alignment deficit

Table 2: The universal structure instantiated across all seven domains.

11.2 Structural, Not Contingent

The Yoneda Constraint is not an artifact of insufficient technology, knowledge, or computational power. It is a structural consequence of the subsystem–supersystem relationship itself.

Theorem 11.2 (Structural Necessity)**.** *For any Yoneda Constraint instance* $(\mathcal{C}, \mathcal{S}, i, T, \Delta)$*:*

(i) The constraint cannot be overcome by enlarging $\mathcal{S}$ *(short of* $\mathcal{S} = \mathcal{C}$*).*

(ii) The constraint cannot be overcome by improving $\mathcal{S}$*'s internal structure (e.g., adding axioms, upgrading sensors, expanding the compiler).*

(iii) The only "escape" is $\mathcal{S} = \mathcal{C}$*, which contradicts the definition of an embedded subsystem.*

Proof. Part (i) follows from Theorem 3.8. Part (ii) follows because the constraint depends on $\mathcal{S}$ being a *proper* subcategory, not on the internal structure of $\mathcal{S}$. No matter how rich $\mathcal{S}$'s internal structure, it remains a subcategory, and the restricted Yoneda presheaf remains restricted. Part (iii) is by definition. □

11.3 The Constraint as Functor

We can formalize the Yoneda Constraint itself as a functor from the category of embedded subsystems to the category of "deficits":

Definition 11.3. Let **Emb** be the category whose objects are proper embeddings ($\mathcal{S} \hookrightarrow \mathcal{C}$) and whose morphisms are commuting squares of functors. The *Yoneda Constraint functor* is:

$$\Delta : \mathbf{Emb} \to \mathbf{Psh}, \quad (\mathcal{S} \hookrightarrow \mathcal{C}) \mapsto \Delta(\mathcal{S}, \mathcal{C})$$

mapping each embedding to its associated deficit presheaf.

This functor is non-trivial on all of **Emb**—it sends no proper embedding to the zero presheaf. The Yoneda Constraint is a universal property of the category of embeddings.

12 The Impossibility of Self-Knowledge

12.1 A Generalization of Gödel

Gödel's incompleteness applies to formal systems containing arithmetic. The Yoneda Constraint generalizes this to *any system with the categorical structure of a proper embedding.* This is strictly more general.

Theorem 12.1 (Impossibility of Complete Self-Knowledge)**.** *For any system* S *with sufficient internal structure (specifically,* S *admits a non-trivial self-representation in a CCC with a fixed-point-free endomorphism on a truth-value object), there exists information about* S*-in-context that* S *cannot derive about itself.*

Proof. Let S be embedded in context C (the "universe" containing S). The self-representation of S is a morphism $\rho : A \to \Omega^A$ in the ambient CCC. By the Yoneda Constraint (Theorem 3.7), ρ is not surjective. The non-image elements of ρ are properties of S-in-C that S cannot derive internally.

More precisely, S can characterize itself *as an isolated object* (via introspection), but it cannot characterize *how it is embedded in* C—the Yoneda deficit includes the "view from outside" that S structurally lacks. □

12.2 The Diagonal Argument Generalized

Theorem 12.2 (Generalized Diagonal Theorem)**.** *The self-representation map $\rho : S \to S^S$ (which sends each "state" of S to the endomorphism it induces) cannot be surjective whenever S has a fixed-point-free endomorphism.*

This is the common root of:

- **Cantor**: $\mathbb{N} \to 2^{\mathbb{N}}$ is not surjective (the diagonal real).
- **Russell**: The "set of all sets not containing themselves" is not a set.
- **Turing**: The halting function is not computable (the diagonal program).
- **Gödel**: The provability predicate misses the Gödel sentence.
- **Yoneda Constraint**: The restricted presheaf misses the deficit.

All are the *same obstruction*, expressed in different categories. The Yoneda Constraint provides the categorical home for this family of results.

12.3 Connections to Consciousness and the Hard Problem

We note briefly—without claiming to solve these problems—that the Yoneda Constraint bears on discussions of consciousness and the "hard problem":

1. A conscious system S embedded in the physical world $\mathcal{C}$ can characterize its own experience only through its accessible fragment $\mathcal{C}|_S$.
2. The "explanatory gap" between neural correlates and subjective experience may be an instance of the Yoneda deficit—the gap between the self-model and the reality of being embedded.
3. Free will, understood as the system's inability to predict its own future states, is a direct consequence of the impossibility of surjective self-representation.

We do not develop these connections further here, but note them as directions for future work.

13 Implications

13.1 For Physics

The measurement problem is not solvable from within spacetime. No experiment, however clever, can probe the pre-geometric substrate directly, because the experimenter and the experiment are objects in $\mathcal{C}_{\mathrm{Emer}}$, not $\mathcal{C}_{\mathrm{Pre}}$. This does not mean the substrate is unknowable—it means it is knowable only indirectly, through the Kan extension of observable data.

The Yoneda Constraint provides a precise language for the "interpretation" question in quantum mechanics: different interpretations correspond to different choices of Kan extension, each consistent with the restricted presheaf but extending it differently to the full category.

13.2 For Mathematics

Incompleteness is a feature, not a bug. It is the Yoneda Constraint applied to formal systems. The constraint tells us that the *relationship* between a formal system and the truths it aims to capture is necessarily one of proper embedding, with an irreducible deficit.

Gödel's second incompleteness theorem gains new meaning: a system cannot verify its own Yoneda embedding is faithful. This is not because the system is "weak" but because faithfulness of the embedding is a statement *about the embedding*, which lives in the deficit.

13.3 For Language Design

Every programming language needs something outside itself to bootstrap. This is not a historical accident but a structural necessity. The Yoneda Constraint tells language designers:

1. Accept the bootstrap: design the bootstrap chain deliberately, not accidentally.

2. Characterize the deficit: know explicitly which features require external compilation.

3. Minimize but don't try to eliminate: the supported fragment can be large, but it cannot be everything.

13.4 For Type Theory

Runtime will always exist. The MRA's ω-data constitutes an irreducible Yoneda deficit that no type system—however sophisticated—can eliminate. Dependent types, linear types, and session types can all reduce the deficit but cannot zero it out.

The practical implication: language designers should not pursue "zero-runtime" type systems as an achievable goal, but rather should characterize and minimize the runtime residual.

13.5 For AI Safety

Perfect alignment is structurally impossible for embedded agents. The Yoneda deficit guarantees that any agent's world model misses aspects of its environment, and therefore the agent cannot verify its alignment with respect to those aspects.

This has concrete policy implications:

1. **External oversight is not optional**: It is structurally necessary, not merely practically useful.

2. **Alignment auditing must be multi-level**: Internal self-checks cover the accessible fragment; external audits cover the deficit.

3. **The "alignment tax" is real**: Resources spent on external verification are not waste but structural necessity.

4. **Corrigibility is essential**: Since the agent cannot verify its own alignment completely, it must remain open to external correction.

13.6 For Epistemology

The scientific enterprise is the systematic exploration of the accessible fragment $\mathrm{Hom}(S, R|_S, -)$. The Yoneda Constraint tells us:

1. Science is internally valid: the restricted presheaf faithfully represents the accessible fragment.
2. Science is externally incomplete: the accessible fragment is not all of reality.
3. Theoretical extrapolation is Kan extension: going beyond data is constructing the best approximation of reality consistent with observations.
4. The gap between theory and reality is the Yoneda deficit: irreducible but characterizable.

14 Conclusion

We have shown that the Yoneda Constraint—the principle that any proper subsystem can characterize its containing system only up to isomorphism of the accessible fragment—is a universal axiom for embedded systems. It manifests identically across seven domains:

1. **Physics**: The measurement boundary problem (observers can't probe the pre-geometric substrate).
2. **Logic**: Gödel incompleteness (formal systems can't prove all truths about themselves).
3. **Category theory**: The SRIP (self-referential maps miss fixed-point-free endomorphisms).
4. **Epistemology**: The embedded observer constraint (observers can't fully characterize reality).
5. **Compiler theory**: The bootstrap paradox (self-hosting compilers can't compile everything).
6. **Type theory**: The minimal runtime axiom (type systems can't eliminate all runtime decisions).
7. **AI**: The alignment problem (embedded agents can't verify their own alignment).

The constraint is formalized as a Kan extension deficit, proved universal via Lawvere's fixed-point theorem, and demonstrated empirically through the JAPL compiler's bootstrap chain. It is structural, not contingent—it cannot be overcome by improving the subsystem, only by dissolving the subsystem–supersystem distinction entirely.

The Yoneda Constraint is not merely a mathematical curiosity. It is a universal law of embedded systems, as fundamental to informational-physical reality as conservation laws are to physics. Any subsystem's knowledge of its containing system is bounded by its representable presheaf. The deficit between what can be known internally and what exists externally is not a failure of method but a structural necessity.

> *An embedded observer knows its world through its relationships. The Yoneda lemma guarantees these relationships determine the observer. They do not determine the world.*

References

[1] N. Yoneda, "On the homology theory of modules," *J. Fac. Sci. Univ. Tokyo*, Sect. I, vol. 7, pp. 193–227, 1954.

[2] S. Mac Lane, *Categories for the Working Mathematician*, Springer-Verlag, New York, 1971.

[3] K. Gödel, "Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I," *Monatshefte für Mathematik und Physik*, vol. 38, pp. 173–198, 1931.

[4] F. W. Lawvere, "Diagonal arguments and cartesian closed categories," in *Category Theory, Homology Theory and their Applications II*, Lecture Notes in Mathematics, vol. 92, Springer, 1969, pp. 134–145.

[5] N. S. Yanofsky, "A universal approach to self-referential paradoxes, incompleteness and fixed points," *Bull. Symbolic Logic*, vol. 9, no. 3, pp. 362–386, 2003.

[6] A. M. Turing, "On computable numbers, with an application to the Entscheidungsproblem," *Proc. London Math. Soc.*, vol. s2-42, pp. 230–265, 1937.

[7] G. Cantor, "Über eine elementare Frage der Mannigfaltigkeitslehre," *Jahresbericht der Deutschen Mathematiker-Vereinigung*, vol. 1, pp. 75–78, 1891.

[8] B. Russell, *The Principles of Mathematics*, Cambridge University Press, 1903.

[9] A. Church, "An unsolvable problem of elementary number theory," *American Journal of Mathematics*, vol. 58, pp. 345–363, 1936.

[10] J. C. Baez and M. Stay, "Physics, topology, logic and computation: A Rosetta Stone," in *New Structures for Physics*, Lecture Notes in Physics, vol. 813, Springer, 2010, pp. 95–172.

[11] S. Abramsky and B. Coecke, "A categorical semantics of quantum protocols," in *Proc. 19th Annual IEEE Symposium on Logic in Computer Science*, IEEE, 2004, pp. 415–425.

[12] B. Coecke and A. Kissinger, *Picturing Quantum Processes: A First Course in Quantum Theory and Diagrammatic Reasoning*, Cambridge University Press, 2017.

[13] C. J. Isham, "Quantum logic and the histories approach to quantum theory," *J. Math. Phys.*, vol. 35, pp. 2157–2185, 1994.

[14] A. Döring and C. J. Isham, "A topos foundation for theories of physics: I–IV," *J. Math. Phys.*, vol. 49, 2008.

[15] C. Heunen, N. P. Landsman, and B. Spitters, "A topos for algebraic quantum theory," *Commun. Math. Phys.*, vol. 291, pp. 63–110, 2009.

[16] J. Butterfield and C. J. Isham, "A topos perspective on the Kochen–Specker theorem: I. Quantum states as generalized valuations," *Int. J. Theor. Phys.*, vol. 37, pp. 2669–2733, 1998.

[17] J. S. Bell, "On the Einstein Podolsky Rosen paradox," *Physics Physique Fizika*, vol. 1, pp. 195–200, 1964.

[18] S. Kochen and E. P. Specker, "The problem of hidden variables in quantum mechanics," *J. Math. Mech.*, vol. 17, pp. 59–87, 1967.

[19] R. Penrose, *The Road to Reality: A Complete Guide to the Laws of the Universe*, Jonathan Cape, London, 2004.

[20] C. Rovelli, “Relational quantum mechanics,” *Int. J. Theor. Phys.*, vol. 35, pp. 1637–1678, 1996.

[21] L. Smolin, “The case for background independence,” in *The Structural Foundations of Quantum Gravity*, Oxford University Press, 2006.

[22] A. Connes, *Noncommutative Geometry*, Academic Press, 1994.

[23] E. Riehl, *Category Theory in Context*, Dover Publications, 2017.

[24] T. Leinster, *Basic Category Theory*, Cambridge University Press, 2014.

[25] S. Awodey, *Category Theory*, 2nd ed., Oxford University Press, 2010.

[26] F. Borceux, *Handbook of Categorical Algebra*, 3 volumes, Cambridge University Press, 1994.

[27] P. T. Johnstone, *Sketches of an Elephant: A Topos Theory Compendium*, Oxford University Press, 2002.

[28] J. Lambek and P. J. Scott, *Introduction to Higher Order Categorical Logic*, Cambridge University Press, 1986.

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

[30] P. Wadler, “Propositions as types,” *Commun. ACM*, vol. 58, no. 12, pp. 75–84, 2015.

[31] R. Milner, “A theory of type polymorphism in programming,” *J. Comput. Syst. Sci.*, vol. 17, pp. 348–375, 1978.

[32] B. C. Pierce, *Types and Programming Languages*, MIT Press, 2002.

[33] K. Thompson, “Reflections on trusting trust,” *Commun. ACM*, vol. 27, no. 8, pp. 761–763, 1984.

[34] N. Wirth, “Compiler construction,” Addison-Wesley, 1996.

[35] A. W. Appel, “Compiling with continuations,” Cambridge University Press, 2007.

[36] S. Russell, *Human Compatible: Artificial Intelligence and the Problem of Control*, Viking, 2019.

[37] N. Bostrom, *Superintelligence: Paths, Dangers, Strategies*, Oxford University Press, 2014.

[38] D. Amodei, C. Olah, J. Steinhardt, P. Christiano, J. Schulman, and D. Mané, “Concrete problems in AI safety,” *arXiv preprint arXiv:1606.06565*, 2016.

[39] P. Christiano, J. Cotra, and M. Xu, “Eliciting latent knowledge,” Technical report, Alignment Research Center, 2022.

[40] E. Hubinger, C. van Merwijk, V. Mikulik, J. Skalse, and S. Garrabrant, “Risks from learned optimization in advanced machine learning systems,” *arXiv preprint arXiv:1906.01820*, 2019.

[41] N. Soares and B. Fallenstein, “Agent foundations for aligning machine intelligence with human interests: A technical research agenda,” *Machine Intelligence Research Institute*, 2015.

[42] S. Garrabrant, T. Benson-Tilsen, A. Critch, N. Soares, and J. Taylor, "Logical induction," *arXiv preprint arXiv:1609.03543*, 2016.

[43] M. Long and YonedaAI Collaboration, "Emergent spacetime and the measurement boundary problem: A categorical approach," YonedaAI Research Collective, Papers 1–2, 2024–2025.

[44] M. Long and YonedaAI Collaboration, "The self-reference incompleteness principle via Lawvere's fixed-point theorem," YonedaAI Research Collective, Paper 3, 2025.

[45] M. Long and YonedaAI Collaboration, "The embedded observer: Epistemic horizons and the limits of measurement," YonedaAI Research Collective, Papers 4–7, 2025.

[46] M. Long and YonedaAI Collaboration, "The minimal runtime axiom: Why type systems cannot eliminate runtime," YonedaAI Research Collective, 2025.

A Kan Extensions: A Brief Review

For the reader's convenience, we recall the definition and key properties of Kan extensions.

Definition A.1. Let $F : \mathcal{S} \to \mathcal{D}$ be a functor and $i : \mathcal{S} \hookrightarrow \mathcal{C}$ an inclusion. The *left Kan extension* of F along i is a functor $\mathrm{Lan}_i F : \mathcal{C} \to \mathcal{D}$ together with a natural transformation $\eta : F \Rightarrow (\mathrm{Lan}_i F) \circ i$ that is universal: for any $G : \mathcal{C} \to \mathcal{D}$ with $\alpha : F \Rightarrow G \circ i$, there exists a unique $\bar{\alpha} : \mathrm{Lan}_i F \Rightarrow G$ with $\bar{\alpha} \circ \eta = \alpha$.

When $\mathcal{D} = \mathbf{Set}$ and colimits exist, the left Kan extension can be computed pointwise:

$$(\mathrm{Lan}_i F)(C) = \mathrm{colim}_{(S, i(S) \to C)} F(S)$$

where the colimit is taken over the comma category $(i \downarrow C)$.

The unit $\eta : F \to (\mathrm{Lan}_i F) \circ i$ is an isomorphism if and only if i is fully faithful. In our setting, i is always fully faithful (it is an inclusion of a full subcategory), so η restricted to $\mathcal{S}$ is an isomorphism. The Yoneda deficit measures the failure of η to be an isomorphism on all of $\mathcal{C}$.

B Lawvere's Fixed-Point Theorem: Full Proof

Theorem B.1 (Lawvere, 1969)**.** *In a CCC $\mathcal{C}$, if $e : A \to \Omega^A$ is point-surjective, then every $t : \Omega \to \Omega$ has a fixed point.*

Proof. Let $t : \Omega \to \Omega$ be arbitrary. Define $g = t \circ \mathrm{eval} \circ \langle \mathrm{id}_A, e \rangle : A \to \Omega$. Since e is point-surjective, there exists $a : 1 \to A$ with $e(a) = \hat{g}$ (the name of g). Then:

$$g(a) = t(\mathrm{eval}(\langle a, e(a) \rangle)) = t(\mathrm{eval}(\langle a, \hat{g} \rangle)) = t(g(a)).$$

So $g(a)$ is a fixed point of t. $\square$

Contrapositive: If Ω has a fixed-point-free endomorphism (e.g., $\neg$ on $\{0, 1\}$, or the successor on $\mathbb{N}$), then no $e : A \to \Omega^A$ is point-surjective. This is the engine of all diagonal arguments and the Yoneda Constraint.