

# Identification Capacity and Rate-Query Tradeoffs in Classification Systems

Tristan Simas

**Abstract**—We study zero-error class identification under constrained observations with three resources: tag rate  $L$  (bits per entity), identification cost  $W$  (attribute queries), and distortion  $D$  (misidentification probability). We prove an information barrier: if the attribute-profile map  $\pi$  is not injective on classes, then attribute-only observation cannot identify class identity with zero error. Let  $A_\pi := \max_u |\{c : \pi(c) = u\}|$  be collision multiplicity. Any  $D = 0$  scheme must satisfy  $L \geq \log_2 A_\pi$ , and this bound is tight. In maximal-barrier domains ( $A_\pi = k$ ), the nominal point  $(L, W, D) = (\lceil \log_2 k \rceil, O(1), 0)$  is the unique Pareto-optimal zero-error point. Without tags ( $L = 0$ ), zero-error identification requires  $W = \Omega(d)$  queries, where  $d$  is the distinguishing dimension (worst case  $d = n$ , so  $W = \Omega(n)$ ). Minimal sufficient query sets form the bases of a matroid, making  $d$  well-defined and linking the model to zero-error source coding via graph entropy. We also state fixed-axis incompleteness: a fixed observation axis is complete only for axis-measurable properties. Results instantiate to databases, biology, typed software systems, and model registries, and are machine-checked in Lean 4 (6707 lines, 296 theorem/lemma statements, 0 sorry).

**Keywords:** rate-distortion theory, identification capacity, zero-error source coding, query complexity, matroid structure, classification systems

## I. INTRODUCTION

### A. The Identification Problem

Consider an encoder-decoder pair communicating about entities from a large universe  $\mathcal{V}$ . The decoder must *identify* each entity, determining which of  $k$  classes it belongs to, using only:

- A tag of  $L$  bits stored with the entity, and/or
- Queries to a binary oracle: “does entity  $v$  satisfy attribute  $I$ ?“

This is not reconstruction (the decoder need not recover  $v$ ), but *identification* in the sense of Ahlsweide and Dueck [1]: the decoder must answer “which class?” with zero or bounded error. Our work extends this framework to consider the trade-off between tag storage, query complexity, and identification accuracy.

We prove three results:

- 1) **Information barrier (identifiability limit).** When the attribute profile  $\pi : \mathcal{V} \rightarrow \{0, 1\}^n$  is not injective on classes, zero-error identification via queries alone is impossible: any decoder produces identical output on colliding classes, so cannot be correct for both.
- 2) **Optimal tagging (achievability).** A tag of  $L = \lceil \log_2 k \rceil$  bits achieves zero-error identification with  $W = O(1)$ .

T. Simas is with McGill University, Montreal, QC, Canada (e-mail: tristan.simas@mail.mcgill.ca).

Manuscript received February 20, 2026.

© 2026 Tristan Simas. This work is licensed under CC BY 4.0. License: <https://creativecommons.org/licenses/by/4.0/>

query cost. For maximal-barrier domains ( $A_\pi = k$ ), this is the unique Pareto-optimal point in the  $(L, W, D)$  tradeoff space at  $D = 0$ ; in general domains, the converse depends on  $A_\pi := \max_u |\{c : \pi(c) = u\}|$ .

- 3) **Matroid structure (query complexity).** Minimal sufficient query sets form the bases of a matroid. The *distinguishing dimension* (the common cardinality of all minimal sets) is well-defined and lower-bounds the query cost  $W$  for any tag-free scheme.

These results are universal: the theory applies to type systems, databases, biological taxonomy, and knowledge graphs. We develop the mathematics in full generality, then exhibit concrete instantiations.

### B. The Observation Model

We formalize the observational constraint as a family of binary predicates. The terminology is deliberately abstract; concrete instantiations follow in Section VII.

**Definition I.1** (Entity space and attribute family). Let  $\mathcal{V}$  be a set of entities (program objects, database records, biological specimens, library items). Let  $\mathcal{I}$  be a finite set of binary *attributes*. Each  $I \in \mathcal{I}$  induces a bipartition of  $\mathcal{V}$  via  $q_I$ , and the full family induces the observational equivalence partition.

**Remark I.2** (Terminology). We use “attribute” for the abstract concept. In type systems, attributes are *interfaces* or *method signatures*. In databases, they are *columns*. In taxonomy, they are *phenotypic characters*. In library science, they are *facets*. The mathematics is identical.

**Definition I.3** (Attribute observation family). For each  $I \in \mathcal{I}$ , define the attribute-membership observation  $q_I : \mathcal{V} \rightarrow \{0, 1\}$ :

$$q_I(v) = \begin{cases} 1 & \text{if } v \text{ satisfies attribute } I \\ 0 & \text{otherwise} \end{cases}$$

Let  $\Phi_{\mathcal{I}} = \{q_I : I \in \mathcal{I}\}$  denote the attribute observation family.

**Remark I.4** (Notation for size parameters). We write  $n := |\mathcal{I}|$  for the ambient number of available attributes. We write  $d$  for the distinguishing dimension (the common size of all minimal distinguishing query sets; Definition IV.9), so  $d \leq n$  and there exist worst-case families with  $d = n$ . We write  $m$  for the number of *query sites* (call sites) that perform attribute checks in a program or protocol (used only in the complexity-of-maintenance discussion). When discussing a particular identification/verification task, we may write  $s$  for the number of attributes actually queried/traversed by the procedure (e.g., members/fields checked in a structural type test, phenotypic characters checked in taxonomy), with

$s \leq n$ . The maintenance-only parameter  $m$  appears only in Section III.

**Definition I.5** (Attribute profile). The attribute profile function  $\pi : \mathcal{V} \rightarrow \{0, 1\}^{|\mathcal{I}|}$  maps each value to its complete attribute signature:

$$\pi(v) = (q_I(v))_{I \in \mathcal{I}}$$

**Definition I.6** (Attribute indistinguishability). Values  $v, w \in \mathcal{V}$  are *attribute-indistinguishable*, written  $v \sim w$ , iff  $\pi(v) = \pi(w)$ .

The relation  $\sim$  is an equivalence relation. We write  $[v]_\sim$  for the equivalence class of  $v$ .

**Definition I.7** (Attribute-only observer). An *attribute-only observer* is any procedure whose interaction with a value  $v \in \mathcal{V}$  is limited to queries in  $\Phi_{\mathcal{I}}$ . Formally, the observer interacts with  $v$  only via primitive attribute queries  $q_I \in \Phi_{\mathcal{I}}$ ; hence any transcript (and output) factors through  $\pi(v)$ .

### C. The Central Question

The central question is: **what semantic properties can an attribute-only observer compute?**

A semantic property is a function  $P : \mathcal{V} \rightarrow \{0, 1\}$  (or more generally,  $P : \mathcal{V} \rightarrow Y$  for some codomain  $Y$ ). We say  $P$  is *attribute-computable* if there exists a function  $f : \{0, 1\}^{|\mathcal{I}|} \rightarrow Y$  such that  $P(v) = f(\pi(v))$  for all  $v$ .

### D. The Information Barrier

**Theorem I.8** (Information barrier). Let  $P : \mathcal{V} \rightarrow Y$  be any function. If  $P$  is attribute-computable, then  $P$  is constant on  $\sim$ -equivalence classes:

$$v \sim w \implies P(v) = P(w)$$

Equivalently: no attribute-only observer can compute any property that varies within an equivalence class.

*Proof.* Suppose  $P$  is attribute-computable via  $f$ , i.e.,  $P(v) = f(\pi(v))$  for all  $v$ . Let  $v \sim w$ , so  $\pi(v) = \pi(w)$ . Then:

$$P(v) = f(\pi(v)) = f(\pi(w)) = P(w)$$

■

**Remark I.9** (Information-theoretic nature). The barrier is *informational*, not computational. Given unlimited time, memory, and computational power, an attribute-only observer still cannot distinguish  $v$  from  $w$  when  $\pi(v) = \pi(w)$ . The constraint is on the evidence itself.

**Remark I.10** (Role in the paper). Theorem I.8 is the foundational invariance statement. The technical contribution is the downstream structure built on top of it: the ambiguity-based converse (Theorem II.30), the Pareto characterization (Theorem VI.3), and the matroid/equicardinality results (Section IV).

**Corollary I.11** (Class identity is not attribute-computable). Let  $C : \mathcal{V} \rightarrow \{1, \dots, k\}$  be the class assignment function. If there exist values  $v, w$  with  $\pi(v) = \pi(w)$  but  $C(v) \neq C(w)$ , then class identity is not attribute-computable.

*Proof.* Direct application of Theorem I.8 to  $P = C$ . ■

### E. The Positive Result: Nominal Tagging

We now show that augmenting attribute observations with a single primitive, nominal-tag access, achieves constant witness cost.

**Definition I.12** (Nominal-tag access). A *nominal tag* is a value  $\tau(v) \in \mathcal{T}$  associated with each  $v \in \mathcal{V}$ , representing the class identity of  $v$ . The *nominal-tag access* operation returns  $\tau(v)$  in  $O(1)$  time.

**Definition I.13** (Primitive query set). The extended primitive query set is  $\Phi_{\mathcal{I}}^+ = \Phi_{\mathcal{I}} \cup \{\tau\}$ , where  $\tau$  denotes nominal-tag access.

**Definition I.14** (Witness cost). Let  $W(P)$  denote the minimum number of primitive queries from  $\Phi_{\mathcal{I}}^+$  required to compute property  $P$ . We distinguish two tasks:

- $W_{\text{id}}$ : Cost to identify the class of a single entity.
- $W_{\text{eq}}$ : Cost to determine if two entities have the same class.

Unless specified,  $W$  refers to  $W_{\text{eq}}$ .

**Theorem I.15** (Constant witness for class identity). Under nominal-tag access, class identity checking has constant witness cost:

$$W(\text{class-identity}) = O(1)$$

Specifically, the witness procedure is: return  $\tau(v_1) = \tau(v_2)$ .

*Proof.* The procedure makes exactly 2 primitive queries (one  $\tau$  access per value) and one comparison. This is  $O(1)$  regardless of the number of available attributes  $|\mathcal{I}|$ . ■

**Theorem I.16** (Attribute-only lower bound). For attribute-only observers, class identity checking requires:

$$W(\text{class-identity}) = \Omega(d)$$

in the worst case, where  $d$  is the distinguishing dimension (Definition IV.9).

*Proof.* Assume a zero-error attribute-only procedure halts after fewer than  $d$  queries on every execution path. Fix any execution path and let  $Q \subseteq \mathcal{I}$  be the set of queried attributes on that path, so  $|Q| < d$ . Since  $d$  is the cardinality of every minimal distinguishing set, no set of size  $< d$  is distinguishing; hence there exist values  $v, w$  from different classes with identical answers on all attributes in  $Q$ .

An adversary can answer the procedure's queries consistently with both  $v$  and  $w$  along this path. Therefore the resulting transcript (and output) is identical on  $v$  and  $w$ , contradicting zero-error class identification. So some execution path must use at least  $d$  queries, giving worst-case cost  $\Omega(d)$ . ■

### F. Main Contributions

This paper establishes the following results:

- 1) **Information Barrier Theorem** (Theorem I.8): Attribute-only observers cannot compute any property that varies within  $\sim$ -equivalence classes. This is an information-theoretic impossibility, not a computational limitation.

- 2) **Constant-Witness Theorem** (Theorem I.15): Nominal tag access achieves  $W(\text{class-identity}) = O(1)$ , with matching lower bound  $\Omega(d)$  for attribute-only observers (Theorem I.16), where  $d$  is the distinguishing dimension (Definition IV.9).
- 3) **Complexity Separation** (Section III): We establish  $O(1)$  vs  $O(k)$  vs  $\Omega(d)$  complexity bounds for error localization under different observation regimes (where  $d$  is the distinguishing dimension).
- 4) **Matroid Structure** (Section IV): Minimal distinguishing query sets form the bases of a matroid. All such sets have equal cardinality, establishing a well-defined “distinguishing dimension.”
- 5)  **$(L, W, D)$  Optimality** (Section VI): We characterize the zero-error converse via collision multiplicity  $A_\pi$  and prove uniqueness of the nominal point in the maximal-barrier regime ( $A_\pi = k$ ).
- 6) **Machine-Checked Proofs:** All results formalized in Lean 4 (6707 lines, 296 theorem/lemma statements, 0 sorry placeholders).

#### G. Related Work and Positioning

**Identification via channels.** Our work extends the identification paradigm introduced by Ahlswede and Dueck [1], [2]. In their framework, a decoder need not reconstruct a message but only answer “is the message  $m$ ?” for a given hypothesis. This yields dramatically different capacity: double-exponential codebook sizes become achievable. Our setting differs in three ways: (1) we consider zero-error identification rather than vanishing error, (2) queries are adaptive rather than block codes, and (3) we allow auxiliary tagging (rate  $L$ ) to reduce query cost. The  $(L, W, D)$  tradeoff generalizes Ahlswede-Dueck to a multi-dimensional operating regime.

**Rate-distortion theory.** The  $(L, W, D)$  framework connects to Shannon’s rate-distortion theory [3], [4] with an important twist: the “distortion”  $D$  is semantic (class misidentification), and there is a second resource  $W$  (query cost) alongside rate  $L$ . Classical rate-distortion asks: what is the minimum rate to achieve distortion  $D$ ? We ask: given rate  $L$ , what is the minimum query cost  $W$  to achieve distortion  $D = 0$ ? Theorem VI.3 gives the converse in terms of collision multiplicity and identifies the unique nominal point in the maximal-barrier regime.

**Rate-distortion-perception tradeoffs.** Blau and Michaeli [5] extended rate-distortion theory by adding a perception constraint, creating a three-way tradeoff. Our query cost  $W$  plays an analogous role: it measures the interactive cost of achieving low distortion rather than a distributional constraint. This parallel suggests that  $(L, W, D)$  tradeoffs may admit similar geometric characterizations. Section VIII develops this connection further.

**Zero-error information theory.** The matroid structure (Section IV) connects to zero-error capacity and graph entropy. Körner [6] and Witsenhausen [7] studied zero-error source coding where confusable symbols must be distinguished. Our distinguishing dimension (Definition IV.9) is the minimum number of binary queries to separate all classes, which is precisely the zero-error identification cost when  $L = 0$ .

**Query complexity and communication complexity.** The  $\Omega(d)$  lower bound for attribute-only identification relates to decision tree complexity [8] and interactive communication [9]. The key distinction is that our queries are constrained to a fixed observable family  $\mathcal{I}$ , not arbitrary predicates.

**Compression in classification systems.** The framework applies uniformly to databases, knowledge graphs, taxonomy, and typed software systems: for zero-error identification, ambiguity induces a minimum metadata requirement  $L \geq \log_2 A_\pi$  (Theorem II.30), with maximal-barrier specialization  $L \geq \log_2 k$ .

**Programming-language corollary (secondary).** In nominal-vs-structural typing settings [10], [11], the model yields a concrete cost statement: under attribute collisions, purely structural identification has worst-case  $\Omega(d)$  witness cost, while nominal tags achieve  $O(1)$  identification using  $O(\log A_\pi)$  bits. This is the paper’s PL-facing corollary; the main contribution remains the information-theoretic characterization.

#### H. Paper Organization

Section II formalizes the compression framework and defines the  $(L, W, D)$  tradeoff. Section III establishes complexity bounds for error localization. Section IV proves the matroid structure of distinguishing query families. Section V analyzes witness cost in detail. Section VI proves the ambiguity-based converse and Pareto characterization. Section VII provides cross-domain instantiations as secondary illustrations. Section IX concludes. Appendix A describes the Lean 4 formalization.

## II. COMPRESSION FRAMEWORK

### A. Semantic Compression: The Problem

The fundamental problem of *semantic compression* is: given a value  $v$  from a large space  $\mathcal{V}$ , how can we represent  $v$  compactly while preserving the ability to answer semantic queries about  $v$ ? This differs from classical source coding in that the goal is not reconstruction but *identification*: determining which equivalence class  $v$  belongs to.

Classical rate-distortion theory [3] studies the tradeoff between representation size and reconstruction fidelity. We extend this to a discrete classification setting with three dimensions: *tag length*  $L$  (bits of storage), *witness cost*  $W$  (queries or bits of communication required to determine class membership), and *distortion*  $D$  (misclassification probability).

This work exemplifies the convergence of classical information theory with modern data systems: we extend Shannon’s rate-distortion framework to contemporary classification problems (databases, knowledge graphs, ML model registries), proving fundamental limits that were implicit in practice but not formalized in classical theory.

### B. Universe of Discourse

**Definition II.1** (Classification scheme). A *classification scheme* is any procedure (deterministic or randomized), with

arbitrary time and memory, whose only access to a value  $v \in \mathcal{V}$  is via:

- 1) The *observation family*  $\Phi = \{q_I : I \in \mathcal{I}\}$ , where  $q_I(v) = 1$  iff  $v$  satisfies attribute  $I$ ; and optionally
- 2) A *nominal-tag primitive*  $\tau : \mathcal{V} \rightarrow \mathcal{T}$  returning an opaque class identifier.

All theorems in this paper quantify over all such schemes.

This definition is intentionally broad: schemes may be adaptive, randomized, or computationally unbounded. The constraint is *observational*, not computational.

**Theorem II.2** (Information barrier). *For all classification schemes with access only to  $\Phi$  (no nominal tag), the output is constant on  $\sim_\Phi$ -equivalence classes. Therefore, no such scheme can compute any property that varies within a  $\sim_\Phi$ -class.*

*Proof.* Let  $v \sim_\Phi w$ , meaning  $q_I(v) = q_I(w)$  for all  $I \in \mathcal{I}$ . Any scheme's execution trace depends only on query responses. Since all queries return identical values for  $v$  and  $w$ , the scheme cannot distinguish them. Any output must therefore be identical. ■

**Proposition II.3** (Model capture). *Any real-world classification protocol whose evidence consists solely of attribute-membership queries is representable as a scheme in the above model. Conversely, any additional capability corresponds to adding new observations to  $\Phi$ .*

This proposition forces any objection into a precise form: to claim the theorem does not apply, one must name the additional observation capability not in  $\Phi$ . “Different universe” is not a coherent objection; it must reduce to “I have access to oracle  $X \notin \Phi$ .”

### C. Two-Axis Instantiation (Programming Languages)

The core information-theoretic results of this paper require only  $(\mathcal{V}, C, \pi)$  and the observation family  $\Phi$ . The two-axis decomposition below is an explicit programming-language instantiation used in Sections VII and A, not an additional axiom for the general theorems.

In that instantiation, each value is characterized by:

- **Lineage axis ( $B$ ):** The provenance chain of the value’s class (which classes it derives from, in what order)<sup>1</sup>
- **Profile axis ( $S$ ):** The observable attribute profile (interfaces/method signatures in the PL instantiation)

**Definition II.4** (Two-axis representation). A value  $v \in \mathcal{V}$  has representation  $(B(v), S(v))$  where:

$$B(v) = \text{lineage(class}(v)\text{)} \quad (\text{class derivation chain}) \quad (1)$$

$$S(v) = \pi(v) = (q_I(v))_{I \in \mathcal{I}} \quad (\text{attribute profile}) \quad (2)$$

The lineage axis captures *nominal* identity: where the class comes from. The profile axis captures *structural* identity: what the value can do.

<sup>1</sup>In the Lean formalization (Appendix A), the lineage axis is denoted *Bases*, reflecting its instantiation as the inheritance chain in object-oriented languages.

In the PL instantiation,  $B$  is carried by the runtime lineage order (e.g., C3/MRO output). Any implementation-specific normalization or lookup machinery is auxiliary and does not define inheritance (Appendix A).

**Theorem II.5** (Fixed-axis completeness). *Let a fixed-axis domain be specified by an axis map  $\alpha : \mathcal{V} \rightarrow \mathcal{A}$  and an observation family  $\Phi$  such that each primitive query  $q \in \Phi$  factors through  $\alpha$ . Then every in-scope semantic property (i.e., any property computable by an admissible  $\Phi$ -only strategy) factors through  $\alpha$ : there exists  $\tilde{P}$  with*

$$P(v) = \tilde{P}(\alpha(v)) \quad \text{for all } v \in \mathcal{V}.$$

In the PL instantiation,  $\alpha(v) = (B(v), S(v))$ , so in-scope semantic properties are functions of  $(B, S)$ .

*Proof.* An admissible  $\Phi$ -only strategy observes  $v$  solely through responses to primitive queries  $q_I \in \Phi$ . By hypothesis each such response is a function of  $\alpha(v)$ . Therefore every query transcript, and hence any strategy's output, depends only on  $\alpha(v)$ , so the computed property factors through  $\alpha$ . ■

**Corollary II.6** (Fixed-Axis Incompleteness). *Any fixed-axis classification system is complete only for properties measurable on the fixed axis map  $\alpha$ , and incomplete for any property that varies within an  $\alpha$ -fiber. Equivalently, if  $\alpha(v) = \alpha(w)$  but  $P(v) \neq P(w)$ , then no admissible  $\Phi$ -only strategy can compute  $P$  with zero error.*

### D. Attribute Equivalence and Observational Limits

Recall from Section 1 the attribute equivalence relation:

**Definition II.7** (Attribute equivalence (restated)). Values  $v, w \in \mathcal{V}$  are attribute-equivalent, written  $v \sim w$ , iff  $\pi(v) = \pi(w)$ , i.e., they induce exactly the same attribute responses.

**Proposition II.8** (Equivalence class structure). *The relation  $\sim$  partitions  $\mathcal{V}$  into equivalence classes. Let  $\mathcal{V}/\sim$  denote the quotient space. An attribute-only observer effectively operates on  $\mathcal{V}/\sim$ , not  $\mathcal{V}$ .*

**Corollary II.9** (Information loss quantification). *The information lost by attribute-only observation is:*

$$H(\mathcal{V}) - H(\mathcal{V}/\sim) = H(\mathcal{V}|\pi)$$

where  $H$  denotes entropy. This quantity is positive whenever multiple classes share the same attribute profile.

### E. Identification Capacity

We now formalize the identification problem in channel-theoretic terms. Let  $C : \mathcal{V} \rightarrow \{1, \dots, k\}$  denote the class assignment function, and let  $\pi : \mathcal{V} \rightarrow \{0, 1\}^n$  denote the attribute profile.

**Definition II.10** (Identification channel). *The identification channel induced by observation family  $\Phi$  is the mapping  $C \rightarrow \pi(V)$  for a random entity  $V$  drawn from distribution  $P_V$  over  $\mathcal{V}$ . The channel output is the attribute profile; the channel input is implicitly the class  $C(V)$ .*

**Theorem II.11** (Zero-Error Identification Feasibility (One Shot)). Let  $\mathcal{C} = \{1, \dots, k\}$  be the class space. The zero-error identification capacity of the observation channel is:

$$C_{id} = \begin{cases} \log_2 k & \text{if } \pi \text{ is injective on classes} \\ 0 & \text{otherwise} \end{cases}$$

That is, zero-error identification of all  $k$  classes is achievable if and only if every class has a distinct attribute profile. When  $\pi$  is not class-injective, no rate of identification is achievable with zero error.

*Proof.* Achievability: If  $\pi$  is injective on classes, then observing  $\pi(v)$  determines  $C(v)$  uniquely. The decoder simply inverts the class-to-profile mapping.

Converse (deterministic): Suppose two distinct classes  $c_1 \neq c_2$  share a profile:  $\exists v_1 \in c_1, v_2 \in c_2$  with  $\pi(v_1) = \pi(v_2)$ . Then any decoder  $g(\pi(v))$  outputs the same class label on both  $v_1$  and  $v_2$ , so it cannot be correct for both. Hence zero-error identification of all classes is impossible. ■

**Remark II.12** (Information-theoretic corollary). Under any distribution with positive mass on both colliding classes,  $I(C; \pi(V)) < H(C)$ . This is an average-case consequence of the deterministic barrier above.

**Remark II.13** (Relation to Ahlswede-Dueck). In the identification paradigm of [1], the decoder asks “is the message *m*?” rather than “what is the message?” This yields double-exponential codebook sizes. Our setting is different: we require zero-error identification of the *class*, not hypothesis testing. The one-shot zero-error identification feasibility threshold ( $\pi$  must be class-injective) is binary rather than a rate.

**Remark II.14** (Terminology). Theorem II.11 is a one-shot feasibility statement, not a Shannon asymptotic coding theorem. We retain  $C_{id}$  notation only to align with identification-theory language.

The key insight is that tagging provides a *side channel* that restores identifiability when the attribute channel fails:

**Theorem II.15** (Tag-Restored Zero Error (Sufficiency)). A class-injective nominal tag of length  $L \geq \lceil \log_2 k \rceil$  bits suffices for zero-error identification, regardless of whether  $\pi$  is class-injective.

*Proof.* A nominal tag  $\tau : \mathcal{V} \rightarrow \{1, \dots, k\}$  assigns a unique identifier to each class. Reading  $\tau(v)$  determines  $C(v)$  in  $O(1)$  queries, independent of the attribute channel. ■

#### F. Witness Cost: Query Complexity for Semantic Properties

**Definition II.16** (Witness procedure). A *witness procedure* for property  $P : \mathcal{V} \rightarrow Y$  is an algorithm  $A$  that:

- 1) Takes as input a value  $v \in \mathcal{V}$  (via query access only)
- 2) Makes queries to the primitive set  $\Phi_{\mathcal{I}}^+$
- 3) Outputs  $P(v)$

**Definition II.17** (Witness cost). The *witness cost* of property  $P$  is:

$$W(P) = \min_{A \text{ computes } P} c(A)$$

where  $c(A)$  is the worst-case number of primitive queries made by  $A$ .

**Remark II.18** (Relationship to query complexity). Witness cost is a form of query complexity [8] specialized to semantic properties. Unlike Kolmogorov complexity,  $W$  is computable and depends on the primitive set, not a universal machine.

**Lemma II.19** (Witness cost lower bounds). For any property  $P$ :

- 1) If  $P$  is attribute-computable:  $W(P) \leq |\mathcal{I}|$
- 2) If  $P$  varies within some  $\sim$ -class:  $W(P) = \infty$  for attribute-only observers
- 3) With nominal-tag access:  $W(\text{class-identity}) = O(1)$

#### G. The $(L, W, D)$ Tradeoff

We now define the three-dimensional tradeoff space that characterizes observation strategies, using information-theoretic units.

**Definition II.20** (Tag rate). For a set of class identifiers (tags)  $\mathcal{T}$  with  $|\mathcal{T}| = k$ , the *tag rate*  $L$  is the minimum number of bits required to encode a class identifier:

$$L \geq \log_2 k \text{ bits per value}$$

For nominal-tag observers,  $L = \lceil \log_2 k \rceil$  (optimal prefix-free encoding). For attribute-only observers,  $L = 0$  (no explicit tag stored). Under a distribution  $P$  over classes, the expected tag length is  $\mathbb{E}[L] \geq H(P)$  by Shannon’s source coding theorem [3].

**Definition II.21** (Witness cost (Query/Communication complexity)). The *witness cost*  $W$  is the minimum number of primitive queries (or bits of interactive communication) required for class identification:

$$W = \min_{A \text{ decides class}} c(A)$$

where  $c(A)$  is the worst-case query count. This is a form of query complexity [8] or interactive identification cost.

**Definition II.22** (Class estimator). Fix class map  $C : \mathcal{V} \rightarrow \{1, \dots, k\}$ . An observation strategy  $g$  induces an estimate

$$\hat{C}_g(v; \omega) \in \{1, \dots, k\}$$

from the available evidence (tag bits, query transcript, and optional internal randomness  $\omega$ ).

**Definition II.23** (Distortion indicator and expected distortion). For strategy  $g$ , define

$$d_g(v; \omega) := \mathbf{1}[\hat{C}_g(v; \omega) \neq C(v)].$$

Under data distribution  $P_V$  and strategy randomness, expected distortion is

$$D(g) = \Pr_{v \sim P_V, \omega} [\hat{C}_g(v; \omega) \neq C(v)].$$

The zero-error regime is  $D(g) = 0$ .

**Remark II.24** (Interpretation). In this paper,  $D$  is strictly class-misidentification probability. Additional semantic notations (e.g., hierarchical or behavior-weighted penalties) are treated as extensions in Section VIII.

## H. The $(L, W, D)$ Tradeoff Space

**Admissible schemes.** To make the Pareto-optimality claim precise, we specify the class of admissible observation strategies:

- **Deterministic or randomized:** Schemes may use randomness;  $W$  is worst-case query count.
- **Computationally unbounded:** No time/space restrictions; the constraint is observational.
- **No preprocessing over class universe:** The scheme cannot precompute a global lookup table indexed by all possible classes.
- **Tags are injective on classes:** A nominal tag  $\tau(v)$  uniquely identifies the class of  $v$ . Variable-length or compressed tags are permitted;  $L$  counts bits.
- **No amortization across queries:**  $W$  is per-identification cost, not amortized over a sequence.

**Justification.** The “no preprocessing” and “no amortization” constraints exclude trivializations:

- *Preprocessing:* With unbounded preprocessing over the class universe  $\mathcal{T}$ , one could build a lookup table mapping attribute profiles to classes. This reduces identification to  $O(1)$  table lookup, but the table has size  $O(|\mathcal{T}|)$ , hiding the complexity in space rather than eliminating it. The constraint models systems that cannot afford  $O(|\mathcal{T}|)$  storage per observer.
- *Amortization:* If  $W$  were amortized over a sequence of identifications, one could cache earlier results. This again hides complexity in state. The per-identification model captures stateless observers (typical in database queries, taxonomy lookup, and protocol/classification services).

Dropping these constraints changes the achievable region but not the qualitative separation: nominal tags still dominate for  $D = 0$  because they provide  $O(1)$  worst-case identification without requiring  $O(|\mathcal{T}|)$  preprocessing.

Under these rules, “dominance” means strict improvement on at least one of  $(L, W, D)$  with no regression on others.

**Definition II.25** (Achievable region). A point  $(L, W, D)$  is *achievable* if there exists an admissible observation strategy realizing those values. Let  $\mathcal{R} \subseteq \mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0} \times [0, 1]$  denote the achievable region.

**Definition II.26** (Pareto optimality). A point  $(L^*, W^*, D^*)$  is *Pareto-optimal* if there is no achievable  $(L, W, D)$  with  $L \leq L^*$ ,  $W \leq W^*$ ,  $D \leq D^*$ , and at least one strict inequality.

The main result of Section VI is: (i) a converse in terms of collision multiplicity  $A_\pi$ , and (ii) uniqueness of the nominal  $D = 0$  Pareto point in the maximal-barrier regime.

**Definition II.27** (Information-barrier domain). A classification domain has an *information barrier* (relative to  $\Phi$ ) if there exist distinct classes  $c_1 \neq c_2$  with identical  $\Phi$ -profiles. Equivalently,  $\pi$  is not injective on classes.

**Definition II.28** (Collision multiplicity). Let  $\mathcal{C} = \{1, \dots, k\}$  and let  $\pi_C : \mathcal{C} \rightarrow \{0, 1\}^n$  be the class-level profile map. Define

$$A_\pi := \max_{u \in \text{Im}(\pi_C)} |\{c \in \mathcal{C} : \pi_C(c) = u\}|.$$

TABLE I  
IDENTIFICATION STRATEGIES FOR 1000 CLASSES WITH 50 ATTRIBUTES.

Strategy	Tag $L$	Witness $W$
Nominal (class ID)	$\lceil \log_2 1000 \rceil = 10$ bits	$O(1)$
Duck typing (query all)	0	$\leq 50$ queries
Adaptive duck typing	0	$\geq d$ queries

Thus  $A_\pi$  is the size of the largest class-collision block under observable profiles.

**Definition II.29** (Maximal-Barrier Regime). The domain is *maximal-barrier* if  $A_\pi = k$ , i.e., all classes collide under the observation map.

### I. Converse: Tag Rate Lower Bound

**Theorem II.30** (Converse). *For any classification domain, any scheme achieving  $D = 0$  requires*

$$2^L \geq A_\pi \quad \text{equivalently} \quad L \geq \log_2 A_\pi,$$

where  $A_\pi$  is the collision multiplicity from Definition II.28.

*Proof.* Fix a collision block  $G \subseteq \mathcal{C}$  with  $|G| = A_\pi$  and identical observable profile. For classes in  $G$ , query transcripts are identical, so zero-error decoding must separate those classes using tag outcomes. With  $L$  tag bits there are at most  $2^L$  outcomes, hence  $2^L \geq |G| = A_\pi$ . ■

**Corollary II.31** (Maximal-Barrier Converse). *If the domain is maximal-barrier ( $A_\pi = k$ ), any zero-error scheme satisfies  $L \geq \log_2 k$ .*

### J. Lossy Regime: Deterministic vs Noisy Models

The zero-error corner ( $D = 0$ ) is governed by Theorem II.30. For  $D > 0$ , the model matters:

- **Deterministic queries (this section):** there is no universal law of the form  $W = O(\log(1/\epsilon) \cdot d)$ . If classes collide on all deterministic observations, repeating those same observations does not reduce error.
- **Noisy queries (Section VIII):** repeated independent observations can reduce error exponentially, yielding logarithmic-in- $1/\epsilon$  sample complexity.

Thus, in the deterministic model, distortion is controlled by collision geometry and decision rules; in noisy models, repetition-based concentration bounds become relevant.

### K. Concrete Example

Consider a classification system with  $k = 1000$  classes, each characterized by a subset of  $n = 50$  binary attributes. Table I compares the strategies.

Here  $d$  is the distinguishing dimension, the size of any minimal distinguishing query set. For typical hierarchies,  $d \approx 5-15$ . The gap between 10 bits of storage vs. 5–50 queries per identification is the cost of forgoing nominal tagging.

### III. COMPLEXITY BOUNDS

#### A. The Error Localization Theorem

a) *Scope of this section.*: This section studies *maintenance/localization* complexity, measured by locations to inspect ( $E(\cdot)$ ), not per-instance identification complexity ( $W$ ) from Sections II and V. The metrics are related but distinct:  $W$  quantifies online class-identification effort, while  $E$  quantifies where constraint logic is distributed in implementations.

**Definition III.1** (Error location count). Let  $E(\mathcal{O})$  be the number of locations that must be inspected to find all potential violations of a constraint under observation family  $\mathcal{O}$ .

**Theorem III.2** (Nominal-tag localization).  $E(\text{nominal-tag}) = O(1)$ .

*Proof.* Under nominal-tag observation, the constraint “ $v$  must be of class  $A$ ” is satisfied iff  $\tau(v) \in \text{subtypes}(A)$ . This is determined at a single location: the definition of  $\tau(v)$ ’s class. One location. ■

**Theorem III.3** (Declared-attribute localization).  $E(\text{attribute-only, declared}) = O(k)$  where  $k = \text{number of entity classes}$ .

*Proof.* With declared attribute sets (interfaces in the PL instantiation), the constraint “ $v$  must satisfy attribute  $I$ ” requires verifying that each class satisfies all attributes in  $I$ . For  $k$  classes,  $O(k)$  locations. ■

**Theorem III.4** (Attribute-only localization).  $E(\text{attribute-only}) = \Omega(m)$  where  $m = \text{number of query sites}$ .

*Proof.* Under attribute-only observation, each query site independently checks “does  $v$  have attribute  $a$ ?” with no centralized declaration. For  $m$  query sites, each must be inspected. Lower bound is  $\Omega(m)$ . ■

**Corollary III.5** (Strict dominance). *Nominal-tag observation strictly dominates attribute-only:  $E(\text{nominal-tag}) = O(1) < \Omega(m) = E(\text{attribute-only})$  for all  $m > 1$ .*

#### B. The Information Scattering Theorem

**Definition III.6** (Constraint encoding locations). Let  $I(\mathcal{O}, c)$  be the set of locations where constraint  $c$  is encoded under observation family  $\mathcal{O}$ .

**Theorem III.7** (Attribute-only scattering). *For attribute-only observation,  $|I(\text{attribute-only}, c)| = O(m)$  where  $m = \text{query sites using constraint } c$ .*

*Proof.* Each attribute query independently encodes the constraint. No shared reference exists. Constraint encodings scale with query sites. ■

**Theorem III.8** (Nominal-tag centralization). *For nominal-tag observation,  $|I(\text{nominal-tag}, c)| = O(1)$ .*

*Proof.* The constraint “must be of class  $A$ ” is encoded once in the definition of  $A$ . All tag checks reference this single definition. ■

**Corollary III.9** (Maintenance entropy). *Attribute-only observation maximizes maintenance entropy; nominal-tag observation minimizes it.*

### IV. MATROID STRUCTURE

#### A. Model Contract (Fixed-Axis Domains)

Model contract (fixed-axis domain). A domain is specified by a fixed observation family  $\Phi$  derived from a fixed axis map  $\alpha : \mathcal{V} \rightarrow \mathcal{A}$  (e.g.,  $\alpha(v) = (B(v), S(v))$ ). An observer is permitted to interact with  $v$  only through primitive queries in  $\Phi$ , and each primitive query factors through  $\alpha$ : for every  $q \in \Phi$ , there exists  $\tilde{q}$  such that  $q(v) = \tilde{q}(\alpha(v))$ . A property is in-scope semantic iff it is computable by an admissible strategy that uses only responses to queries in  $\Phi$  (under our admissibility constraints: no global preprocessing tables, no amortized caching, etc.).

We adopt  $\Phi$  as the complete observation universe for this paper: to claim applicability to a concrete runtime one must either (i) exhibit mappings from each runtime observable into  $\Phi$ , or (ii) enforce the admissibility constraints (no external registries, no reflection, no preprocessing/amortization). Under either condition the theorems apply without qualification.

**Proposition IV.1** (Observational Quotient). *For any admissible strategy using only  $\Phi$ , the entire interaction transcript (and hence the output) depends only on  $\alpha(v)$ . Equivalently, any in-scope semantic property  $P$  factors through  $\alpha$ : there exists  $\tilde{P}$  with  $P(v) = \tilde{P}(\alpha(v))$  for all  $v$ .*

**Corollary IV.2** (Why “ad hoc” = adding an axis/tag). *If two values  $v, w$  satisfy  $\alpha(v) = \alpha(w)$ , then no admissible  $\Phi$ -only strategy can distinguish them with zero error. Any mechanism that does distinguish such pairs must introduce additional information not present in  $\alpha$  (equivalently, refine the axis map by adding a new axis/tag).*

#### B. Query Families and Distinguishing Sets

The classification problem is: given a set of queries, which subsets suffice to distinguish all entities?

**Definition IV.3** (Query family). Let  $\mathcal{Q}$  be the set of all primitive queries available to an observer. For a classification system with attribute set  $\mathcal{I}$ , we have  $\mathcal{Q} = \{q_I : I \in \mathcal{I}\}$  where  $q_I(v) = 1$  iff  $v$  satisfies attribute  $I$ .

In this section, “queries” are the primitive attribute predicates  $q \in \Phi$  (equivalently, each  $q$  factors through the axis map:  $q = \tilde{q} \circ \alpha$ ). See the Convention above where  $\Phi := \mathcal{Q}$ .

**Convention:**  $\Phi := \mathcal{Q}$ . All universal quantification over “queries” ranges over  $q \in \Phi$  only.

**Definition IV.4** (Distinguishing set). A subset  $S \subseteq \mathcal{Q}$  is *distinguishing* if, for all values  $v, w$  with  $\text{class}(v) \neq \text{class}(w)$ , there exists  $q \in S$  such that  $q(v) \neq q(w)$ .

**Definition IV.5** (Minimal distinguishing set). A distinguishing set  $S$  is *minimal* if no proper subset of  $S$  is distinguishing.

### C. Matroid Structure of Query Families

**Scope and assumptions.** The matroid theorem below is unconditional within the fixed-axis observational theory defined above. In this section, “query” always means a primitive predicate  $q \in \Phi$  (equivalently,  $q$  factors through  $\alpha$  as in the Model Contract). It depends only on:

- $E = \Phi$  is the ground set of primitive queries (attribute predicates).
- “Distinguishing”: for all values  $v, w$  with  $\text{class}(v) \neq \text{class}(w)$ , there exists  $q \in S$  such that  $q(v) \neq q(w)$  (Def. above).
- “Minimal” means inclusion-minimal: no proper subset suffices.

No further assumptions are required within this theory (i.e., beyond the fixed observation family  $\Phi$  already specified). In Lean, the mechanization is now explicit end-to-end: closure-operator lemmas (extensive/monotone/idempotent) are proved in `proofs/abstract_class_system.lean` (`AxisClosure`); exchange/equicardinality lemmas are proved in `proofs/axis_framework.lean` (`orthogonal_implies_exchange`, `matroid_basis_equicardinality`); and the composition point is formalized by `closureInducedAxisMatroid` and `closureInducedBasisEquicardinality` (with closure-induced nodup/subset/exchange hypotheses made explicit).

**Definition IV.6** (Bases family). Let  $E = \Phi (= \mathcal{Q})$  be the ground set of primitive queries (attribute predicates). Let  $\mathcal{B} \subseteq 2^E$  be the family of minimal distinguishing sets.

**Lemma IV.7** (Basis exchange). *For any  $B_1, B_2 \in \mathcal{B}$  and any  $q \in B_1 \setminus B_2$ , there exists  $q' \in B_2 \setminus B_1$  such that  $(B_1 \setminus \{q\}) \cup \{q'\} \in \mathcal{B}$ .*

*Proof.* Define the closure operator  $\text{cl}(X) = \{q : X\text{-equivalence implies } q\text{-equivalence}\}$ . We verify the matroid axioms:

- 1) **Closure axioms:**  $\text{cl}$  is extensive, monotone, and idempotent. These follow directly from the definition of logical implication.
- 2) **Exchange property:** If  $q \in \text{cl}(X \cup \{q'\}) \setminus \text{cl}(X)$ , then  $q' \in \text{cl}(X \cup \{q\})$ .

For exchange, take  $q \in \text{cl}(X \cup \{q'\}) \setminus \text{cl}(X)$ . Since  $q \notin \text{cl}(X)$ , there exist  $v, w$  that are  $X$ -equivalent but disagree on  $q$ . Because  $q \in \text{cl}(X \cup \{q'\})$ , any pair that is  $(X \cup \{q'\})$ -equivalent must agree on  $q$ ; therefore this witness pair cannot be  $(X \cup \{q'\})$ -equivalent, so it must disagree on  $q'$ . Now fix any pair  $v', w'$  that are  $(X \cup \{q\})$ -equivalent. They are in particular  $X$ -equivalent and agree on  $q$ . If they disagreed on  $q'$ , then by the previous implication we could derive disagreement on  $q$ , contradiction. Hence  $v', w'$  agree on  $q'$ , proving  $q' \in \text{cl}(X \cup \{q\})$ .

Minimal distinguishing sets are exactly the bases of the matroid defined by this closure operator. The closure component is machine-checked in `AxisClosure`; the exchange/equicardinality component and the closure-to-matroid bridge used

for distinguishing-dimension well-definedness are machine-checked in `axis_framework.lean`. ■

**Theorem IV.8** (Matroid bases).  *$\mathcal{B}$  is the set of bases of a matroid on ground set  $E$ .*

*Proof.* By the basis-exchange lemma and the standard characterization of matroid bases [12]. ■

**Definition IV.9** (Distinguishing dimension). The *distinguishing dimension* of a classification system is the common cardinality of all minimal distinguishing sets.

*Remark IV.10* (Ambient attribute count vs. distinguishing dimension). Let  $n := |\mathcal{I}|$  be the ambient number of available attributes. Clearly  $d \leq n$ , and there exist worst-case families with  $d = n$ .

**Corollary IV.11** (Well-defined distinguishing dimension). *All minimal distinguishing sets have equal cardinality. Thus the distinguishing dimension (Definition IV.9) is well-defined.*

### D. Implications for Witness Cost

**Corollary IV.12** (Lower bound on attribute-only witness cost). *For any attribute-only observer,  $W(\text{class-identity}) \geq d$  where  $d$  is the distinguishing dimension.*

*Proof.* If a procedure queried fewer than  $d$  attributes on every execution path, each such queried set would be non-distinguishing by definition of  $d$ . For that path, there would exist two different classes with identical answers on all queried attributes, yielding identical transcripts and forcing the same output on both values. This contradicts zero-error class identification. Hence some path requires at least  $d$  queries. ■

The key insight: the distinguishing dimension is invariant across all minimal query strategies. The difference between nominal-tag and attribute-only observers lies in *witness cost*: a nominal tag achieves  $W = O(1)$  by storing the identity directly, bypassing query enumeration.

## V. WITNESS COST ANALYSIS

### A. Witness Cost for Class Identity

Recall from Section 2 that the witness cost  $W(P)$  is the minimum number of primitive queries required to compute property  $P$ . For class identity, we ask: what is the minimum number of queries to determine if two values have the same class?

**Theorem V.1** (Nominal-Tag Observers Achieve Minimum Witness Cost). *Nominal-tag observers achieve the minimum witness cost for class identity:*

$$W_{eq} = O(1)$$

*Specifically, the witness is a single tag read: compare  $\text{tag}(v_1) = \text{tag}(v_2)$ .*

*Attribute-only observers require  $W_{eq} = \Omega(d)$  where  $d$  is the distinguishing dimension (and  $d \leq n$ , with worst-case  $d = n$ ).*

*Proof.* See `proofs/nominal_resolution.lean`. The proof shows:

- 1) Nominal-tag access is a single primitive query
  - 2) Attribute-only observers must query at least  $d$  attributes in the worst case (a generic strategy queries all  $n$ )
  - 3) No shorter witness exists for attribute-only observers (by the information barrier)
- 

### B. Witness Cost Comparison

Observer Class	Witness Procedure	Witness Cost $W$
Nominal-tag	Single tag read	$O(1)$
Attribute-only	Query a distinguishing set	$\Omega(d)$

TABLE II

WITNESS COST FOR CLASS IDENTITY BY OBSERVER CLASS.

The Lean 4 formalization (Appendix A) provides a machine-checked proof that nominal-tag access minimizes witness cost for class identity.

## VI. $(L, W, D)$ OPTIMALITY

### A. Three-Dimensional Tradeoff: Tag Length, Witness Cost, Distortion

Recall from Section 2 that observer strategies are characterized by three dimensions:

- **Tag length**  $L$ : bits required to encode a class identifier ( $L \geq \log_2 k$  for  $k$  classes under full class tagging)
- **Witness cost**  $W$ : minimum number of primitive queries for class identification
- **Distortion**  $D$ : probability of misclassification,  $D = \Pr[\hat{C} \neq C]$ .

We compare two observer classes:

**Definition VI.1** (Attribute-only observer). An observer that queries only attribute membership ( $q_I \in \Phi_{\mathcal{I}}$ ), with no access to explicit class tags.

**Definition VI.2** (Nominal-tag observer). An observer that may read a single class identifier (nominal tag) per value, in addition to attribute queries.

**Theorem VI.3** (Pareto Optimality of Nominal-Tag Observers). Let  $A_\pi$  be the collision multiplicity (Definition II.28). Then:

- 1) Any  $D = 0$  scheme must satisfy  $L \geq \log_2 A_\pi$  (Theorem II.30).
- 2) In maximal-barrier domains ( $A_\pi = k$ ), nominal-tag observers achieve the unique Pareto-optimal  $D = 0$  point:
  - **Tag length:**  $L = \lceil \log_2 k \rceil$  bits for  $k$  classes
  - **Witness cost:**  $W = O(1)$  queries (one tag read)
  - **Distortion:**  $D = 0$  (zero misclassification probability)
- 3) In general (non-maximal) domains, nominal tagging remains Pareto-optimal at  $D = 0$  but need not be unique: partial tags can coexist on the frontier.

In information-barrier domains, attribute-only observers (the  $L = 0$  face) satisfy:

- **Tag length:**  $L = 0$  bits (no explicit tag)

- **Witness cost:**  $W = \Omega(d)$  queries (must query at least one minimal distinguishing set of size  $d$ , see Definition IV.9)
- **Distortion:**  $D > 0$  (probability of misclassification is strictly positive due to collisions)

*Proof.* Converse item (1) is Theorem II.30. For item (2), maximal barrier means all classes are observationally colliding, so any  $D = 0$  scheme must carry full class identity in tag bits (Corollary II.31), while nominal tags realize this lower bound with one tag read. Pareto uniqueness follows because any competing  $D = 0$  point cannot reduce  $L$  below  $\log_2 k$  nor reduce  $W$  below constant-time tag access under the admissibility rules of Section II.

The converse proof path is machine-checked in Lean: `proofs/lwd_converse.lean` formalizes (i) constant transcript on a collision block implies tag injectivity under zero-error decoding, and (ii) counting then yields  $2^L \geq A_\pi$ . The maximal-barrier corollary is formalized in the same module. Runtime cost instantiations (e.g., unbounded gap examples) remain in `proofs/python_instantiation.lean`. ■

*Remark VI.4* (General  $D = 0$  Frontier). When  $1 < A_\pi < k$ , the  $D = 0$  frontier can include mixed designs: a partial tag identifies collision blocks and queries resolve within blocks. This does not contradict nominal optimality; it only removes global uniqueness outside maximal-barrier domains.

- 1) Maximal barrier ( $A_\pi = k$ ): unique  $D = 0$  nominal point.
- 2) Intermediate barrier ( $1 < A_\pi < k$ ): multiple  $D = 0$  Pareto points may exist.
- 3) No barrier ( $A_\pi = 1$ ):  $L = 0$  zero-error identification is feasible.

### B. Pareto Frontier

The three-dimensional frontier shows:

- In maximal-barrier domains, the unique  $D = 0$  Pareto point is nominal tagging at  $L = \lceil \log_2 k \rceil$ .
- In general domains, attribute-only observers trade tag length for distortion on the  $L = 0$  face when collisions are present.

Figure 1 visualizes the  $(L, W, D)$  tradeoff space. The key observation is the ambiguity converse: the minimum zero-error tag rate is  $\log_2 A_\pi$ , with the maximal-barrier special case  $\log_2 k$ .

The Lean 4 formalization (Appendix A) machine-checks the full ambiguity-based converse chain and maximal-barrier lower bound that anchor this tradeoff analysis.

*Remark VI.5* (Programming language instantiations). In programming language terms: *nominal typing* corresponds to nominal-tag observers (e.g., CPython’s `isinstance`, Java’s `.getClass()`). *Duck typing* corresponds to attribute-only observers (e.g., Python’s `hasattr`). *Structural typing* is an intermediate case with  $D = 0$  but  $W = O(n)$ .

*Remark VI.6* (Structural-check cost parameter). When structural typing checks traverse  $s$  members/fields (rather than ranging over the full attribute universe), the natural bound is  $W = O(s)$  with  $s \leq n$ .

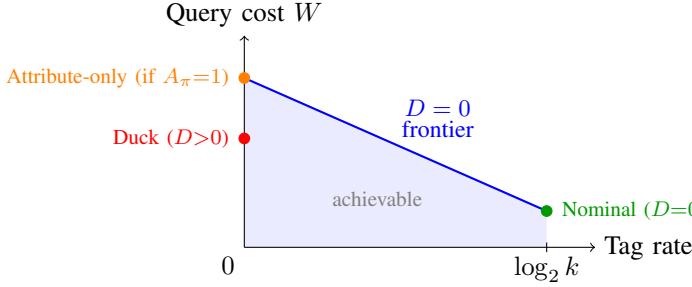


Fig. 1. Schematic illustration of the  $(L, W, D)$  tradeoff. For a concrete example with  $k = 1000$  classes, distinguishing dimension  $d = 10$ , and maximal barrier ( $A_\pi = k$ ), the nominal-tag strategy achieves  $L = 10$  bits,  $W = O(1)$ ,  $D = 0$ , while the attribute-only strategy requires  $W = 10$  queries and incurs  $D > 0$  due to collisions.

## VII. CROSS-DOMAIN INSTANTIATIONS

The preceding sections established abstract information-theoretic results (Sections II–VI). This section provides secondary cross-domain illustrations. The mathematics is unchanged by domain; these examples only instantiate the same observer-model primitives.

### A. Biological Taxonomy: Phenotype vs Genotype

Linnean taxonomy classifies organisms by observable phenotypic characters: morphology, behavior, habitat. This is attribute-only observation. The information barrier applies: phenotypically identical organisms from distinct species are indistinguishable.

**The cryptic species problem:** Cryptic species share identical phenotypic profiles but are reproductively isolated and genetically distinct. Attribute-only observation (morphology) cannot distinguish them:  $\pi(A) = \pi(B)$  but  $\text{species}(A) \neq \text{species}(B)$ .

**The nominal tag:** DNA barcoding provides the resolution [13]. A short genetic sequence (e.g., mitochondrial COI) acts as the nominal tag:  $O(1)$  identity verification via sequence comparison. This reduced cryptic species identification from  $\Omega(s)$  phenotypic examination (checking  $s$  characters) to constant-time molecular lookup.

### B. Library Classification: Subject vs ISBN

Library classification systems like Dewey Decimal observe subject matter, a form of attribute-only classification. Two books on the same subject are indistinguishable by subject code alone.

**The nominal tag:** The ISBN (International Standard Book Number) is the nominal tag [14]. Given two physical books, identity verification is  $O(1)$ : compare ISBNs. Without ISBNs, distinguishing two copies of different editions on the same subject requires  $O(s)$  attribute inspection (publication date, page count, publisher, etc.).

### C. Database Systems: Columns vs Primary Keys

In big-data systems, relational databases observe entities via column values. The information barrier applies: rows with

identical column values, excluding the key, are indistinguishable.

**The nominal tag:** The primary key is the nominal tag [15]. Entity identity is  $O(1)$ : compare keys. This is why database theory requires keys—without them, the system cannot answer “is this the same entity?”

**Natural vs surrogate keys:** Natural keys (composed of attributes) are attribute-only observation and inherit its limitations. Surrogate keys (auto-increment IDs, UUIDs) are pure nominal tags: no semantic content, pure identity.

### D. Programming-Language Snapshot (Secondary Illustration)

Programming-language runtimes are one instantiation of the same abstraction, not the source of the theory. Table III summarizes the mapping from runtime mechanisms to  $(L, W, D)$  model primitives.

Runtime	Nominal mechanism	Identity cost
Cython	<code>ob_type / type(a) is type(b)</code> [16]	$O(1)$
Java	<code>class tag via .getClass() / instanceof</code> [17], [18]	$O(1)$ to $O(d)$
TypeScript	structural compatibility only [19]	$O(s)$
Rust	<code>TypeId</code> for nominal identity [20]	$O(1)$

TABLE III  
PROGRAMMING-LANGUAGE SNAPSHOT AS A SECONDARY ILLUSTRATION  
OF THE ABSTRACT OBSERVER MODEL.

Without a class tag, identity checks are structural and scale with inspected structure size ( $O(s)$ ). With a class tag, identity is constant-time (or near-constant with bounded hierarchy traversal). This is exactly the generic witness-cost separation from Sections V and VI.

### E. Cross-Domain Summary

Domain	Attribute-Only	Nominal Tag	$W$
Biology	Phenotype (morphology)	DNA barcode (COI)	$O(1)$
Libraries	Subject (Dewey)	ISBN	$O(1)$
Databases	Column values	Primary key	$O(1)$
Cython	<code>hasattr</code> probing	<code>ob_type</code> pointer	$O(1)$
Java	<code>Attribute/interface check</code>	<code>.getClass()</code>	$O(1)$
TypeScript	Structural check	(none at runtime)	$O(s)$
Rust (static)	Trait bounds	<code>TypeId</code>	$O(1)$

TABLE IV  
WITNESS COST FOR IDENTITY ACROSS CLASSIFICATION SYSTEMS.  
NOMINAL TAGS ACHIEVE  $O(1)$ ; ATTRIBUTE-ONLY PAYS  $O(s)$  PER  
STRUCTURAL CHECK (OR  $O(k)$  WHEN ENUMERATING CLASSES UNDER  
DECLARED ATTRIBUTE CATALOGS, E.G., INTERFACES IN PL RUNTIMES).

The pattern is universal: systems with nominal tags achieve  $O(1)$  witness cost; systems without them pay  $O(s)$  or  $O(k)$ . This is not domain-specific; it is the information barrier theorem instantiated across classification systems.

### F. Machine Learning: Model Identification and Versioning

Neural network models in production systems face the identification problem: given two model instances, determine if they represent the same architecture. Model registries must compress model metadata while enabling efficient identification.

**Attribute-only approach:** Compare architecture fingerprints (layer counts, activation functions, parameter counts, connectivity patterns). Cost:  $O(s)$  where  $s$  is the number of architectural features.

**Nominal tag:** Model hash (e.g., SHA-256 of architecture definition) or registry ID. Cost:  $O(1)$ .

The  $(L, W, D)$  tradeoff applies directly: storing  $\lceil \log_2 k \rceil$  bits per model (where  $k$  is the number of distinct architectures in the registry) enables  $O(1)$  identification with  $D = 0$ . Attribute-based versioning requires  $\Omega(d)$  feature comparisons and risks false positives ( $D > 0$ ) when architectures share identical fingerprints but differ in subtle structural details.

**Example:** A model registry with  $k = 10^6$  architectures requires only 20 bits per model for perfect identification via nominal tags, versus  $O(d)$  queries over potentially hundreds of architectural features for attribute-based approaches.

### VIII. EXTENSIONS

#### A. Noisy Query Model

Throughout this paper, queries are deterministic:  $q_I(v) \in \{0, 1\}$  is a fixed function of  $v$ . In practice, observations may be corrupted. We sketch an extension to noisy queries and state the resulting open problems.

**Definition VIII.1** (Noisy observation channel). A *noisy observation channel* with crossover probability  $\epsilon \in [0, 1/2]$  returns:

$$\tilde{q}_I(v) = \begin{cases} q_I(v) & \text{with probability } 1 - \epsilon \\ 1 - q_I(v) & \text{with probability } \epsilon \end{cases}$$

Each query response is an independent BSC( $\epsilon$ ) corruption of the true value.

**Definition VIII.2** (Noisy identification capacity). The  $\epsilon$ -noisy identification capacity is the supremum rate (in bits per entity) at which zero-error identification is achievable when all attribute queries pass through a BSC( $\epsilon$ ).

In the noiseless case ( $\epsilon = 0$ ), Theorem II.11 shows the capacity is binary:  $\log_2 k$  if  $\pi$  is class-injective, 0 otherwise. For  $\epsilon > 0$ , several questions arise.

**Open problem (noisy identification cost).** For  $\epsilon > 0$  and class-injective  $\pi$ , zero-error identification is impossible with finite queries (since BSC has nonzero error probability). With bounded error  $\delta > 0$ , we expect the identification cost to scale as  $W = \Theta\left(\frac{\log(1/\delta)}{(1-2\epsilon)^2}\right)$  queries per entity. A key observation is that a nominal tag of  $L \geq \lceil \log_2 k \rceil$  bits (transmitted noiselessly) should restore  $O(1)$  identification regardless of query noise.

The third point is the key insight: *nominal tags provide a noise-free side channel*. Even when attribute observations are corrupted, a clean tag enables  $O(1)$  identification. This strengthens the case for nominal tagging in noisy environments, precisely the regime where “duck typing” would require many repeated queries to achieve confidence.

**Connection to identification via channels.** The noisy model connects more directly to Ahlsweide-Dueck identification [1]. In their framework, identification capacity over

a noisy channel can exceed Shannon capacity (double-exponential codebook sizes). Our setting differs: we have *adaptive queries* rather than block codes, and the decoder must identify a *class* rather than test a hypothesis. Characterizing the interplay between adaptive query strategies and channel noise is an open problem.

#### B. Rate-Distortion-Query Tradeoff Surface

The  $(L, W, D)$  tradeoff admits a natural geometric interpretation. In the maximal-barrier regime we identify a unique Pareto-optimal point at  $D = 0$  (Theorem VI.3); outside that regime, the full tradeoff surface can contain multiple  $D = 0$  frontier points.

**Fixed- $W$  slices.** For fixed query budget  $W$ , what is the minimum tag rate  $L$  to achieve distortion  $D$ ? When  $W \geq d$  (the distinguishing dimension), zero distortion is achievable with  $L = 0$  via exhaustive querying. When  $W < d$ , the observer cannot distinguish all classes, and either:

- Accept  $D > 0$  (misidentification), or
- Add tags ( $L > 0$ ) to compensate for insufficient queries.

**Fixed- $L$  slices.** For fixed tag rate  $L < \log_2 k$ , the tag partitions the  $k$  classes into  $2^L$  groups. Within each group, the observer must use queries to distinguish. The query cost is determined by the distinguishing dimension *within each group*, which is potentially much smaller than the global dimension.

**Open problem (subadditivity of query cost).** For a tag of rate  $L$  partitioning classes into groups  $G_1, \dots, G_{2^L}$ , we expect  $W(L) \leq \max_i d(G_i)$ , where  $d(G_i)$  is the distinguishing dimension within group  $G_i$ . Optimal tag design should minimize this maximum. Characterizing the optimal partition remains open.

#### C. Semantic Distortion Measures

We have treated distortion  $D$  as binary (correct identification or not). Richer distortion measures are possible:

- **Hierarchical distortion:** Misidentifying a class within the same genus (biological) or module (type system) is less severe than cross-genus errors.
- **Weighted distortion:** Some misidentifications have higher cost than others (e.g., type errors causing security vulnerabilities vs. benign type confusion).

#### D. Privacy and Security

**Privacy-preserving identification.** Nominal tags enable zero-knowledge proofs of class membership without revealing attribute profiles. An entity can prove “I belong to class  $C$ ” by revealing  $\tau(v) = C$  without exposing  $\pi(v)$ , preserving attribute privacy. Attribute-only schemes must reveal the complete profile  $\pi(v)$  to prove membership, leaking structural information.

**Secure model verification.** In machine learning deployment, compressed model identifiers prevent model substitution attacks. Verifying model identity via nominal tags ( $O(1)$  hash comparison) is more efficient and secure than attribute-based verification ( $O(s)$  architecture inspection), which is vulnerable to adversarial perturbations that preserve structural fingerprints while altering behavior.

### E. Connection to Rate-Distortion-Perception Theory

Blau and Michaeli [5] extended classical rate-distortion theory by adding a *perception* constraint: the reconstructed distribution must match a target distribution under some divergence measure. This creates a three-way tradeoff between rate, distortion, and perceptual quality.

Our  $(L, W, D)$  framework admits a parallel interpretation. The query cost  $W$  plays a role analogous to the perception constraint: it measures the *interactive cost* of achieving low distortion, rather than a distributional constraint. Just as rate-distortion-perception theory asks “what is the minimum rate to achieve distortion  $D$  while satisfying perception constraint  $P$ ”, we ask “what is the minimum tag rate  $L$  to achieve distortion  $D$  with query budget  $W$ ?”

The analogy suggests several directions:

- **Perception as identification fidelity:** In classification systems that must preserve statistical properties (e.g., sampling from a type distribution), a perception constraint would require the observer’s class estimates to have the correct marginal distribution, not just low expected error.
- **Three-resource tradeoffs:** The  $(L, W, D)$  Pareto frontier (Theorem VI.3) is a discrete analogue of the rate-distortion-perception tradeoff surface. Characterizing this surface for specific classification problems would extend the geometric rate-distortion program to identification settings.

Formalizing these connections would unify identification capacity with the broader rate-distortion-perception literature.

## IX. CONCLUSION

This paper presents an information-theoretic analysis of classification under observational constraints. We prove three main results:

- 1) **Information Barrier:** Observers limited to attribute-membership queries cannot compute properties that vary within indistinguishability classes. This is universal: it applies to biological taxonomy, database systems, library classification, and programming language runtimes alike.
- 2) **Witness Optimality:** Nominal-tag observers achieve  $W(\text{identity}) = O(1)$ , the minimum witness cost. The gap from attribute-only observation ( $\Omega(d)$ , with a worst-case family where  $d = n$ ) is unbounded.
- 3) **Matroid Structure:** Minimal distinguishing query sets form the bases of a matroid. The distinguishing dimension of a classification problem is well-defined and computable.

### A. The Universal Pattern

Across domains, the same structure recurs:

- **Biology:** Phenotypic observation cannot distinguish cryptic species. DNA barcoding (nominal tag) resolves them in  $O(1)$ .
- **Databases:** Column-value queries cannot distinguish rows with identical attributes. Primary keys (nominal tag) provide  $O(1)$  identity.

- **Type systems:** Attribute observation (interfaces/method signatures in this instantiation) cannot distinguish structurally identical types. Type tags provide  $O(1)$  identity.

The information barrier is not a quirk of any particular domain; it is a mathematical necessity arising from the quotient structure induced by limited observations.

### B. Implications

- **The necessity of nominal information is a theorem, not a preference.** Any zero-error scheme must satisfy the ambiguity converse  $L \geq \log_2 A_\pi$  (Theorem II.30), where  $A_\pi$  is the largest collision block induced by observable profiles. In maximal-barrier domains ( $A_\pi = k$ ), this becomes  $L \geq \log_2 k$  and nominal tagging gives the unique  $D = 0$  Pareto point with  $W = O(1)$  (Theorem VI.3).
- **The barrier is informational, not computational:** even with unbounded resources, attribute-only observers cannot overcome it.
- **Fixed-axis systems are necessarily incomplete outside their axis:** by Corollary II.6, any fixed-axis classifier is complete only for axis-measurable properties and cannot represent properties that vary within an axis fiber unless a new axis/tag is introduced.
- **Classification system design is constrained:** the choice of observation family determines which properties are computable.

### C. Future Work

- 1) **Other classification domains:** What is the matroid structure of observation spaces in chemistry (molecular fingerprints), linguistics (phonetic features), or machine learning (feature embeddings)?
- 2) **Witness complexity of other properties:** Beyond identity, what are the witness costs for provenance, equivalence, or subsumption?
- 3) **Hybrid observers:** Can observer strategies that combine tags and attributes achieve better  $(L, W, D)$  tradeoffs for specific query distributions?

### D. Conclusion

Classification under observational constraints admits a clean information-theoretic analysis. The zero-error converse is governed by collision multiplicity: any  $D = 0$  scheme necessarily has  $L \geq \log_2 A_\pi$  (Theorem II.30). In maximal-barrier domains ( $A_\pi = k$ ), nominal-tag observation achieves the unique Pareto-optimal  $D = 0$  point in the  $(L, W, D)$  tradeoff (Theorem VI.3). The results are universal within the stated observation model, and all proofs are machine-verified in Lean 4.

### AI Disclosure

This work was developed with AI assistance (Claude, Anthropic). The AI contributed to exposition, code generation, and proof exploration. All mathematical claims were verified by the authors and machine-checked in Lean 4. The Lean proofs are the authoritative source; no theorem depends solely on AI-generated reasoning.

## REFERENCES

- [1] R. Ahlswede and G. Dueck, “Identification via channels,” *IEEE Transactions on Information Theory*, vol. 35, no. 1, pp. 15–29, 1989.
- [2] ——, “Identification in the presence of feedback—a discovery of new capacity formulas,” *IEEE Transactions on Information Theory*, vol. 35, no. 1, pp. 30–36, 1989.
- [3] C. E. Shannon, “Coding theorems for a discrete source with a fidelity criterion,” *IRE National Convention Record*, vol. 7, pp. 142–163, 1959.
- [4] T. Berger, *Rate Distortion Theory: A Mathematical Basis for Data Compression*. Prentice-Hall, 1971.
- [5] Y. Blau and T. Michaeli, “Rethinking lossy compression: The rate-distortion-perception tradeoff,” *Proceedings of the 36th International Conference on Machine Learning*, pp. 675–685, 2019.
- [6] J. Körner, “Coding of an information source having ambiguous alphabet and the entropy of graphs,” *6th Prague Conference on Information Theory*, pp. 411–425, 1973.
- [7] H. S. Witsenhausen, “The zero-error side information problem and chromatic numbers,” *IEEE Transactions on Information Theory*, vol. 22, no. 5, pp. 592–593, 1976.
- [8] H. Buhrman, L. Fortnow, M. Koucký, and B. Loff, “Computational complexity of discrete problems,” *Bulletin of the EATCS*, vol. 77, pp. 42–56, 2002.
- [9] A. Orlitsky, “Worst-case interactive communication i: Two messages are almost optimal,” *IEEE Transactions on Information Theory*, vol. 37, no. 4, pp. 994–1009, 1991.
- [10] L. Cardelli and P. Wegner, “On understanding types, data abstraction, and polymorphism,” *ACM Computing Surveys*, vol. 17, no. 4, pp. 471–523, 1985.
- [11] W. R. Cook, W. Hill, and P. S. Canning, “Inheritance is not subtyping,” in *Proceedings of the 17th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, 1990, pp. 125–135.
- [12] D. J. A. Welsh, *Matroid Theory*. Academic Press, 1976.
- [13] P. D. N. Hebert, A. Cywinski, S. L. Ball, and J. R. deWaard, “Biological identifications through dna barcodes,” pp. 313–321, 2003.
- [14] International ISBN Agency, *ISBN Users’ Manual*, 7th ed. International ISBN Agency, 2017.
- [15] E. F. Codd, *The Relational Model for Database Management: Version 2*. Addison-Wesley, 1990.
- [16] P. S. Foundation, “Cpython object implementation: Object structure,” <https://github.com/python/cpython/blob/main/Include/object.h>, 2024.
- [17] Oracle, “The java virtual machine specification: Chapter 2.5 runtime data areas,” <https://docs.oracle.com/javase/specs/jvms/se21/html/jvms-2.html>, 2024.
- [18] ——, “Java object: getclass(),” <https://docs.oracle.com/javase/8/docs/api/java/lang/Object.html#getClass-->, 2024.
- [19] Microsoft, “TypeScript handbook: Type compatibility,” <https://www.typescriptlang.org/docs/handbook/type-compatibility.html>, 2024.
- [20] R. Foundation, “Rust std::any::type\_id,” [https://doc.rust-lang.org/std/any/fn.type\\_id.html](https://doc.rust-lang.org/std/any/fn.type_id.html), 2024.

## APPENDIX

### A. Formalization and Verification

The core claims in this paper are machine-checked in Lean 4. We keep the appendix concise for JSAIT and move full operational listings and implementation-level proof scripts to the supplementary artifact.

a) *What is in scope in the mechanization.*: The formalization covers the abstract observer model, the information barrier, constant-witness vs. query lower-bound separation, matroid structure of minimal distinguishing query sets, and the  $(L, W, D)$  zero-error frontier claims stated in the main text. For matroid statements, closure axioms are mechanized in `AxisClosure(abstract_class_system.lean)`; exchange/equicardinality lemmas are mechanized in `axis_framework.lean`; and the closure-to-matroid composition is formalized there via `closureInducedAxisMatroid` and `closureInducedBasisEquicardinality` (with induced nodup/subset/exchange hypotheses explicit).

For the converse in particular, `lwd_converse.lean` formalizes: `collision_block_requires_outcomes`, `collision_block_requires_bits`, `maximal_barrier_requires_bits`, and `impossible_when_bits_too_small`.

b) *What is moved to supplementary artifact.*: Implementation-specific operational details and extended code listings are included in supplementary material and are not required to follow the IT contribution in the main paper.

c) *Artifact totals.*: The complete artifact contains 14 Lean files totaling 6707 lines and 296 theorem/lemma statements; the table above highlights the core modules directly used by the main-text derivations. The remaining utility files (`Paper1.lean`, `PrintAxioms.lean`, `check_axioms.lean`, `lakefile.lean`) contribute 142 lines and 0 theorem/lemma statements.

### B. Attribute-Only Formalization

Attribute-only observation is formalized by an equivalence relation on values induced by observable query responses.

```
structure InterfaceValue where
  fields : List (String * Nat)
deriving DecidableEq

def getField (obj : InterfaceValue) (name : String) : Option Nat :=
  match obj.fields.find? (fun p => p.1 == name) with
  | some p => some p.2 | none => none

def interfaceEquivalent (a b : InterfaceValue) : Prop :=
  forall name, getField a name = getField b name

def InterfaceRespecting (f : InterfaceValue -> a) : Prop :=
  forall a b, interfaceEquivalent a b -> f a = f b
```

### C. Corollary 6.3: Provenance Impossibility

Under attribute-only observation, provenance is constant on attribute-equivalence classes; therefore provenance cannot be recovered when distinct classes collide under the observable profile.

```
theorem interface_provenance_indistinguishable
  (getProvenance : InterfaceValue -> Option DuckProvenance)
  (h_interface : InterfaceRespecting getProvenance)
  (obj1 obj2 : InterfaceValue)
  (h_equiv : interfaceEquivalent obj1 obj2) :
  getProvenance obj1 = getProvenance obj2 :=
  h_interface obj1 obj2 h_equiv
```

This is the mechanized form of the main-text impossibility statement: if an observer factors through attribute profile alone, it cannot separate equal-profile values by source/provenance.

### D. Abstract Model Lean Formalization

The abstract model is formalized directly at the axis level and then connected to concrete instantiations.

TABLE V  
LEAN 4 FORMALIZATION MODULES

Module	Lines	Theorems	Purpose
abstract_class_system.lean	3278	155	Two-axis instantiation, barrier, dominance
axis_framework.lean	1816	63	Query families, closure, matroid bridge
nominal_resolution.lean	609	27	Nominal identification and witness procedures
discipline_migration.lean	142	11	Discipline vs. migration consequences
context_formalization.lean	215	7	Greenfield/retrofit context model
python_instantiation.lean	249	17	Python instantiation
typescript_instantiation.lean	65	4	TypeScript instantiation
java_instantiation.lean	63	4	Java instantiation
rust_instantiation.lean	64	4	Rust instantiation
lwd_converse.lean	64	4	Zero-error converse chain on collision blocks
<b>Core modules subtotal</b>	<b>6565</b>	<b>296</b>	<b>10 representative modules shown</b>

```
-- Axis-indexed representation
abbrev Typ (A : Finset Axis) := (a : Axis) -> a
  \in A -> axisType a

-- Two-axis setting used in the paper
abbrev Typ2 := Typ ({Axis.Bases, Axis.Shape} :
  Finset Axis)

-- Projectors
abbrev projBases (t : Typ2) := t Axis.Bases (by
  simp)
abbrev projShape (t : Typ2) := t Axis.Shape (by
  simp)
```

The corresponding isomorphism theorem establishes that the two-axis representation is complete for in-scope observables in the formal model.

### E. Reproducibility

The full Lean development is provided in supplementary material. To verify locally:

- 1) Install Lean 4 and Lake (<https://leanprover.github.io/>).
- 2) From the release package root, run:

```
cd proofs
lake build
```

- 3) Confirm successful build with no sorry placeholders.