Identification Capacity and Rate-Query Tradeoffs in Classification Systems

Tristan Simas

***Abstract*—We extend classical rate-distortion theory to a discrete classification setting with three resources: *tag rate* L (bits of storage per entity), *identification cost* W (queries to determine class membership), and *distortion* D (misidentification probability). The fundamental question is: how do these resources trade off when an observer must identify an entity's class from limited observations?**

Information barrier (zero-error identifiability). When distinct classes share identical attribute profiles, no algorithm, regardless of computational power, can identify the class from attribute queries alone. Formally: if π is not injective on classes, then zero-error identification from attribute queries alone is impossible.

Rate-identification tradeoff. We identify the unique Pareto-optimal zero-error point in the (L, W, D) tradeoff space and describe the induced tradeoff geometry. A nominal tag of $L = \lceil \log_2 k \rceil$ bits for k classes yields $W = O(1)$ with $D = 0$. Without tags ($L = 0$), zero-error identification requires $W = \Omega(d)$ attribute queries, where d is the distinguishing dimension; in the worst case $d = n$ (the ambient attribute count), giving $W = \Omega(n)$. In the presence of attribute collisions, any tag-free scheme incurs $D > 0$.

Converse. In any information-barrier domain, any scheme achieving $D = 0$ requires $L \geq \log_2 k$ bits. This is tight: nominal tagging achieves the bound with $W = O(1)$.

Matroid structure. Minimal sufficient query sets form the bases of a matroid. The *distinguishing dimension* (the common cardinality of all minimal query sets) is well-defined, connecting to zero-error source coding via graph entropy.

Applications. The theory instantiates to databases (key vs. attribute lookup), knowledge graphs, biological taxonomy (genotype vs. phenotype), and type systems (nominal vs. structural typing). The unbounded gap $\Omega(d)$ vs. $O(1)$ (with a worst-case family where $d = n$) explains convergence toward hybrid systems combining attribute-based observation with nominal tagging.

All results are machine-checked in Lean 4 (6,000+ lines, 0 `sorry`).

In machine learning systems, the framework characterizes optimal compression of model metadata: $\lceil \log_2 k \rceil$ bits suffice for model identification and versioning, while attribute-based approaches (architecture fingerprints, hyperparameter profiles) require $\Omega(d)$ feature comparisons.

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 Ahlswede and Dueck [1]: the decoder must answer "which class?" with zero or bounded error. Our work extends this framework to consider the tradeoff between tag storage, query complexity, and identification accuracy.

We prove three results:

1) **Information barrier (identifiability limit).** When the attribute profile $\pi : \mathcal{V} \to \{0,1\}^n$ 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 $W = O(1)$ query cost. This is the unique Pareto-optimal point in the (L, W, D) tradeoff space for $D = 0$ in the information-barrier regime.
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 *attributes*: observable properties that partition the entity space.

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.

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

Manuscript received January 15, 2026.

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

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

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

Remark I.4 (Notation for size parameters). We write $n := |\mathcal{I}|$ for the ambient number of available attributes (interfaces). 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 $s \leq n$.

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

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

Definition I.6 (Interface indistinguishability)**.** Values $v, w \in \mathcal{V}$ are *interface-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 (Interface-only observer)**.** An *interface-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 interface 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 interface-only observer compute?**

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

D. The Information Barrier

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

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

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

Proof. Suppose P is interface-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 interface-only observer still cannot distinguish v from w when $\pi(v) = \pi(w)$. The constraint is on the evidence itself.

Corollary I.10 (Provenance is not interface-computable)**.** *Let type : $\mathcal{V} \to \mathcal{T}$ be the type assignment function. If there exist values v, w with $\pi(v) = \pi(w)$ but type$(v) \neq$ type(w), then type identity is not interface-computable.*

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

E. The Positive Result: Nominal Tagging

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

Definition I.11 (Nominal-tag access)**.** A *nominal tag* is a value $\tau(v) \in \mathcal{T}$ associated with each $v \in \mathcal{V}$, representing the type identity of v. The *nominal-tag access* operation returns $\tau(v)$ in $O(1)$ time.

Definition I.12 (Primitive query set)**.** The extended primitive query set is $\Phi_{\mathcal{I}}^+ = \Phi_{\mathcal{I}} \cup \{\tau\}$, where τ denotes nominal-tag access.

Definition I.13 (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_{id}: Cost to identify the class of a single entity.
- W_{eq}: Cost to determine if two entities have the same class (type identity).

Unless specified, W refers to W_{eq} (type identity checking).

Theorem I.14 (Constant witness for type identity)**.** *Under nominal-tag access, type identity checking has constant witness cost:*

$$W(\textit{type-identity}) = O(1)$$

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

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

Theorem I.15 (Interface-only lower bound)**.** *For interface-only observers, type identity checking requires:*

$$W(\textit{type-identity}) = \Omega(d)$$

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

Proof. By definition of the distinguishing dimension, any witness procedure must query at least one minimal distinguishing set of size d. Therefore the worst-case query cost is $\Omega(d)$. ■

F. Main Contributions

This paper establishes the following results:

1) **Information Barrier Theorem** (Theorem I.8): Interface-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.14): Nominal-tag access achieves $W(\text{type-identity}) = O(1)$, with matching lower bound $\Omega(d)$ for interface-only observers (Theorem I.15), 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): Nominal-tag observers achieve the unique Pareto-optimal point in the (L, W, D) tradeoff space (tag length, witness cost, distortion).
6) **Machine-Checked Proofs**: All results formalized in Lean 4 (6,086 lines, 265 theorems, 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 identifies the unique Pareto-optimal point in this three-dimensional tradeoff at $D = 0$.

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 interface-only identification relates to decision tree complexity [8] and interactive communication [9]. The key distinction is that our queries are constrained to a fixed attribute family $\mathcal{I}$, not arbitrary predicates. This constraint models practical systems where the observer's interface to entities is architecturally fixed.

Compression in classification systems. Our framework instantiates to type systems, where the compression question becomes: how many bits must be stored per object to enable $O(1)$ type identification? The answer ($\lceil \log_2 k \rceil$ bits for k classes) matches the converse bound (Theorem II.27). This provides an information-theoretic foundation for the nominal-vs-structural typing debate in programming language theory [10], [11].

Historical context. Structural classification approaches (exemplified by "duck typing" in programming languages: "if it walks like a duck and quacks like a duck, it's a duck") advocate attribute-based observation over nominal tagging. Within our model, we prove this incurs $\Omega(d)$ witness cost where tagging achieves $O(1)$ (see Definition IV.9). The result does not "resolve" the broader debate (which involves usability and tooling concerns beyond this model) but establishes that the tradeoff has a precise information-theoretic component.

Practical convergence. Modern systems have converged on hybrid classification: Python's Abstract Base Classes, TypeScript's branded types, Rust's trait system, DNA barcoding in taxonomy [12]. This convergence is consistent with the rate-query tradeoff: nominal tags provide $O(1)$ identification at cost $O(\log k)$ bits. The contribution is not advocacy for any design, but a formal framework for analyzing identification cost in classification systems.

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 type axes. Section V analyzes witness cost in detail. Section VI proves Pareto optimality. Section VII instantiates the theory in real runtimes. 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 (semantic fidelity).

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* $\tau : \mathcal{V} \to \mathcal{T}$ returning an opaque type 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 Φ (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 Φ.*

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 Φ. "Different universe" is not a coherent objection; it must reduce to "I have access to oracle $X \notin \Phi$."

C. The Two-Axis Model

We adopt a two-axis model of semantic structure, where each value is characterized by:

- **Lineage axis** (B): The provenance chain of the value's class (which classes it derives from, in what order)[1]
- **Profile axis** (S): The observable interface (which attributes/methods the value provides)

[1] In the Lean formalization (Appendix A), the lineage axis is denoted `Bases`, reflecting its instantiation as the inheritance chain in object-oriented languages.

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

$$B(v) = \text{lineage}(\text{class}(v)) \quad \text{(class derivation chain)} \tag{1}$$
$$S(v) = \pi(v) = (q_I(v))_{I \in \mathcal{I}} \quad \text{(interface profile)} \tag{2}$$

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

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 $\alpha : \mathcal{V} \to \mathcal{A}$ and an observation interface Φ such that each primitive query $q \in \Phi$ factors through α. Then every in-scope semantic property (i.e., any property computable by an admissible Φ-only strategy) factors through α: there exists $\tilde{P}$ with*

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

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

Proof. An admissible Φ-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 α. ■

D. Interface Equivalence and Observational Limits

Recall from Section 1 the interface equivalence relation:

Definition II.6 (Interface equivalence (restated))**.** Values $v, w \in \mathcal{V}$ are interface-equivalent, written $v \sim w$, iff $\pi(v) = \pi(w)$, i.e., they satisfy exactly the same interfaces.

Proposition II.7 (Equivalence class structure)**.** *The relation $\sim$ partitions $\mathcal{V}$ into equivalence classes. Let $\mathcal{V}/{\sim}$ denote the quotient space. An interface-only observer effectively operates on $\mathcal{V}/{\sim}$, not $\mathcal{V}$.*

Corollary II.8 (Information loss quantification)**.** *The information lost by interface-only observation is:*

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

where H denotes entropy. This quantity is positive whenever multiple types share the same interface profile.

E. Identification Capacity

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

Definition II.9 (Identification channel)**.** The *identification channel* induced by observation family Φ is the mapping

$C \to \pi(V)$ 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.10 (Identification capacity)**.** *Let $\mathcal{C} = \{1, \ldots, k\}$ 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 π is not class-injective, no rate of identification is achievable with zero error.

Proof. Achievability: If π 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.11 (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.12 (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 (π must be class-injective) is binary rather than a rate.

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

Theorem II.13 (Tag-restored capacity)**.** *A tag of length $L \geq \lceil \log_2 k \rceil$ bits restores zero-error identification capacity, regardless of whether π is class-injective. The tag provides a noiseless side channel with capacity L bits.*

Proof. A nominal tag $\tau : \mathcal{V} \to \{1, \ldots, k\}$ 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.14 (Witness procedure)**.** A *witness procedure* for property $P : \mathcal{V} \to Y$ 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.15 (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.16 (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.17 (Witness cost lower bounds)**.** *For any property P:*

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

G. The (L, W, D) Tradeoff

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

Definition II.18 (Tag rate)**.** For a set of type identifiers (tags) $\mathcal{T}$ with $|\mathcal{T}| = k$, the *tag rate* L is the minimum number of bits required to encode a type identifier:

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

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

Definition II.19 (Witness cost (Query/Communication complexity))**.** The *witness cost* W is the minimum number of primitive queries (or bits of interactive communication) required for type identity checking:

$$W = \min_{A \text{ decides type-identity}} c(A)$$

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

Definition II.20 (Behavioral equivalence)**.** Let behavior : $\mathcal{V} \to \mathcal{B}$ be a function mapping each entity to its observable behavior (the set of responses to all possible operations). Two entities v, w are *behaviorally equivalent*, written $v \equiv w$, iff behavior(v) = behavior(w).

In type systems, behavior(v) is the denotational semantics: the function computed by v. In databases, it is the set of query results. In taxonomy, it is the phenotype. The formalism is parametric in this choice.

Definition II.21 (Distortion function)**.** Let $d : \mathcal{V} \times \mathcal{V} \to \{0, 1\}$ be the misclassification indicator:

$$d(v, \hat{v}) = \begin{cases} 0 & \text{if type}(v) = \text{type}(\hat{v}) \Rightarrow v \equiv \hat{v} \\ 1 & \text{otherwise (semantic error)} \end{cases}$$

That is, $d = 0$ when type equality implies behavioral equivalence (soundness), and $d = 1$ when the observer conflates behaviorally distinct entities.

Definition II.22 (Expected and worst-case distortion)**.** Given a distribution P over values, the *expected distortion* is:

$$D = \mathbb{E}_{v \sim P}[d(v, \hat{v})]$$

The *zero-error regime* requires $D = 0$ (no semantic errors for any v). All theorems in this paper are proved in the zero-error regime, the strongest case where the separation is sharpest.

Remark II.23 (Distortion interpretation). $D = 0$ means the observation strategy is *sound*: type equality (as computed by the observer) implies behavioral equivalence. $D > 0$ means the strategy may conflate behaviorally distinct values with positive probability. We use D as expected misclassification rate, and in the core theorems we focus on the zero-error regime $D = 0$ vs $D > 0$; richer distortion structures are discussed 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 type universe**: The scheme cannot precompute a global lookup table indexed by all possible types.
- **Tags are injective on classes**: A nominal tag $\tau(v)$ uniquely identifies the type 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 type universe $\mathcal{T}$, one could build a lookup table mapping attribute profiles to types. 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 type checking, database queries, and biological identification).

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.24 (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.25 (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.

TABLE I
IDENTIFICATION STRATEGIES FOR 1000 CLASSES WITH 50 METHODS.

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

The main result of Section VI is that nominal-tag observation achieves the unique Pareto-optimal point with $D = 0$ in the information-barrier regime.

Definition II.26 (Information-barrier domain)**.** A classification domain has an *information barrier* (relative to Φ) if there exist distinct classes $c_1 \neq c_2$ with identical Φ-profiles. Equivalently, π is not injective on classes.

I. Converse: Tag Rate Lower Bound

Theorem II.27 (Converse)**.** *In any information-barrier domain, any scheme achieving $D = 0$ (zero-error identification) requires tag length $L \geq \log_2 k$, where k is the number of distinct classes.*

Proof. Zero error requires that distinct classes map to distinct decoder outputs. With k classes, at least $\log_2 k$ bits are needed to encode k distinct values. The converse is tight: $L = \lceil \log_2 k \rceil$ suffices. ■

J. Rate-Distortion Tradeoff

When $D > 0$ is permitted, we obtain a classic rate-distortion tradeoff:

Proposition II.28 (Lossy identification)**.** *For any $\epsilon \in (0, 1)$, there exist schemes with $L = 0$ (no tags) achieving $D \leq \epsilon$ with query cost $W = O(\log(1/\epsilon) \cdot d)$, where d is the distinguishing dimension. The tradeoff: smaller D requires larger W.*

The zero-error corner ($D = 0$) is special: nominal tagging is the unique Pareto optimum in the information-barrier regime. Relaxing to $D > 0$ enables tag-free schemes at the cost of increased query complexity. This mirrors the classical distinction between zero-error and ϵ-error capacity in channel coding.

K. Concrete Example

Consider a type system with $k = 1000$ classes, each characterized by a subset of $n = 50$ interface methods. 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

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)**.** $E(\textit{nominal-tag}) = O(1)$.

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 interface localization)**.** *$E(\textit{interface-only, declared}) = O(k)$ where k = number of entity classes.*

Proof. With declared interfaces, the constraint "v must satisfy interface I" requires verifying that each class implements all attributes in I. For k classes, $O(k)$ locations. ∎

Theorem III.4 (Attribute-only localization)**.** *$E(\textit{attribute-only}) = \Omega(m)$ where m = 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: $E(\textit{nominal-tag}) = O(1) < \Omega(m) = E(\textit{attribute-only})$ 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, $|I(\textit{attribute-only}, c)| = O(m)$ where m = 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, $|I(\textit{nominal-tag}, c)| = O(1)$.*

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 interface Φ derived from a fixed axis map $\alpha : \mathcal{V} \to \mathcal{A}$ (e.g., $\alpha(v) = (B(v), S(v))$). An observer is permitted to interact with v only through primitive queries in Φ, and each primitive query factors through α: 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 Φ (under our admissibility constraints: no global preprocessing tables, no amortized caching, etc.).

We adopt Φ 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 Φ, 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 Φ, the entire interaction transcript (and hence the output) depends only on $\alpha(v)$. Equivalently, any in-scope semantic property P factors through α: 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 Φ-only strategy can distinguish them with zero error. Any mechanism that does distinguish such pairs must introduce additional information not present in α (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 interface set $\mathcal{I}$, we have $\mathcal{Q} = \{q_I : I \in \mathcal{I}\}$ where $q_I(v) = 1$ iff v satisfies interface I.

In this section, "queries" are the primitive interface 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 $\text{type}(v) \neq \text{type}(w)$, 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 α as in the Model Contract). It depends only on:

- $E = \Phi$ is the ground set of primitive queries (interface predicates).
- "Distinguishing": for all values v, w with $\text{type}(v) \neq \text{type}(w)$, 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 interface Φ already specified). The proof constructs a closure operator satisfying extensivity, monotonicity, and idempotence, from which basis exchange follows (see Lean formalization).

Definition IV.6 (Bases family)**.** Let $E = \Phi$ $(= \mathcal{Q})$ be the ground set of primitive queries (interface 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 sketch. Define the closure operator $\mathrm{cl}(X) = \{q : X\text{-equivalence implies } q\text{-equivalence}\}$. We verify the matroid axioms:

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

The exchange property is the non-trivial step. It follows from the symmetry of indistinguishability. If adding q' to X allows distinguishing v, w that were previously X-equivalent (thus determining q), then v, w must differ on q'. This same pair v, w witnesses that adding q to X allows distinguishing q'. Thus the dependency is symmetric.

Minimal distinguishing sets are exactly the bases of the matroid defined by this closure operator. Full machine-checked proof: `proofs/abstract_class_system.lean`, namespace `AxisClosure`. ■

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 [13]. ■

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 (interfaces). 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 interface-only witness cost)**.** *For any interface-only observer, $W(\text{type-identity}) \geq d$ where d is the distinguishing dimension.*

Proof. Any witness procedure must query at least one minimal distinguishing set. ■

The key insight: the distinguishing dimension is invariant across all minimal query strategies. The difference between nominal-tag and interface-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 Type Identity

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

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

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

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

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

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

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

■

B. Witness Cost Comparison

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

TABLE II
Witness cost for type identity by observer class.

The Lean 4 formalization (Appendix A) provides a machine-checked proof that nominal-tag access minimizes witness cost for type 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 type identifier ($L \geq \log_2 k$ for k types)
- **Witness cost** W: minimum number of primitive queries for type identity checking
- **Distortion** D: probability of misclassification, $D = \Pr[\hat{C} \neq C]$.

We compare two observer classes:

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

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

Theorem VI.3 (Pareto Optimality of Nominal-Tag Observers)**.** *In any information-barrier domain, nominal-tag observers achieve the unique Pareto-optimal point in the* (L, W, D) *space with* $D = 0$*:*

- ***Tag length****:* $L = \lceil \log_2 k \rceil$ *bits for* k *types*
- ***Witness cost****:* $W = O(1)$ *queries (one tag read)*
- ***Distortion****:* $D = 0$ *(zero misclassification probability)*

In such domains, interface-only observers achieve:

- ***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. See Lean formalization: `proofs/python_instantiation.lean`. The proof verifies:

1) `nominal_cost_constant`: Nominal-tag achieves $(L, W, D) = (O(1), O(1), 0)$
2) `interface_cost_linear`: Interface-only admits the generic upper bound $W \leq n$ (query all n attributes); combined with the lower bound $W = \Omega(d)$, this yields an unbounded separation from nominal tagging
3) `python_gap_unbounded`: The cost gap is unbounded in the limit
4) Interface observations alone cannot distinguish provenance; nominal tags can

■

B. Pareto Frontier

The three-dimensional frontier shows:

- Nominal-tag observers dominate interface-only observers on all three dimensions
- Interface-only observers trade tag length for distortion (zero L, but $D > 0$)

Figure 1 visualizes the (L, W, D) tradeoff space. The key observation: *nominal tags trade storage for query cost*, achieving the optimal $(L, W, D) = (\log_2 k, O(1), 0)$ point.

The Lean 4 formalization (Appendix A) provides a machine-checked proof of Pareto optimality for nominal-tag observers in the (L, W, D) tradeoff.

Remark VI.4 (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 interface-only observers (e.g., Python's `hasattr`). *Structural typing* is an intermediate case with $D = 0$ but $W = O(n)$.

Remark VI.5 (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 $s \leq n$.

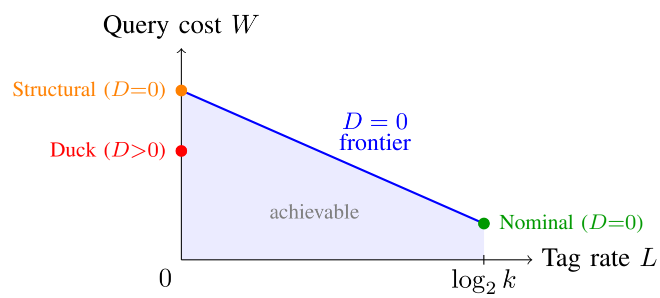

Fig. 1. Schematic illustration of the (L, W, D) tradeoff. For a concrete example with $k = 1000$ classes and distinguishing dimension $d = 10$, the nominal-tag strategy achieves $L = 10$ bits, $W = O(1)$, $D = 0$, while the interface-only strategy requires $W = 10$ queries and incurs $D > 0$ due to attribute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

VII. Instantiations in Real Runtimes

The preceding sections established abstract results about observer classes and witness cost. We now ground these in concrete systems across multiple domains, showing that real classification systems instantiate the theoretical categories and that the complexity bounds are not artifacts of the model but observable properties of deployed implementations.

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 [12]. 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 Runtimes

Type systems are the motivating example for this work. We survey four runtimes.

1) CPython: The `ob_type` Pointer: Every CPython heap object begins with a `PyObject` header containing an `ob_type` pointer to its type object [16]. This is the nominal tag: a single machine word encoding complete type identity.

Witness procedure: Given objects `a` and `b`, type identity is `type(a) is type(b)`—two pointer dereferences and one pointer comparison. Cost: $O(1)$ primitive operations, independent of interface count.

Contrast with `hasattr`: Interface-only observation in Python uses `hasattr(obj, name)` for each required method. To verify an object satisfies a protocol with s required methods requires s attribute lookups. Worse: different call sites may check different subsets, creating $\Omega(m)$ total checks where m is the number of call sites. The nominal tag eliminates this entirely.

2) Java: `.getClass()` and the Method Table: Java's object model stores a pointer to the class object in every instance header [17]. The `.getClass()` method exposes this [18], and `instanceof` checks traverse the class hierarchy.

Key observation: Java's `instanceof` is $O(d)$ where d is inheritance depth, not $O(|\mathcal{I}|)$ where $|\mathcal{I}|$ is the number of interfaces. This is because `instanceof` walks the inheritance hierarchy (a nominal-tag query), not the interface list (an attribute query).

3) TypeScript: Structural Equivalence: TypeScript uses attribute-only (declared) observation [19]: the compiler checks structural compatibility, not nominal identity. Two types are assignment-compatible iff their structures match.

Implication: Type identity checking requires traversing the structure. For a type with s fields/methods, $W(\text{type-identity}) = O(s)$. This is inherent to the observation model: no compilation strategy can reduce this to $O(1)$ without adding nominal tags.

4) Rust: Static Nominal Tags: Rust resolves type identity at compile time via its nominal type system. At runtime, `std::any::TypeId` provides nominal-tag access [20].

The `dyn Trait` case: Rust's trait objects include a vtable pointer but not a type tag [21]. This is attribute-only observation: the vtable encodes which methods exist, not which type provided them.

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	$O(1)$
CPython	`hasattr` probing	`ob_type` pointer	$O(1)$
Java	Interface check	`.getClass()`	$O(1)$
TypeScript	Structural check	(none at runtime)	$O(s)$
Rust (static)	Trait bounds	`TypeId`	$O(1)$

TABLE III
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/DECLARED INTERFACES).

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 $\text{BSC}(\epsilon)$ corruption of the true value.

Definition VIII.2 (Noisy identification capacity)**.** The ϵ*-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(ϵ).

In the noiseless case ($\epsilon = 0$), Theorem II.10 shows the capacity is binary: $\log_2 k$ if π is class-injective, 0 otherwise. For $\epsilon > 0$, several questions arise.

Open problem (noisy identification cost). For $\epsilon > 0$ and class-injective π, 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 Ahlswede-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. We have identified the unique Pareto-optimal point at $D = 0$ (Theorem VI.3), but the full tradeoff surface contains additional structure.

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 $G_1, \ldots, G_{2^L}$, 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**: Interface observation 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 tags is a theorem, not a preference.** In information-barrier domains (Definition II.26), any scheme achieving zero-error identification ($D = 0$) requires tag length $L \geq \log_2 k$ (Theorem II.27). Nominal-tag observation achieves the unique Pareto-optimal $D = 0$ point with $W = O(1)$ at this bound (Theorem VI.3).
- **The barrier is informational, not computational**: even with unbounded resources, attribute-only observers cannot overcome it.
- **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. In information-barrier domains (Definition II.26), nominal-tag observation achieves the unique Pareto-optimal $D = 0$ point in the (L, W, D) tradeoff (Theorem VI.3), and any $D = 0$ scheme necessarily has $L \geq \log_2 k$ (Theorem II.27). 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] P. D. N. Hebert, A. Cywinska, S. L. Ball, and J. R. deWaard, "Biological identifications through dna barcodes," pp. 313–321, 2003.
[13] D. J. A. Welsh, *Matroid Theory*. Academic Press, 1976.
[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.

TABLE IV
LEAN 4 FORMALIZATION MODULES

Module	Lines	Theorems	Purpose
`abstract_class_system.lean`	3082	90+	Two-axis model, information barrier, dominance
`axis_framework.lean`	1667	40+	Query families, closure, matroid structure
`nominal_resolution.lean`	556	21	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`	247	12	Python instantiation
`typescript_instantiation.lean`	65	3	TypeScript instantiation
`java_instantiation.lean`	63	3	Java instantiation
`rust_instantiation.lean`	64	3	Rust instantiation
Core modules subtotal	**6100+**	**190+**	**9 representative modules shown**

[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, 2024.
[21] ——, "The rust reference: Trait objects," https://doc.rust-lang.org/reference/types/trait-object.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.

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 13 Lean files and 265 theorems; the table above highlights the core modules directly used by the main-text derivations.

B. Interface-Only Formalization

Interface-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 interface-only observation, provenance is constant on interface-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 interface 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.

```
-- 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.