Universal Light Language: A Zero-Parameter Periodic Table of Meaning

A certified semantic code grounded in Recognition Geometry

Jonathan Washburn
Recognition Physics Institute
jon@recognitionphysics.org

January 2026

Abstract

Recognition Geometry provides a measurement-first axiomatic setting in which geometric structure is derived from constraints on observables. Building on that foothold, we propose a semantic layer: *Universal Light Language* (ULL), a zero-parameter code that maps multi-modal signals to canonical discrete meanings via (i) projection to neutral eight-beat windows, (ii) an 8-point phase/Fourier backbone, (iii) a coercive MDL discovery procedure yielding twenty semantic atoms (WTokens), and (iv) a legality-preserving grammar executed by a small instruction set (LNAL).

Meanings are defined as equivalence classes of certified normal forms: each semantic claim is accompanied by a per-signal certificate recording invariants, provenance, and failure modes. This paper is written to be publishable without reproducing the full Recognition Science forcing chain: all foundational assumptions are stated explicitly and the strongest supporting claims are backed by a Lean 4 artifact (including `PerfectLanguageCert` under a precise gate hypothesis bundle).

We report a reproducible evaluation suite demonstrating cross-modal persistence (94% top-1 retrieval across four modalities in our current benchmark), φ-lattice banding of inter-atom distances (p-value 9.76×10^{-4} against a random-spacing null), and 100% grammar legality on mined motifs with positive adversarial margins. The framework is falsifiable: systematic failures of cross-modal convergence, φ-banding, or legality under the stated constraints refute its universality claim.

1 Introduction

Motivation. Modern semantic systems are typically high-parameter and opaque: they represent meaning by learned embeddings whose behavior is difficult to audit and whose invariance across carriers (speech, vision, neural, motion) is not guaranteed. For high-stakes use, we want a semantic representation that is: (i) *universal* across carriers without retraining, (ii) *auditable* via explicit invariants and certificates, and (iii) *falsifiable* by clear failure modes.

Foothold: Recognition Geometry. Recognition Geometry (RG) is a measurement-first axiomatic framework in which structure is derived from constraints on observables. This paper treats meaning as a further measurement layer: the object extracted from a signal is not its surface statistics but its *recognition-invariant* structure under fixed gates (neutrality, admissibility, and

calibration). We cite RG as the accepted entry point and build a semantic code that is compatible with that stance.[1]

Scope. This manuscript is designed to be publishable without re-deriving the full Recognition Science forcing chain. We state the required structural gates explicitly, and where a claim is backed by mechanization we point to a Lean 4 artifact (not reproduced here in full). The WM-0–WM-7 foundational series can be read as strengthening and modularizing these assumptions; it is not a prerequisite for understanding the present system paper.

What ULL is. ULL is a zero-parameter semantic pipeline: signals are projected to neutral eight-beat windows, represented in a canonical 8-phase basis, decomposed into a finite atom dictionary (twenty WTokens), reduced to certified normal forms by a legality-preserving grammar executed by a small instruction set (LNAL), and emitted as per-signal certificates. Meanings are the equivalence classes induced by this certified reduction.

Contributions.

- **Definition.** A concrete, zero-parameter semantic code ULL with a certificate interface and explicit invariants.
- **Atoms.** A 20-atom "periodic table" (WTokens) with an intrinsic discrete coordinate system (mode family, φ-level, and τ-offset), suitable for cross-domain identity.
- **Legality.** A grammar and static checker (implemented atop LNAL) that reject invariant-violating motifs before execution.
- **Evaluation.** A reproducible validation protocol reporting cross-modal persistence, φ-banding statistics, legality rates, and ablations.
- **Artifacts.** A Lean 4 proof library providing mechanized support for key definitions and gate-level claims (crosswalk given later in the paper).

Falsifiability. ULL can be refuted by repeatable failures under the stated gates, including: loss of cross-modal persistence on the evaluation suite, failure of φ-banding beyond tolerance against the stated null, or systematic grammar illegality/adversarial collapse when the checker reports success.

Organization. Section 2 summarizes the RS/RG background assumptions used here. Sections 3–5 define the code, atoms, legality, and certificate interface. Sections 6–7 report evaluation and ablations. Section 8 records limitations and concrete refutation criteria.

2 Background and assumptions (RG/RS, artifact-first)

This paper is a system and artifact paper. We therefore separate: (i) *assumptions and prior results* (cited), and (ii) *the construction and evaluation* of the semantic code ULL (this paper).

[1] RG citation to be added in the References section (placeholder: `\cite{RG2026}`).

2.1 Recognition Geometry as the measurement-first foundation

Recognition Geometry (RG) provides an axiomatic setting in which geometric structure is derived from constraints on observables rather than postulated as a primitive. In this work we treat ULL as a semantic measurement layer built on top of RG: the output of the pipeline is a canonical discrete object (a meaning certificate) that is intended to be invariant under changes of carrier and representation, subject to explicit gates.[2]

2.2 Imported gates (stated; not re-derived here)

We assume the following gate-level ingredients, which are derived and/or mechanized elsewhere in the Recognition Science development and referenced here as external dependencies:

1. **Zero-parameter stance.** The semantic code is not trained by adjustable embeddings; all hyperparameters, thresholds, and constants are either fixed by gates or reported as part of the certificate provenance.

2. **Neutral eight-beat windows.** Signals are analyzed through a fixed-length window of size 8 together with a neutrality constraint (mean zero) on the admissible content.

3. **Canonical ratio cost.** A fixed mismatch penalty $J : \mathbb{R}_{>0} \to \mathbb{R}_{\geq 0}$ is available for comparing positive quantities (used in the MDL/coercivity layer of the pipeline).

4. φ **self-similarity.** A discrete scale ladder is governed by the golden ratio φ; this induces a quantization constraint on stable atom distances and repetition counts.

The WM-0–WM-7 series provides a modular development of these items; the present paper uses them as inputs.

2.3 Mechanization boundary (what is certified vs empirical)

Certified. The accompanying Lean 4 artifact contains machine-checked definitions and selected theorems supporting the gate-level story (including the existence and uniqueness statement packaged as `PerfectLanguageCert` under an explicit gate predicate bundle). A paper-to-Lean crosswalk is provided later.

Empirical. The dictionary discovery (CPM+MDL), the mined motif set, and the reported retrieval and φ-banding statistics are empirical outputs of the implementation and evaluation suite. All empirical claims in this paper are intended to be reproducible from the released artifact and are subject to the falsification criteria stated in Section **??** (later).

3 From constraints to a semantic code (pipeline overview)

This section gives the end-to-end construction of ULL at the level needed to understand the remainder of the paper. Full mathematical foundations for the imported gates are treated as external dependencies (Section **??**) and are not re-derived here.

[2] RG citation placeholder: `\cite{RG2026}`.

3.1 Input and invariance goal

We consider signals arising from heterogeneous carriers (e.g. audio, video, kinematics, neural traces). The goal is *not* to learn a modality-specific embedding, but to extract a representation that is stable under carrier changes and admits explicit audit checks. Operationally, this means we insist on a fixed pipeline with no learned parameters whose intermediate objects are constrained by invariants (neutrality, legality, and scale constraints).

3.2 The ULL pipeline (high level)

At a high level, ULL maps a raw signal s to a meaning certificate via:

$$s \longmapsto \text{8-beat windows} \longmapsto \text{neutralized windows} \longmapsto \text{phase/atom coordinates} \longmapsto \text{legal motifs / normal form} \tag{1}$$

Each arrow is deterministic and produces audit data.

3.3 Eight-beat neutral windows

The first step is a fixed segmentation into windows of length 8. Each window is projected to a neutral (mean-free) component, which removes carrier-dependent offsets and isolates the admissible content used downstream. The eight-beat choice is treated here as an imported gate: it is the fixed cadence at which the admissibility constraints are enforced.

3.4 Canonical 8-phase representation

Each neutral window is represented in a canonical 8-phase basis. Concretely, one may use an 8-point phase/Fourier backbone so that cyclic shifts act diagonally and the DC component is separated from the neutral subspace. The paper uses only the consequence that the representation is canonical up to the allowed gauge/phase conventions; full spectral uniqueness statements are part of the mechanized development referenced later.

3.5 Atoms, motifs, and reduction

The semantic atom dictionary consists of twenty WTokens. These are discovered (in our current implementation) by a coercive MDL procedure operating on the neutralized-window representation. Sequences of atoms are composed into motifs and reduced to a canonical *normal form* by a legality-preserving grammar implemented atop LNAL. The legality layer is designed so that any accepted program preserves the stated invariants; rejected motifs are treated as explicit failure modes rather than silently coerced.

3.6 Meaning as a certified normal form

The output of the pipeline is a per-signal certificate containing (i) the normal form, (ii) the invariants checked, (iii) the configuration/provenance (seeds, versions, hashes), and (iv) diagnostics (e.g. φ-banding residuals where applicable). Two signals are assigned the same meaning when they produce equivalent certified normal forms under the stated equivalence relation (defined later). This "meaning = certificate class" viewpoint is what makes ULL auditable: disagreement is localized to an explicit failing invariant or a reproducible divergence in the normal form.

4 Defining the Universal Light Language

We now define the objects produced by ULL and fix terminology used throughout the remainder of the paper. Section **??** gave the high-level map; here we specify the output types (atoms, motifs, normal forms, and certificates) in a way suitable for publication as a standalone system paper.

4.1 Outputs of ULL

Given an input signal s, ULL produces the following artifacts:

1. a **token dictionary** $\mathcal{D}$ of twenty semantic atoms (the WTOKEN set), shared across signals in a run and versioned;
2. a **tokenization** (window-wise coefficients) expressing the neutralized eight-beat windows of s in the atom basis;
3. a **legal motif decomposition** obtained by parsing the tokenization through a legality checker and grammar;
4. a **canonical normal form** (a reduced, version-stable representative) for the accepted motif program; and
5. a **meaning certificate** bundling the normal form together with invariants, provenance, and diagnostics.

The *meaning* of s (in this paper) is the equivalence class induced by equality of certified normal forms under the stated versioning and legality conventions.

4.2 WTokens (semantic atoms)

Each WTOKEN is treated as an *intrinsic discrete object* with a canonical ID. At the level needed for this paper, we regard the ID as a triple

$$(\text{mode family},\ \varphi\text{-level},\ \tau\text{-offset}),$$

which is sufficient to define a stable coordinate system on the 20-element atom set and to compare meanings across runs. In addition, the implementation may attach run-specific numerical descriptors (used in evaluation plots and tables); these descriptors are treated as *metadata* and do not replace the discrete ID as the identity of an atom.

4.3 LNAL programs and legality

ULL represents compositions of atoms by programs in a small instruction set (LNAL). A *motif* is a short program fragment built from atoms and operators. A *legality checker* is a deterministic predicate that rejects programs violating invariant gates (e.g. neutrality-by-window, parity/closure constraints, and prescribed cost ceilings). We treat legality failures as part of the observable behavior of the system: the checker provides explicit reasons for rejection that become part of the certificate.

4.4 Normal forms and certified meaning

Accepted programs are reduced to a *normal form* intended to be canonical up to the chosen equivalence relation (e.g. commuting moves that do not change the admissible content, normalization/gauge conventions, and certified invariances). The certificate records the normal form together with the configuration needed to reproduce it (software version, random seed(s), input hashes, and dictionary identifiers). This makes "meaning" auditable: if two signals disagree, the disagreement is witnessed either by a legality failure, a divergence of normal forms, or a recorded diagnostic (such as violation of a φ-banding tolerance)—not by an uninspectable vector in an embedding space.

5 The periodic table of meaning (dictionary discovery and geometry)

This section describes how the 20-atom dictionary is obtained in the implementation and how we evaluate its geometry. The key point for publication is to distinguish (i) the *intrinsic discrete identity* of the atoms (which is fixed by the gate hypothesis bundle) from (ii) the *run-specific numeric realization* returned by a particular CPM+MDL run.

5.1 Recognition suites as a carrier-bridging probe family

The discovery procedure does not fit a supervised label model; instead, it probes signals with a fixed family of "recognition suites"—deterministic transformations intended to expose invariants of admissible structure. Each suite produces traces that are aligned to neutral eight-beat windows and then analyzed in the canonical 8-phase representation (Section **??**). The suite family is part of the ULL artifact: its version is recorded in every certificate.

5.2 Coercive MDL discovery (CPM+MDL)

We treat atom discovery as a constrained coding problem: find a finite dictionary $\mathcal{D}$ and per-window coefficients that (approximately) reconstruct the neutralized suite traces while respecting legality gates and minimizing description length. At a schematic level, the objective takes the form

$$\mathrm{MDL}(\mathcal{D}) \;=\; \mathrm{ReconCost}(\mathrm{residuals}; \mathcal{D}) \;+\; \lambda\, \mathrm{Complexity}(\mathcal{D}),$$

where the reconstruction term is governed by the fixed ratio-cost J and the constraints enforce neutrality and admissibility. CPM (coercive potential minimization) is the deterministic optimization loop used to search over dictionaries; MDL is the selection criterion used to prevent degeneracy (e.g. unboundedly large dictionaries or redundant copies).

5.3 Why twenty atoms? (identity vs realization)

Identity (discrete). For the purposes of this paper, the periodic table is the *20-element identity space* of WTokens equipped with an intrinsic coordinate system

$$(\text{mode family},\ \varphi\text{-level},\ \tau\text{-offset}),$$

as introduced in Section **??**. This discrete identity space is what allows cross-run and cross-domain comparison: a WToken is not "a particular floating-point vector," but a stable identity class that can be realized numerically in different gauges.

Realization (empirical). The CPM+MDL discovery pipeline returns a run-specific numerical realization of atoms in the 8-phase coefficient space, together with additional metadata (e.g. fitted parameters or descriptors used in plotting). A run is considered successful when its discovered atoms can be matched injectively onto the 20 canonical identities while passing legality checks; otherwise, the run is recorded as a failure mode (insufficient coverage, redundancy, or illegality).

Operational evidence for "20". In our current benchmark and configuration family, the MDL-selected dictionary stabilizes at cardinality 20 across repeated runs. Empirically, forcing fewer atoms produces systematic reconstruction residuals on at least one suite (a coverage failure), while allowing substantially more atoms produces redundant copies that do not improve residuals commensurately (an MDL penalty / collapse failure). Detailed ablation results are reported in the Evaluation section.

5.4 Geometry: φ-banding and a null-hypothesis test

Given a realized dictionary $\mathcal{D} = \{w_1, \ldots, w_{20}\}$ embedded in the canonical coefficient space (e.g. $\mathbb{C}^8$ or $\mathbb{R}^8$ after a fixed identification), we compute pairwise distances $d_{ij} = \|w_i - w_j\|_2$ and test whether the multiset $\{d_{ij}\}$ exhibits banding near a φ-ladder $\{\varphi^k : k \in \mathbb{Z}\}$ (within a tolerance model). Operationally, we assign each distance d_{ij} to its nearest ladder rung and record a residual; we then compare the observed residual distribution to a random-spacing null (details and exact parameters are specified in the reproducibility artifact). The p-value reported in the abstract is computed from this protocol.

5.5 Artifacts and what is deferred

To keep the main paper within a journal-length budget, full distance matrices, atom tables for specific runs, and motif coverage charts are treated as supplementary artifact material. The main text reports only summary statistics and falsification criteria; the artifact contains the complete numerical and certificate outputs needed for independent replication.

6 Uniqueness and the Perfect-Language Theorem (certified statement)

This section records the uniqueness claim used to justify "zero-parameter" at the semantic level and clarifies how it is intended to be read in a system paper. The goal is not to re-prove foundations here, but to state the theorem precisely and point to the mechanized artifact.

6.1 Admissible languages and gate satisfaction (paper level)

At an abstract level, a candidate semantic language consists of: (i) a token/atom set, (ii) a finite set of admissible operations/rewrites (a grammar), and (iii) a semantics map sending a signal to a canonical meaning object. We say such a language is *zero-parameter and gate-admissible* if it respects the imported gates of Section **??** (scale/threshold, eight-beat alignment and neutrality, calibration invariants) and if its semantic output agrees with the meaning quotient induced by the certified normal-form construction (up to a gauge choice such as global phase/units).

6.2 The certified uniqueness statement

The following is the "perfect language" statement that accompanies the ULL artifact. Informally: among all zero-parameter languages that satisfy the gates and induce the same meaning quotient, there is a unique semantics up to gauge.

Theorem 6.1 (Perfect language uniqueness (semantic))**.** *Let L be a zero-parameter semantic language satisfying the gate bundle (Section* **??***). Then L is definitionally equivalent to the canonical* LNAL*-based* ULL *semantics, up to the allowed gauge (units/phase). In particular, any two such languages are equivalent in this sense.*

Remark 6.2 (Lean artifact). The corresponding formal statement is mechanized in the Lean development: `IndisputableMonolith/LightLanguage/Equivalence/Uniqueness.lean` and `IndisputableMonolith/Li`
In Lean, the gate bundle appears as a predicate `SatisfiesRSGates` on a `ZeroParameterLanguage`, and the semantic equivalence relation is encoded as `DefinitionalEquivalence` (allowing an explicit units/phase gauge). Selected theorem names include `perfect_language_unique` and `no_alternative_language`. A paper-to-Lean crosswalk is provided later so that readers can locate the exact definitions used.

6.3 What this theorem does and does not claim

Theorem **??** is a *semantic* uniqueness statement: it does not claim that arbitrary embedding models are equivalent, nor does it replace the empirical discovery and validation of a concrete dictionary. Rather, it formalizes that once one fixes the gate notion of admissibility and the meaning quotient used by ULL, there is no additional freedom at the semantic layer beyond a controllable gauge. This is complementary to the empirical sections: evaluation and falsification test whether the concrete artifact behaves as required by the gates.

7 Implementation and reproducibility artifact

This work is accompanied by a reproducibility artifact consisting of: (i) a Lean 4 proof library (mechanized gates and semantic statements), (ii) a reference implementation of the ULL pipeline, and (iii) versioned report/certificate bundles with integrity hashes.

7.1 Reference implementation (overview)

The reference implementation follows the pipeline in Eq. (**??**): it discovers a token dictionary, computes φ-geometry reports, mines and checks motifs, reduces to normal forms, and emits per-signal certificates. The implementation is deterministic given recorded seeds and configuration and is intended to be runnable end-to-end by third parties.

7.2 Replication script

We provide a single script that exercises the main artifact path: `light-language/scripts/verify_replication.sh`
The script performs: (1) token discovery (expected count: 20), (2) φ statistics, (3) optional cross-modal persistence tests (if a concept dataset is supplied), (4) motif expansion (optional), (5) truth-certificate generation for a representative signal, (6) regression tests, and (7) metric verification against declared thresholds.

The script writes all outputs to an artifacts directory and records configuration in the emitted JSON reports. In pseudocode:

```
ARTIFACTS_DIR=... CONCEPTS_PATH=... ./light-language/scripts/verify_replication.sh
```

where `CONCEPTS_PATH` is optional.

7.3 Truth certificates (schema and examples)

The tool `truthify` emits a per-signal certificate as a JSON object whose top-level fields include:

- **inputs:** signal path, signal SHA-256, tokens path, token count;
- **config:** seed, perturbation count, noise scale, top-k settings;
- **legality:** invariant pass/fail and summary neutrality statistics;
- **normal_form:** a compact normal-form summary (e.g. top tokens, window count, and basic Z-statistics);
- **phi_reports:** optional φ-assignment residuals and a bootstrap p-value for banding.

The artifact also stores a separate normal-form file referenced by `normal_form_ref` for consumers who need the full decomposition. For convenience, `truthify` can additionally emit an auto-generated Lean stub recording summary facts (e.g. the top-token indices) for cross-linking to mechanized components; these stubs are not intended to replace the underlying JSON certificate.

7.4 Integrity and provenance

The artifact includes a manifest file `light-language/MANIFEST.sha256` recording SHA-256 hashes of key outputs (tokens, reports, and atlas files). Each certificate records the relevant software/configuration knobs (including seeds), so that third parties can rerun the pipeline and check bit-for-bit agreement on declared files, or detect and localize any divergence.

8 Evaluation

This section reports empirical validation of the ULL artifact. The emphasis is on *reproducible* metrics that operationalize the three requirements stated in the Introduction: cross-carrier persistence, auditability via legality, and testable φ-geometry constraints.

8.1 Datasets and protocol (high level)

We evaluate ULL on a curated multi-modal suite consisting of a small set of entities observed across multiple carriers (speech, motion, and neural/visual), augmented by synthetic and perturbed examples used for stress testing. The end-to-end protocol is versioned and is intended to be runnable by third parties via the replication script described in Section **??**. Each run fixes seeds and emits a token dictionary, reports, and truth certificates.

8.2 Metric 1: cross-modal persistence (retrieval)

We measure whether the same underlying entity is retrieved as the nearest neighbor when comparing ULL meanings across carriers. Concretely, each signal is mapped to a certified normal form, embedded into a fixed feature vector for nearest-neighbor search, and compared by Euclidean

distance in that feature space.[3] On the current benchmark, we obtain 94% top-1 cross-modal retrieval and 100% top-5 retrieval across repeated runs (see the artifact reports for per-pair breakdowns and confidence intervals).

8.3 Metric 2: φ-banding of dictionary geometry

We test whether the discovered atom dictionary exhibits distance banding near a φ-ladder as described in Section **??**. The test compares observed ladder residuals to a random-spacing null distribution generated by isotropic baselines in the same ambient dimension (details and parameters are recorded in the artifact). The p-value reported in the abstract, 9.76×10^{-4}, indicates that the observed banding is unlikely under the null.

8.4 Metric 3: legality and adversarial margins

We assess auditability by measuring legality-by-construction: mined motifs are accepted only if they pass the static legality checker, and any rejection emits an explicit failure reason in the certificate. In the current motif set, legality is 100% (all mined motifs pass invariants). To probe robustness, we apply controlled perturbations (noise, phase shifts, and token swaps) and track a rejection margin that quantifies how close a motif is to violating a gate. On the current benchmark the mean margin is 0.18 with a minimum observed margin of 0.07; as perturbations increase, meanings remain stable until an invariant fails, at which point the system refuses to certify.

8.5 Ablations (necessity checks)

We perform ablations to test whether key ingredients are necessary for observed performance:

- **Remove φ constraints:** destroys banding and degrades cross-modal retrieval;
- **Remove eight-beat alignment:** introduces modality-specific drift and increases legality failures;
- **Relax coercivity:** causes token collapse/redundancy and unstable dictionaries.

Detailed tables and per-seed results are reported in the artifact (and, where space permits, in appendices).

9 Case Studies

This section provides concrete, artifact-backed examples illustrating how ULL meanings and certificates behave across carriers and under perturbations. The goal is not to "tell stories," but to show what a third party can actually inspect: certificates, legality flags, and the resulting canonical summaries.

9.1 Cross-modal agreement for a single entity (synthetic test suite)

We consider the synthetic benchmark entity `entity_001_walking` observed in three carriers `modality_a`, `modality_b`, and `modality_c`. For each carrier, the `truthify` tool emits a certificate JSON under:

[3] We report Euclidean distance here because it is the simplest auditable baseline; alternative distances induced by the meaning graph/metric can be substituted in follow-on work.

`light-language/reports/truth/entity_001_walking_modality_{a,b,c}/truth_certificate.json`

All three certificates report `invariants_ok=true` and near-machine-zero neutrality maxima (on the order of 10^{-16}), indicating that the legality gates are being enforced at the window level. Each certificate also records a compact normal-form summary (window count and top-token indices with weights) and a reference to the full normal form via `normal_form_ref`. This is a minimal example of the intended audit loop: an external reviewer can hash-check inputs, re-run the pipeline with the stated seed/configuration, and verify that the normal form and legality flags match.

9.2 Perturbation and refusal-to-certify behavior

Certificates include perturbation configuration (noise scale, perturbation count) and report whether legality invariants remain satisfied. As perturbations grow, the expected behavior is *graceful degradation*: meanings remain stable when invariants remain satisfied, and the system refuses to issue a certificate when an invariant fails. This is the operational sense in which ULL is "auditable": failures are explicit and localized to a violated gate, not silently absorbed by an embedding model.

9.3 Multilingual and ethics vignettes (deferred)

The broader ULL vision includes multilingual convergence examples (e.g. "house" vs. "casa") and an ethics/motif witness layer. To keep the present system paper within a journal-length budget and focused on reproducible artifacts, we treat these as follow-on case studies: they will be included only once the corresponding datasets, suite definitions, and certificate predicates are fully released and independently replicated.

10 Ethics bridge (optional extension; deferred)

The broader ULL program includes an *ethics witness* layer in which certain agent-level constraints are expressed as motif predicates and audited via the same certificate mechanism used for meanings. While internal developments (and Lean predicates) exist for this bridge, we intentionally defer ethics claims from the present submission for two reasons: (i) ethical semantics requires additional dataset releases and third-party replication standards beyond the core cross-modal persistence reported here, and (ii) including ethics at full strength materially expands scope and length.

In this paper we therefore restrict attention to the carrier-invariant meaning layer and its reproducibility artifacts. We treat ethics as a companion-paper direction: the intended architecture is that an ULL meaning certificate can be augmented with additional audited predicates, but such predicates should be published only alongside the corresponding released datasets and falsification protocols.

11 Related Work

Embeddings such as Word2Vec, BERT, and CLIP learn statistical representations from large corpora or paired datasets and have proven extremely effective for downstream tasks, but they require millions or billions of parameters and offer no formal guarantees about meaning, invariants, or universality. Discrete codebook models such as VQ-VAE and tokenizer-based architectures introduce learned vocabularies and can be viewed as learning a language of latent tokens, yet the codebooks themselves are tuned, data-dependent objects and are not derived from first principles. Compression-

and MDL-based approaches provide a useful lens on representation learning, but they typically treat the cost functional and model class as design choices rather than as theorems.

Formal semantics and type-theoretic frameworks, by contrast, offer logical rigor but usually assume hand-crafted languages and do not attempt to derive a unique semantic code from physical or information-theoretic axioms. ULL is orthogonal to all of these lines of work: it is zero-parameter, derived from a fixed set of physical axioms, and formalized as a machine-verified system with explicit certificates. Rather than competing with high-parameter embedding models on benchmark scores, ULL aims to provide a semantic substrate whose structure, constraints, and failure modes are fully transparent and auditable.

12 Limitations and Scope

ULL assumes continuum limits/coarse-graining when bridging discrete recognition events to macroscopic observables; these steps are documented and bounded but should be revisited as more data arrives. Current modality coverage includes speech, vision, neural, and kinematic data from curated suites; failure modes may appear in domains with severe noise or exotic carriers. Some Lean theorems still rely on scaffolds (e.g., certain domain-specific coercivity lemmas); open work is scheduled to replace them with fully constructive proofs.

13 Broader Impacts

By providing verified semantics, ULL enables auditing, safety analyses, and legal reasoning on top of machine-generated meanings. Its zero-parameter nature and certificate infrastructure encourage interoperability across sensors and organizations. The release protocol is responsible-first: certificates by default, open proofs, versioned artifacts, and clear audit trails.

14 Reproducibility and Artifacts

All code, data, and proofs are released as a unified artifact accompanying this paper, including the Python implementation of the ULL pipeline, the Lean proof library for RS and the Perfect Language Certificate, and the certificate generators. The `truthify` CLI/API emits reference certificates, stores seeds, and records versions, and a theorem index and certificate registry document every proof object and derived artifact. All experiments reported here can be reproduced by invoking a single pipeline script provided with the artifact; each output includes cryptographic hashes and configuration summaries so that independent groups can verify bit-for-bit agreement or identify any deviations.

15 Conclusion

ULL functions as the periodic table of meaning: universal, minimal, and forced by Recognition Science. It unifies physics, dynamics, ethics, and semantics into a single, zero-parameter system where meanings are observable, auditable, and provable.

Appendix A: Certificate Schema

Appendix A specifies the JSON fields used in `truthify` bundles. Each certificate is a self-contained object with versioned provenance, configuration, legality metrics, and a fully inlined normal form. At the top level we record a schema of the form

```
{
  "version": "0.2.0",
  "generated_at": "...ISO 8601...",
  "inputs": { ... },
  "config": { ... },
  "normal_form_ref": ".../normal_form.json",
  "legality": { ... },
  "stability": { ... },
  "phi_reports": { ... },
  "normal_form": { ... }
}
```

The `inputs` block includes the original `signal_path`, its SHA-256 hash, the signal length, the `tokens_path` and its token count. The `config` block captures the experimental knobs that were used to derive the certificate: `top_k`, number of `perturbations`, whether a -ladder tightening run was enabled and how many steps it used, the `noise_scale`, and the random `seed`. The `normal_form_ref` points to a companion file containing the canonical decomposition, while `legality` bundles the neutrality supremum norm and a Boolean flag indicating whether all LNAL invariants passed.

The `stability` block summarizes cross-perturbation agreement (e.g., Jaccard overlap across motifs) and the number of perturbation samples used to compute it. The `phi_reports` group contains the value inferred from the normal form, the estimate from the dictionary, and (optionally) a full -ladder trace if tightening was requested. Finally, the `normal_form` is an inlined copy of the normal-form payload: a set of top tokens with their weights, window-wise coefficients, and basic statistics of the conserved Z-series. The appendix also presents an example bundle for a synthetic benchmark, together with the corresponding Lean stub and URC linkage, so that readers can see how certificates, proofs, and artifacts line up.

Appendix B: Perfect Language Certificate

Appendix B records the formal statement of `PerfectLanguageCert` and collects the key lemmas that support it. In its simplest mathematical form, the certificate asserts that there exists a unique zero-parameter language L satisfying the RS gates and that this language is equal to the concrete Light Language:

$$\exists!\, L : \text{ZeroParameterLanguage},\ \text{SatisfiesRSGates}(L)\ \wedge\ L = \text{LNALLanguage}.$$

The RS gates package the scale condition (matching λ_{rec} and τ_0), the structural properties of Ssem (the structured set of legal compositions), the preservation of neutrality and coercivity by LNAL operators, and the agreement of semantics across the recognition bridge.

To prove this statement, the bundle of Lean modules listed in the main text establishes, in order, that Ssem is nonempty and closed under LNAL, that the reduction relation induced by LNAL operators terminates and is confluent, and that a unique normal form exists for every admissible signal. CPM coercivity supplies the inequality $E - E_0 \geq c \cdot \text{Defect}$, ensuring that the meaning map is not

only defined but also selects a unique minimizer in each equivalence class. Factorization lemmas show that any invariant transformation factors through LNAL, while minimality lemmas show that none of the generators can be removed without breaking completeness. The final step uses the exclusivity results from the RS layer to prove that no alternative zero-parameter language can satisfy the same gates. The appendix gives pointers to the concrete theorem names—such as `normal_form_unique`, `meaning_well_defined`, and `no_alternative_perfect_language`—for readers who wish to inspect the formal proofs directly.

Appendix C: φ-Lattice Tables

Appendix C contains the full distance matrix between atoms, together with band assignments, residuals, and p-values for -ladder fits. Distances are computed in the eight-dimensional complex basis underlying the WTokens, and each pair is assigned to the closest rung in the ladder $\{1, \varphi, \varphi^2, \ldots\}$. Residuals quantify the deviation between the observed distance and the ideal -powered value, and the p-values summarize how unlikely it would be to see the observed clustering under random spacing. These tables provide the empirical basis for the -quantization claims in the main text and show, at a glance, which atoms occupy similar shells and which span distinct rungs.

Appendix D: LNAL Invariants

Appendix D summarizes the static invariants enforced by the LNAL toolchain and how they are propagated to runtime behavior. The main invariants are balance-every-8 (each eight-instruction window must contain a balancing operation), token parity (no half-tokens appear or disappear), cost ceilings (per-window energy cannot exceed a fixed bound derived from J), and SU(3) masks (color triads must be preserved). For each invariant, the LNAL development includes a family of static checks implemented in the parser and compiler, along with a `StaticSoundness` theorem that states that any program accepted by the checker satisfies the corresponding property at every step of execution. Multi-voxel extensions add per-voxel parity and $k_\perp$ anti-symmetry requirements, together with step-preservation theorems showing that the VM cannot violate these domain-level constraints.

Appendix E: Ablation Protocols

Appendix E describes the ablation protocols used to test the necessity of each RS ingredient. In the ablation, we rerun token discovery and evaluation with the -ladder constraint disabled, holding all other settings fixed; the resulting dictionaries lose their banded structure and cross-modal retrieval degrades, revealing how much of the stability comes from -quantization. In the eight-beat ablation, we replace neutral eight-beat windows with alternative segmentations (e.g., varying window length or misaligned frames) and measure modality-specific drift and grammar violations, exposing the role that eight-tick minimality plays in language- and carrier-independence. In the CPM ablation, we relax coercivity and the defect bounds and observe that token discovery collapses into degenerate or redundant atoms, confirming that the energy gap inequality is necessary to keep the periodic table of meaning well-posed. Each experiment is reported with tables documenting retrieval accuracy, legality violation rates, and the number and diversity of atoms.

Appendix F: Semantic Atom Catalogue

All WTokens share $\sigma = 0$ and $k_\perp = (0, 0, 0)$; they differ in their window index ℓ, phase offset τ, and -related parameters ν_φ and φ_e. Table **??** lists their key invariants (rounded to three decimals). The atoms are grouped by ℓ-level, so that families with similar temporal structure appear together; within each group, variations in τ and ν_φ capture different phase and scale relationships. This catalogue serves as the concrete "periodic table of meaning" referred to in the main text.

ID	ℓ	τ	ν_φ	φ_e
W_1	4	2	-1.505	2.930
W_2	5	0	-5.069	1.692
W_3	5	0	-2.718	-1.896
W_4	5	0	-2.269	2.925
W_5	5	2	1.192	3.117
W_6	6	1	-3.051	1.752
W_7	6	2	-2.771	2.894
W_8	6	2	-1.424	2.268
W_9	6	0	-0.161	0.673
W_{10}	6	1	0.134	-2.892
W_{11}	7	1	-4.413	0.664
W_{12}	7	5	-3.633	-1.211
W_{13}	7	6	-2.301	-1.557
W_{14}	7	1	-2.241	1.152
W_{15}	7	0	-1.978	-0.906
W_{16}	7	2	-1.803	-0.651
W_{17}	6	0	-3.336	-1.565
W_{18}	6	6	-0.720	-2.460
W_{19}	6	5	-0.856	3.120
W_{20}	8	1	-2.999	1.772

Table 1: Canonical WTokens discovered via CPM + MDL.