A Lean-Referenced Derivation of the Electromagnetic Fine-Structure Constant from Recognition Ledger Geometry

Recognition Science Collaboration

Recognition Physics Institute

(Dated: December 23, 2025)

Abstract

This paper documents the Recognition Science derivation of the electromagnetic fine-structure constant in the exact form implemented in this repository's Lean 4 development. The derived inverse constant is:

$$\alpha^{-1} = 4\pi \cdot 11 \; - \; f_{\text{gap}} \; - \; \delta_\kappa, \quad \text{where} \quad \delta_\kappa = -\frac{103}{102\,\pi^5}.$$

We explain each factor: *what* enters this formula, *why* it is structurally forced by the ledger/cube interface, and *where* each component lives in the Lean codebase. Integer/combinatorial components (11, 102, 103), the solid-angle factor (4π), and the π^5 power are proved in Lean; the gap weight w_8 is currently an external certificate, with a DFT-8 derivation in progress.

I. WHAT "ALPHA" MEANS IN THIS REPO

This repository contains *two* distinct "alpha" quantities:

- **Electromagnetic fine-structure constant:** $\alpha_{\text{EM}} \approx 1/137$, implemented as `alpha := 1/alphaInv` in `Constants.Alpha`, with $\alpha^{-1} \approx 137.036$. This is the subject of this paper.

- **Locked kernel exponent:** $\alpha_{\text{lock}} := (1 - 1/\phi)/2 \approx 0.191$, implemented as `alphaLock` and used as an exponent in the ILG kernel. It is *not* equal to α_{EM}.

Throughout, "α" or "fine-structure constant" refers to α_{EM}.

II. THE LEAN TOP-LEVEL DEFINITION

The *symbolic* definition of α_{EM} is in `Constants/Alpha.lean`. The core definitions are:

$$\begin{aligned}
\texttt{alpha_seed} &:= 4 \cdot \pi \cdot 11, \\
\texttt{delta_kappa} &:= -103/(102 \cdot \pi^5), \\
\texttt{alphaInv} &:= \texttt{alpha_seed} - (\texttt{f_gap} + \texttt{delta_kappa}), \\
\texttt{alpha} &:= 1/\texttt{alphaInv}.
\end{aligned}$$

For reference, here is the exact Lean snippet (abridged):

```
-- IndisputableMonolith/Constants/Alpha.lean
@[simp] def alpha_seed : Real := 4 * Real.pi * 11
@[simp] def delta_kappa : Real := -(103 : Real) / (102 * Real.pi ^ 5)
@[simp] def alphaInv : Real := alpha_seed - (f_gap + delta_kappa)
@[simp] def alpha : Real := 1 / alphaInv
```

Everything else in this paper explains why these specific factors appear.

III. FIRST-PRINCIPLES STRUCTURE: SEED–GAP–CURVATURE

In Recognition Science, the coupling constant is not introduced as a measured parameter; it is assembled as a *dimensionless closure balance* across the discrete ledger-to-continuum interface:

$$\alpha^{-1} = \underbrace{(\text{geometric seed})}_{\text{baseline closure}} - \underbrace{(\text{gap cost})}_{\text{8-tick spectral deficit}} - \underbrace{(\text{curvature correction})}_{\text{tiling/phase-space mismatch}}.$$

Lean implements this as:

$$\alpha^{-1} = \texttt{alpha\_seed} - (\texttt{f\_gap} + \texttt{delta\_kappa}),$$

where `delta_kappa` < 0. Thus subtracting it adds a small positive curvature correction.

IV. WHY $D = 3$ (BRIEF): THE GAP45 SYNCHRONIZATION FORCING

In Recognition Science, the cube geometry used in the α_{EM} pipeline is not an arbitrary modeling choice: $D = 3$ is structurally forced by the eight-tick closure and the "Gap45" synchronization constraint. The certificate is `Verification/Gap45DimensionCert.lean`.

In that development, the main idea is:

- The eight-tick period is 2^D. For $D = 3$, $2^D = 8$.
- A closure/fibonacci construction yields a "gap" of 45:

  $$45 = (8 + 1) \cdot 5,$$

  where $8 + 1$ is a wrap-around closure factor and $5 = \text{fib}(4)$ is the smallest nontrivial Fibonacci factor coprime to 8.

- A global synchronization period is imposed:

$$\mathrm{lcm}(2^D, 45) = 360.$$

  Since $\gcd(2^D, 45) = 1$ (no factor of 2 in 45), this reduces to $2^D \cdot 45 = 360$, hence $2^D = 8$ and therefore $D = 3$.

For the α_{EM} derivation, `AlphaDerivation.lean` sets $D := 3$ directly; the Gap45 certificate is the upstream justification.

V. WHY THE NUMBER 11: PASSIVE EDGES OF THE D=3 CUBE

The integer **11** is the passive-edge count of the 3-cube Q_3. This is formalized in `AlphaDerivation.lean` and certified in `CubeGeometryCert.lean`.

A. Cube combinatorics in Lean

Lean defines the hypercube edge/face/vertex counts:

$$V(D) = 2^D, \tag{1}$$

$$E(D) = D \cdot 2^{D-1}, \tag{2}$$

$$F(D) = 2D, \tag{3}$$

as `cube_vertices`, `cube_edges`, `cube_faces`. With $D = 3$, Lean proves: $V(3) = 8$, $E(3) = 12$, $F(3) = 6$.

B. Active vs. passive edges

The ledger dynamics require one "active" edge transition per atomic tick. Lean encodes:

$$\texttt{active\_edges\_per\_tick} := 1, \qquad \texttt{passive\_field\_edges}(D) := E(D) - 1.$$

For $D = 3$, the key theorem is `passive_field_edges` $3 = 11$.

Interpretation: the single active edge realizes the local transition; the remaining 11 edges represent the "field dressing" paths around that transition—this is the structural origin of 11 in `alpha_seed`.

VI. WHY 4π: ISOTROPIC SOLID-ANGLE CLOSURE IN D=3

The factor 4π appears because the seed is a *spherical closure* over directions in three spatial dimensions. In Lean, the derivation is in `Constants/SolidAngleExclusivity.lean`.

A. Sphere surface area in arbitrary D

Lean defines the surface area of the unit $(D-1)$-sphere embedded in $\mathbb{R}^D$ by the standard Gamma-function formula:

$$S_{D-1} = \frac{2\pi^{D/2}}{\Gamma(D/2)}.$$

This is `unitSphereSurface`.

B. Specializing to $D = 3$

Lean proves $S_2 = 4\pi$ via `unitSphereSurface_D3`.

C. Seed assembly

With the passive-edge count 11 and isotropic direction measure 4π, Lean defines:

$$\texttt{alpha\_seed} := 4\pi \cdot 11.$$

This is the "baseline" inverse coupling before spectral gap and curvature corrections.

VII. THE CURVATURE CORRECTION: WHY 103/102 AND WHY π^5

The curvature correction is:

$$\texttt{delta\_kappa} = -\frac{103}{102\pi^5}.$$

The integer provenance is proved in `Constants/AlphaDerivation.lean`; the π^5 power is explained in `Constants/CurvatureSpaceDerivation.lean`.

A. Why 102 and 103: voxel seam counting

Lean defines the denominator as:

$$\texttt{seam\_denominator}(D) := F(D) \cdot W,$$

where $W := 17$ is the number of wallpaper groups (plane symmetry groups). For $D = 3$, $F(3) = 6$, so $6 \cdot 17 = 102$.

Lean then adds an explicit closure increment:

$$\texttt{euler\_closure} := 1, \quad \texttt{seam\_numerator}(D) := \texttt{seam\_denominator}(D) + 1,$$

so for $D = 3$: $102 + 1 = 103$.

B. Why π^5: 5D configuration-space angular measure

Lean models the curvature integral as living on an effective configuration space of dimension:

$$3 \text{ (space)} + 1 \text{ (time/phase)} + 1 \text{ (dual-balance)} = 5,$$

encoded as `configSpaceDim` $:= 5$. Each dimension contributes a factor of π under the angular normalization, yielding π^5 in the denominator.

C. Sign of the correction

Lean encodes the curvature term as negative (`delta_kappa` < 0). In the top-level assembly, subtracting a negative value adds a small positive correction. The seam mismatch is a *defect* term; the sign convention treats it as a negative correction inside the parenthesis.

VIII. THE GAP TERM $f_{\text{gap}} = w_8 \ln(\phi)$

The gap term is the information/spectral deficit arising from the eight-tick periodicity and the ϕ-lattice scaling at the discrete–continuous interface. In Lean, two layers exist:

- **Pipeline constant:** `Constants/GapWeight.lean` (currently used).
- **DFT-8 derivation:** `Constants/GapWeightDerivation.lean`, with positivity certified in `Verification/GapWeightDerivationCert.lean`.

A. Why $\ln(\phi)$

Recognition Science distinguishes multiplicative self-similar scaling (natural on the ϕ-lattice) from additive bookkeeping cost (natural for ledger balance). The logarithm is the unique bridge between the two:

$$\ln(\phi^n) = n \ln(\phi).$$

Thus a unit "scale step" carries additive cost proportional to $\ln \phi$. Lean uses `Real.log phi` for this quantity.

B. What w_8 is

In the DFT-8 derivation module, Lean defines the canonical ϕ-pattern on eight ticks:

$$p(t) = \phi^t, \qquad t \in \{0, 1, \ldots, 7\},$$

as `phiPattern`. It then defines DFT coefficients c_k and mode energies $|c_k|^2$. The weight w_8 is defined as a weighted sum over the neutral modes $k \neq 0$:

$$w_8 = \sum_{k=1}^{7} |c_k|^2 \, g_k(\phi),$$

where $g_k(\phi)$ are geometric weights from the 8-tick frequency alignment and ϕ-scaling. This is `w8_computed`.

C. DFT-8 coefficients and closed form

The DFT-8 coefficient implemented in `GapWeightDerivation.lean` is:

$$c_k \; := \sum_{t \in \mathrm{Fin}\, 8} \overline{\mathrm{DFT}_{t,k}} \, \phi^t,$$

encoded as:

```
-- IndisputableMonolith/Constants/GapWeightDerivation.lean
noncomputable def phiDFTCoeff (k : Fin 8) : Complex :=
  Finset.univ.sum fun t => star (dft8_entry t k) * (phi ^ t.val : Complex)
```

Two key proved facts are:

- **Nontriviality:** the mode-1 coefficient is nonzero (`phiDFTCoeff_one_ne_zero`). The proof uses the geometric series identity $\sum_{t=0}^{7} r^t = (r^8 - 1)/(r - 1)$ with $r := \overline{\omega_8}\phi$, together with $\phi^8 = 21\phi + 13$, to show $r^8 \neq 1$.

- **Positivity:** $w_8 > 0$ (`w8_computed_pos`), since every summand $|c_k|^2 g_k(\phi)$ is nonnegative and at least k=1 is strictly positive.

D. The geometric weights $g_k(\phi)$

The per-mode weight is:

$$g_k(\phi) = \begin{cases} 0, & k = 0, \\ \sin^2\!\left(\frac{\pi k}{8}\right) \phi^{-k}, & k \in \{1, \ldots, 7\}. \end{cases}$$

E. Current pipeline status

The top-level `alphaInv` currently uses:

$$\texttt{w8\_from\_eight\_tick} := 2.488254397846, \quad \texttt{f\_gap} := \texttt{w8\_from\_eight\_tick} \cdot \ln(\phi),$$

documented as a quarantined certificate in `GapWeight.lean`. The JSON file `w8.json` records w_8, ϕ, $\ln\phi$, f_{gap}, δ_κ, and α^{-1}.

Separately, `GapWeightDerivation.lean` proves positivity of w_8 and non-vanishing of the mode-1 coefficient, and records `w8_computed_eq_abstract` as a named proposition pending final integration.

IX. ASSEMBLING α^{-1} AND NUMERICAL BOUNDS

A. Assembly theorem

The symbolic assembly is in `Alpha.lean`, connected to the ingredient derivations by `alphaInv_derived_eq_formula`.

B. Numeric evaluation (quarantined)

Numeric checks are quarantined in `AlphaNumericsScaffold.lean`; the JSON certificate records $\alpha^{-1} \approx 137.036$.

C. Rigorous interval bounds (proved)

Rigorous inequality bounds are in `AlphaBounds.lean`. That module proves:

$$137.031 < \texttt{alphaInv} < 137.040,$$

using certified bounds on π and coarse bounds on $\ln \phi$. Tightening the logarithm bounds correspondingly tightens the α^{-1} interval.

X. LEAN CROSS-REFERENCE MAP (WHERE EACH FACTOR COMES FROM)

Factor	**Meaning**	**Lean module**
$D = 3$	dimension forcing	`Verification/Gap45DimensionCert`
11	passive edges $(12-1)$	`Constants/AlphaDerivation`
4π	solid angle in 3D	`Constants/SolidAngleExclusivity`
17	wallpaper groups	`Constants/AlphaDerivation`
102	6×17 (seam denom.)	`Constants/AlphaDerivation`
103	$102 + 1$ (Euler closure)	`Constants/AlphaDerivation`
π^5	5D config. space	`Constants/CurvatureSpaceDerivation`
w_8	8-tick gap weight	`Constants/GapWeight[Derivation]`
$\ln \phi$	additive scale cost	Mathlib `Real.log`

TABLE I. Provenance map for each element of the α^{-1} derivation.

XI. CONCLUSION

The Recognition Science fine-structure pipeline implemented in Lean decomposes $\alpha_{\mathrm{EM}}^{-1}$ into:

- a geometric seed $4\pi \cdot 11$ forced by $D{=}3$ cube combinatorics and isotropy,
- an eight-tick gap cost $f_{\text{gap}} = w_8 \ln(\phi)$ capturing the neutral spectral deficit of the ϕ-pattern under DFT-8,
- and a curvature/tiling correction $\delta_\kappa = -103/(102\pi^5)$ from seam topology over a 5D effective configuration space.

Lean makes the dependency structure explicit and (for most components) machine-checkable. The remaining open step is to replace the pipeline constant `w8_from_eight_tick` with the derived DFT-8 expression `w8_computed` (or prove their equality), closing the last derivation gap inside the certified surface.

[1] Lean 4 theorem prover, https://leanprover.github.io/.

[2] Mathlib: the Lean mathematical library, https://github.com/leanprover-community/mathlib.

[3] E. S. Fedorov, "Symmetry of regular systems of figures," *Zapiski Imp. S.-Peterburgskogo Mineral. Obshchestva* **28**, 1–146 (1891).

[4] J. H. Conway, H. Burgiel, C. Goodman-Strauss, *The Symmetries of Things*, A K Peters (2008).

[5] Lean modules referenced: `Constants/{Alpha, AlphaDerivation, SolidAngleExclusivity, CurvatureSpaceDerivation, GapWeight, GapWeightDerivation}.lean`, `Numerics/Interval/AlphaBounds.lean`, and `data/certificates/w8.json`.