The Geometric Observer Network

A Zero-Parameter Architecture for Relativistic Phase Dynamics from S^3 Geometry

Léon Fernando Vlegels
Independent Researcher

2025-12

Contents

Abstract

We specify a concrete computational architecture, the Geometric Observer Network (GON), whose forward pass exactly computes the relativistic lapse, Hopf observables, and phase dynamics established in the geometric Theory of Everything. The architecture contains zero trainable parameters: every operation (quaternion multiplication, Hopf projection, log map, measure scaling) is hardcoded from proven identities. Inputs are initial conditions and generator schedules; outputs are time-dependent observables including the Lorentz lapse $\sqrt{1-v^2}$. We define a rigorous benchmarking protocol comparing the GON against trained multilayer perceptrons (MLPs) of matched or greater capacity, and formulate three falsifiable predictions: (1) the GON achieves zero error on the Hopf identity by construction; (2) a trained MLP cannot reach zero error and converges to a residual floor determined by its capacity; (3) promoting GON constants to trainable parameters and applying gradient descent *increases* mean error. These predictions operationalize the training–exposure distinction of Paper 28 as a reproducible experiment.

1 Introduction

Paper 27 [4] derived a set of exact identities governing dual-observer dynamics on S^3, and Paper 28 [5] established that computational graphs implementing such identities are degraded by gradient descent. This paper completes the program by specifying the architecture explicitly, defining the evaluation protocol, and stating falsifiable predictions.

The term "zero-parameter" deserves precision at the outset. Every architecture contains numbers; the GON contains five fixed geometric constants (Proposition 2.3) and the coefficients of quaternion arithmetic. What it does not contain is any number set by fitting. A parameter, in the sense used throughout this paper, is a degree of freedom that an optimization procedure is permitted to move, and the GON has none: each constant is the unique value forced by a derivation, and perturbing any of them breaks the identity the architecture computes. There is no nearby configuration that is "almost as good"; correctness here is not a scalar to be maximized but a property that either holds or fails.

The corpus needs such an object for three reasons. First, an explicit architecture makes the identities of Paper 27 *executable*: a coding error, a wrong sign, or a misremembered constant becomes immediately visible as a nonzero residual. Second, the training–exposure distinction of Paper 28 needs a concrete exact graph as its experimental target; the GON supplies it. Third, downstream systems need an engine whose outputs carry a definite epistemic status: a fitted engine delivers estimates, the GON delivers evaluations of theorems.

The Geometric Observer Network is not a neural network in the conventional sense. It has no hidden layers to be trained, no activation functions to be chosen, no hyperparameters to be tuned. It is a *computational graph* that implements quaternion algebra and the Hopf map with fixed operations. Its relationship to standard ML architectures is analogous to the relationship between a hand-derived formula and a polynomial regression: both can produce outputs from inputs, but one is exact by construction while the other approximates.

Section 2 specifies the architecture layer by layer; Section 3 states the no-training property; Sections 4 and 5 define the protocol and the predictions; Sections 6 through 8 cover variants, implementation, and discussion. Open items confirmed as load-bearing are recorded in place, in the registry remarks and their status notes, rather than deferred to an appendix.

2 Architecture Specification

The GON is organized as six layers, evaluated by a single forward pass; the difference from a trained network lies entirely in what the layers do. Layers 1 through 4 form the core pipeline: integrate two observer states on S^3, form their relational quaternion, project through the Hopf map, and read off the lapse. Layers 5 and 6 are optional stages for diagnostics and density coupling. Each layer is stated with its operations and its parameter count, which is in every case zero.

2.1 Inputs

The GON accepts the following inputs at each evaluation:

(i) Initial observer states: $q_0^{(+)}, q_0^{(-)} \in S^3 \subset \mathbb{R}^4$ (8 real numbers, subject to 2 unit-norm constraints).

(ii) Generator schedules: $\Omega^{(+)}(t), \Omega^{(-)}(t) \in \mathbb{R}^3$ for $t \in [0, T]$, discretized at N time steps ($6N$ real numbers).

(iii) Integration parameters: step size h, horizon T.

2.2 Layer 1: Quaternion Evolution

For each observer $\sigma \in \{+, -\}$, the state is updated by the Lie group integrator:

$$q_{k+1}^{\sigma} = \exp\Bigl(\frac{h}{2}\,\Omega_k^{\sigma}\Bigr)\,q_k^{\sigma}, \tag{1}$$

where the quaternion exponential of a pure imaginary quaternion $\omega = \omega_x \mathbf{i} + \omega_y \mathbf{j} + \omega_z \mathbf{k}$ with $|\omega| = \phi$ is

$$\exp(\omega) = \cos\phi + \frac{\omega}{|\omega|}\sin\phi. \tag{2}$$

Remark 2.1. Using the exponential map (rather than Euler integration) preserves the unit-norm constraint $|q| = 1$ exactly at every step, eliminating drift. This is a structural choice, not a numerical convenience.

Parameters: None. The factor $\frac{1}{2}$ is fixed by the SU(2) Lie algebra structure.

Layer 1 is where the dynamics lives; everything downstream is observation. The integrator is a Lie group method in the sense of geometric numerical integration [8]: instead of stepping in the ambient $\mathbb{R}^4$ and renormalizing, each step multiplies the state by a group element, so the trajectory never leaves S^3 in exact arithmetic.

2.3 Layer 2: Relational Quaternion

$$q_{\mathrm{rel},k} = q_k^{(-)} \cdot \overline{q_k^{(+)}}. \tag{3}$$

Operations: Quaternion conjugation (negate imaginary part) and quaternion multiplication (bilinear, 16 multiply-adds).

Parameters: None.

The relational quaternion is the object the identities of Paper 27 are stated on. Everything downstream sees only q_{rel}, never the individual states: the architecture commits, structurally, to the position that the physically meaningful quantity is the relation between the observers rather than either observer alone.

2.4 Layer 3: Hopf Projection

From $q_{\mathrm{rel},k} = (a, b, c, d)$:

$$v_k = (a^2 + b^2) - (c^2 + d^2). \tag{4}$$

Operations: Four squarings, one addition, one subtraction.

Parameters: The coefficients $(+1, +1, -1, -1)$ are the Hopf structure, not trainable weights.

The quantity v_k is the Hopf latitude of the relational state: writing $a^2 + b^2 = \cos^2\beta$ and $c^2 + d^2 = \sin^2\beta$ for the Hopf angle β, equation (4) reads $v_k = \cos(2\beta)$. For a unit quaternion, $v_k \in [-1, 1]$ automatically. This is the quantity Paper 27 identifies with the velocity observable, computed by six arithmetic operations with hardwired signs.

2.5 Layer 4: Lapse Computation

$$m_k = \sqrt{1 - v_k^2}. \tag{5}$$

Operations: One squaring, one subtraction, one square root.

Parameters: None.

The lapse $m_k = \sqrt{1 - v_k^2}$ is the central output of the network. Because $v_k \in [-1, 1]$ for unit-norm input, the square root is well defined wherever Layer 3 is. Note what is absent: no series expansion, no small-velocity approximation, no fitted correction term. The relativistic form is computed exactly because it is an identity of the geometry, not a model of it.

2.6 Layer 5: Log Map and Axis-Resolved Observable (Optional)

For diagnostics and extended observables:

$$r_k = \arccos(a_k), \tag{6}$$

$$\hat{n}_k = \frac{(b_k, c_k, d_k)}{|(b_k, c_k, d_k)|}, \tag{7}$$

$$\eta_k = 2r_k, \tag{8}$$

$$n_{\parallel,k} = \hat{n}_k \cdot \hat{e}_{\mathrm{Hopf}}. \tag{9}$$

The axis-resolved identity then provides a redundant check:

$$v_k \stackrel{!}{=} n_{\parallel,k}^2 + (1 - n_{\parallel,k}^2)\cos\eta_k. \tag{10}$$

The redundancy of (10) is the point of the layer: the same observable v_k is computed along two independent arithmetic routes, and the routes must agree to floating-point precision at every step. A zero-parameter architecture cannot be validated by held-out data, since nothing was fitted; it is validated by internal identities of exactly this kind.

Remark 2.2 (TBS). The Riemannian logarithmic map on S^3 is singular at the antipodal point of any given base point and is not globally defined on all of S^3; all results depending on the global log map must be restricted to the cut locus complement or reformulated using an alternative atlas.

Status. Recorded in the registry as P030_1 (scope), confirmed by A295. The restriction to the cut locus complement is a domain hypothesis on the log-map observables, not a defect in the dynamics; corpus use is unaffected. Ledger: Paper 40.

Parameters: The Hopf axis $\hat{e}_{\mathrm{Hopf}} = (1, 0, 0)$ is determined by the complex-splitting convention, not by fitting.

2.7 Layer 6: Θ-Cycle and Monad Map (Optional)

For coupling to the density structure:

$$\dot{\Theta}_k = \Omega_0 \,\|\Omega_{\mathrm{rel},k}\|, \tag{11}$$

$$\Theta_k = \Theta_{k-1} + h\,\dot{\Theta}_k, \tag{12}$$

with $\Omega_0 = \pi^3/4$ and the monad closure at $\Omega_{\text{monad}} = 4\pi^3 + \pi^2 + \pi$.

Parameters: Ω_0 and Ω_{monad} are fixed constants derived in Paper 27 [4].

The Θ-cycle accumulates phase at a rate proportional to the magnitude of the relational generator, with the proportionality fixed by Ω_0. The closure constant $\Omega_{\text{monad}} = 4\pi^3 + \pi^2 + \pi$ is the expression derived from (B^4, S^3) geometry at the start of the corpus [1]. Both enter the layer as derived, to full precision.

2.8 Complete Forward Pass

$$\mathrm{GON}\big(q_0^{(\pm)},\ \Omega_{0:N}^{(\pm)}\big) \;=\; \big\{v_k,\ m_k,\ \eta_k,\ n_{\|,k},\ \Theta_k\big\}_{k=0}^{N}. \tag{13}$$

Proposition 2.3 (Total Parameter Count)**.** *The* GON *contains exactly zero trainable parameters and five fixed geometric constants:* $\frac{1}{2}$ *(Lie algebra factor),* $(+1, +1, -1, -1)$ *(Hopf structure),* $\hat{e}_{\text{Hopf}} = (1, 0, 0)$ *(splitting convention),* $\Omega_0 = \pi^3/4$ *(measure constant),* $\Omega_{\text{monad}} = 4\pi^3 + \pi^2 + \pi$ *(monad closure). All are derived, none are fitted.*

The proposition is checkable by inspection: the layer specifications above are the complete architecture. It is stated as a proposition rather than a remark because it is the load-bearing structural fact of the paper; every prediction in Section 5 rests on the network having exactly this inventory.

3 The No-Training Property

The GON is not merely untrained; it is an architecture for which training is the wrong operation. This section connects that statement to the result of Paper 28 [5].

Paper 28 established the training–exposure distinction: a computational graph that implements an exact identity sits at an isolated global minimum of the loss landscape, and gradient descent applied to such a graph generically degrades it. The intuition is direct. Optimization searches a neighborhood of configurations for one with lower loss. When the current configuration is exact, the loss is already zero on clean data and every neighboring configuration is strictly worse; any displacement, whether from gradient noise, regularization pressure, or a finite learning rate, moves the system off the identity. For a fitted model, nearby configurations are nearly as good, and training navigates among them productively. For an exact graph there is no "nearly."

The operational consequence is that the GON runs in exposure mode. Inputs vary; the architecture does not. A new generator schedule updates nothing inside the network; it selects a different trajectory through the same fixed graph, as evaluating a proven formula at a new argument selects a different value without amending the proof. Adaptation, where needed, belongs outside the geometric core (Section 6).

Two boundaries should be drawn honestly. Zero trainable parameters is a statement about the core architecture, not the full protocol, which still chooses a step size, a horizon, and comparator configurations. And the property does not by itself prove that training would degrade a parameterized copy of the GON; that is Conjecture 5.4, stated as a falsifiable prediction so that it is settled by experiment rather than by appeal to Paper 28.

4 Benchmarking Protocol

The protocol is not a horse race between architectures. Its purpose is to operationalize the training–exposure distinction as a reproducible experiment: a task on which the GON is exact by construction, a trained comparator of standard design, and metrics under which the difference between exactness and approximation is visible rather than averaged away.

4.1 Task Definition

Definition 4.1 (Lapse Prediction Task)**.** Given a generator schedule $\{\Omega_k^{(\pm)}\}_{k=0}^N$ and initial conditions $q_0^{(\pm)}$, predict the time series $\{m_k\}_{k=0}^N$ where $m_k = \sqrt{1 - v(q_{\mathrm{rel},k})^2}$.

4.2 Evaluation Scenarios

The following scenarios span the kinematic range, matching those used in the numerical experiments of Paper 27:

Scenario	**Description**	**Regime**
S1: Rest	$\Omega^{(+)} = -\Omega^{(-)}$, small	$v \approx 0$
S2: Constant v	Fixed generator ratio	$v = 0.6$
S3: Acceleration	Smooth ramp + oscillation	$v \in [0, 0.85]$
S4: Near-light	Large generator asymmetry	$v \to 0.999$

For each scenario, generate $M = 1000$ trajectories with random initial conditions $q_0^{(\pm)} \sim \mathrm{Uniform}(S^3)$ and $N = 2500$ time steps.

The scenarios are graded by difficulty for an approximator. S4 is the discriminating case: as $v \to 0.999$ the lapse becomes steep, small errors in v amplify into large errors in m, and an approximator must spend capacity on a region the exact computation handles with the same operations it uses everywhere else.

4.3 Comparator: Trained MLP

Definition 4.2 (Baseline MLP)**.** A multilayer perceptron with:

(i) Input: $(\Omega_k^{(+)}, \Omega_k^{(-)}, q_k^{(+)}, q_k^{(-)}) \in \mathbb{R}^{14}$.

(ii) Hidden layers: L layers of width W with ReLU activation.

(iii) Output: $\hat{m}_k \in \mathbb{R}$.

(iv) Training: MSE loss, Adam optimizer, 10^5 gradient steps.

Test configurations: $(L, W) \in \{(2, 64), (4, 128), (8, 256)\}$ to span capacity from small to large.

The comparator is deliberately conventional and trained generously, and it receives at each step the same per-step information the GON layers operate on. The contrast being tested is therefore not one of information but of structure: one system carries the geometry in its architecture, the other must recover it from samples.

4.4 Metrics

(i) **RMSE**: $\sqrt{\frac{1}{NM} \sum_{i,k} (m_{i,k} - \hat{m}_{i,k})^2}$.

(ii) **Max absolute error**: $\max_{i,k} |m_{i,k} - \hat{m}_{i,k}|$.

(iii) **Identity residual**: for the GON, compute $|v_k - [n_\parallel^2 + (1 - n_\parallel^2) \cos \eta_k]|$ as an internal consistency check.

(iv) **Norm drift**: $\max_k \big||q_{\mathrm{rel},k}| - 1\big|$ (should be $\sim 10^{-15}$ for exponential integrator).

5 Falsifiable Predictions

Three predictions follow, stated as conjectures because each is settled by running the protocol, not by argument. The first concerns the GON itself and would be falsified by any implementation error or wrong constant; the second concerns trained approximators; the third concerns what happens when the geometric constants are handed to an optimizer, and is the direct experimental form of the Paper 28 result.

Conjecture 5.1 (GON Exactness)**.** *The* GON *achieves:*

(a) RMSE $= 0$ *(up to floating-point arithmetic, i.e.,* $< 10^{-14}$*) on the lapse prediction task across all scenarios S1–S4.*

(b) Identity residual $< 10^{-13}$ *at every time step.*

(c) Norm drift $< 10^{-15}$*.*

Remark 5.2 (TBS). The claimed norm drift bound $< 10^{-15}$ is inconsistent with the $O(\varepsilon \cdot N)$ estimate derived from the internal error propagation analysis; one of these estimates must be in error, and a coherent bound with proof must be supplied.

Status. Recorded in the registry as P030_2 (confirmed-load-bearing), confirmed by A295. The inconsistency between the claimed $< 10^{-15}$ bound and the $O(\varepsilon \cdot N)$ estimate stands. It is directly testable against the GON implementation; that test has not yet been run, and the item remains open until it is. Ledger: Paper 40.

This is "falsifiable" in the sense that a coding error or incorrect constant would immediately produce nonzero error.

Conjecture 5.3 (MLP Residual Floor)**.** *For any MLP configuration* (L, W) *with finite capacity, there exists a scenario-dependent residual floor* $\epsilon(L, W) > 0$ *such that:*

(a) $\mathrm{RMSE}_{\mathrm{MLP}} \geq \epsilon(L, W) > 0$.

(b) ϵ *decreases with capacity but does not reach zero.*

(c) The MLP generalizes poorly to scenarios not in its training set (e.g., trained on S1–S3, tested on S4).

Conjecture 5.4 (Training Degradation)**.** *If the five geometric constants of the* GON *are promoted to trainable parameters and gradient descent is applied:*

(a) With clean data and zero-noise gradients: parameters remain at geometric values (zero gradient at minimum).

(b) With stochastic mini-batch noise ($B < NM$*): parameters drift from geometric values and RMSE increases monotonically with training steps.*

(c) With L_2 *regularization (*$\lambda > 0$*): the regularizer penalizes the geometric constants (which are* $O(\pi^3)$*, not small), pulling them toward zero and increasing error.*

6 Architecture Variants

Three variants extend the base architecture, in decreasing order of maturity: a design principle ready for use, a structured proposal, and a sketch whose central claim is recorded below as open.

6.1 Hybrid Architecture

In practical applications, the generator schedule $\Omega^{(\pm)}(t)$ may not be known a priori but must be inferred from raw sensor data. The hybrid architecture separates the system into:

(i) **Encoder** (trainable): maps raw input $x_{\text{raw}} \in \mathbb{R}^d$ to $(\Omega^{(\pm)}, q_0^{(\pm)})$. This is a standard neural network, trained by gradient descent.

(ii) **Geometric core** (fixed): the GON layers 1–6, with all constants hardcoded.

The design principle from Paper 28 applies: *train the unknown, freeze the proven.* The boundary between the two components is the boundary between what has been derived and what has not. The encoder faces a genuine unknown, the mapping from raw sensor data to geometric inputs, and training is the right tool for it. The core faces no unknown at all, and the hybrid design keeps the optimizer strictly on the encoder side of the interface.

6.2 Z_3 Extension

To incorporate the three-family structure from earlier papers [2, 3], the generator space is decomposed into three sectors with a cyclic coupling:

$$\Omega_k^{(\pm)} = \sum_{j=0}^{2} \omega^j \, \Omega_{k,j}^{(\pm)}, \qquad \omega = e^{2\pi i/3}. \tag{14}$$

This Hermitian cyclic structure was shown in the Z_3-phase experiments to produce triplet splitting in the eigenvalue spectrum. In the GON framework, it adds a discrete symmetry layer between the generator input and the evolution layer, still with no trainable parameters.

6.3 $J_3(\mathbb{O})$ Extension

The upstream Jordan algebra embedding from Paper 27 suggests a higher-dimensional variant where the state is a 3×3 Hermitian matrix over an associative subalgebra of $\mathbb{O}$, and the dual observers are extracted from off-diagonal entries. This extends the generator space from $\mathbb{R}^6$ to $\mathbb{R}^{27}$ and introduces the cubic invariant $\det_J(X)$ as an additional conserved quantity alongside the quadratic $\mathrm{Tr}(X^2)$.

The dimension cascade

$$J_3(\mathbb{O})\ (27) \ \to\ \mathbb{R}^6_\Omega \ \to\ S^3 \ \to\ S^2 \ \to\ v \ \to\ \sqrt{1-v^2} \tag{15}$$

maps directly onto a layered computational graph with progressively reducing dimensionality.

Remark 6.1 (TBS). The $J_3(\mathbb{O})$ extension claims that the cubic Jordan invariant $\det_J(X)$ is "an additional conserved quantity alongside $\mathrm{Tr}(X^2)$" in the GON evolution. This claim is not proved. Conserved quantities of a dynamical system are established by showing they commute with the flow (or satisfy a Noether argument); neither is provided here. The $J_3(\mathbb{O})$ extension modifies the generator space from $\mathbb{R}^6$ to $\mathbb{R}^{27}$ and involves the non-associative octonion algebra $\mathbb{O}$; the GON evolution equations (Layers 1–6, all defined for associative quaternion arithmetic) do not obviously extend to $J_3(\mathbb{O})$, and the extension is sketched rather than specified. The dimension cascade diagram above is an informal illustration, not a derived result.

Status. Recorded in the registry as P030_3_c (confirmed-load-bearing), confirmed by A295. The $J_3(\mathbb{O})$ conservation claim is structural and unverified: no flow-commutation or Noether argument has been supplied here or in any later paper. The extension should be read as a sketch, not a result. Ledger: Paper 40.

7 Implementation Notes

7.1 Numerical Precision

The exponential-map integrator (1) preserves $|q| = 1$ exactly in exact arithmetic. In floating-point, drift accumulates at rate $O(\epsilon_{\text{mach}} \cdot N)$. For $N = 2500$ steps in double precision ($\epsilon_{\text{mach}} \approx 10^{-16}$), expected drift is $\sim 10^{-13}$, consistent with the Hopf metric identity residuals observed in Paper 27.

This estimate is in tension with the $< 10^{-15}$ drift bound stated in Conjecture 5.1: the two cannot both be right as written. The inconsistency is recorded as P030_2 (Remark 5.2) and is directly testable against the implementation; the item remains open until that test is run. Neither number is quietly adjusted here, because which estimate is in error is precisely what the test will determine.

7.2 Computational Cost

Per time step, the GON requires:

Operation	**FLOPs**
Quaternion exponential ($\times 2$ observers)	~ 40
Quaternion multiplication ($\times 2$ evolutions)	~ 32
Conjugation + relational product	~ 20
Hopf projection	~ 8
Lapse computation	~ 4
Total per step	~ 104

For $N = 2500$ steps: $\sim 2.6 \times 10^5$ FLOPs total. This is orders of magnitude cheaper than a single forward pass through even a small MLP on the same input sequence.

7.3 Reproducibility

The architecture is fully deterministic given inputs. No random initialization, no stochastic training, no hyperparameter search. Two implementations of the GON that correctly implement quaternion arithmetic will produce bitwise-identical outputs (in the same floating point environment).

8 Discussion

8.1 What the GON Is Not

The GON is not a universal function approximator. It computes a specific set of geometric identities. It cannot learn new functions, adapt to distribution shift, or generalize beyond its mathematical domain. It is a *calculator*, not a *learner*.

This is precisely its strength. Within its domain, it is exact. Outside its domain, it is silent: it produces no output rather than a wrong output. The failure mode of the GON is "not applicable," never "approximately wrong."

Silence is the property that universal approximators structurally lack. A trained network always returns an answer; far from its training distribution it extrapolates, and the answer degrades without announcing that it has degraded. The GON's contract is the opposite: applicability is decided by whether the input lies in the mathematical domain of the identities, before evaluation, not graded afterward by how plausible the output looks. Two caveats attach. The domain has genuine boundaries, of which the log-map restriction P030_1 (Remark 2.2) is the concrete instance here. And silence is a specification discipline, not a mechanism the formulas enforce by themselves; an implementation must validate its domain at the boundary, since the arithmetic will evaluate whatever it is handed.

8.2 What the GON Demonstrates

The existence of a zero-parameter architecture that exactly computes relativistic phase dynamics demonstrates that:

(i) The identities are computationally real, not merely algebraic curiosities.

(ii) The "unreasonable effectiveness of mathematics" has a computational counterpart: proven identities can be hardcoded, and the resulting system outperforms any trained approximation within the identity's domain.

(iii) The training–exposure distinction from Paper 28 has concrete architectural consequences.

The first point deserves emphasis: an identity wired into an executable graph is checked every time the graph runs, along redundant routes (equation (10)). The GON turns the corpus's central kinematic results from claims that must be trusted into computations that can be watched.

8.3 Position in the Corpus

The GON is the executable terminal of a three-paper arc. Paper 27 [4] derived the identities: the relational quaternion, the Hopf observable, the emergent Lorentz lapse, the measure constants. Paper 28 [5] established what those identities imply for computation. This paper turns both into an artifact. Its constants reach further back: the monad closure $4\pi^3 + \pi^2 + \pi$ is the expression of [1], and the Z_3 and $J_3(\mathbb{O})$ variants connect to the three-family and master-operator structure of [2, 3]. The architecture adds no new physics; its contribution is that the physics already derived can now be run.

8.4 Relationship to Geometric Deep Learning

The GON shares the *philosophy* of geometric deep learning [6], encoding symmetries and invariances into architecture rather than learning them from data, but takes it to its logical endpoint. In geometric deep learning, equivariance is a *constraint on trainable layers* (e.g., group-equivariant convolutions [7]). In the GON, the geometry is the *entire computation*, and there is nothing left to train. The progression is natural: as more of a problem's structure is known exactly, the trainable fraction of the architecture shrinks. The GON is the limit point of that progression for a problem whose structure is known completely, and the hybrid architecture of Section 6 shows that the limit point composes with conventional practice rather than replacing it.

9 Conclusion

We have specified the Geometric Observer Network: a zero-parameter computational architecture whose forward pass is the physics of dual-observer phase dynamics on S^3. The architecture is:

(i) **Exact:** it computes proven identities, not approximations.

(ii) **Parameter-free:** every constant is derived from geometry.

(iii) **Efficient:** ~ 100 FLOPs per time step.

(iv) **Falsifiable:** three concrete predictions distinguish it from trained alternatives.

(v) **Extensible:** hybrid, Z_3, and $J_3(\mathbb{O})$ variants are structurally defined.

Three items remain open and are recorded in place. P030_1 (Remark 2.2) is a scope restriction: the log-map observables carry a cut locus complement domain hypothesis. P030_2 (Remark 5.2) is a confirmed-load-bearing inconsistency between the stated norm-drift bound and the error-propagation estimate, testable and untested. P030_3_c (Remark 6.1) is the unverified conservation claim of the $J_3(\mathbb{O})$ sketch. None of the three touches the core pipeline of Layers 1 through 4; all three bound what may be claimed beyond it.

The GON is not a model of physics. It is a direct computational instantiation of the geometric identities proven in Papers 27–28. Its outputs are not predictions; they are theorems, evaluated.

References

[1] L. F. Vlegels, *The Perfect Stable Sphere: Deriving α^{-1} from (B^4, S^3) Geometry*, This volume (2025).

[2] L. F. Vlegels, *The Fermion Sector from S^3 Topology and Z_3 Symmetry*, This volume (2025).

[3] L. F. Vlegels, *The Master Operator: Assembling the Geometric Spectral Engine*, This volume (2025).

[4] L. F. Vlegels, *Universal Wave Geometry: Emergent Lorentz Structure from Dual-Observer Phase Dynamics on S^3*, This volume (2025).

[5] L. F. Vlegels, *Geometric Exactness and the Training–Exposure Distinction*, This volume (2025).

[6] M. M. Bronstein, J. Bruna, T. Cohen, and P. Veličković, “Geometric deep learning: grids, groups, graphs, geodesics, and gauges,” *arXiv:2104.13478* (2021).

[7] T. S. Cohen and M. Welling, “Group equivariant convolutional networks,” *Proc. ICML* (2016), 2990–2999.

[8] E. Hairer, C. Lubich, and G. Wanner, *Geometric Numerical Integration*, Springer (2006).

[9] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press (2016).