Proteus: Adaptive Scale-Space Analysis for Learning Non-Linear Fuzzy Manifold Memberships (*Draft*)

Franz Wollang
Independent Researcher

Dated: 2025-??-??

DRAFT — NOT FOR CITATION

This is a preliminary working version posted for discussion and feedback. Content may change significantly before formal submission.

Abstract

We present Proteus, a novel, multi-stage framework for discovering and modeling the structure of complex, high-dimensional data at multiple scales. Traditional methods for manifold learning often force a trade-off between computational scalability and the theoretical rigor of the final model. Proteus resolves this by decomposing the problem into two distinct stages. Stage 1 performs a principled, multi-scale search grounded in scale-space theory, using a fast Growing Neural Gas (GNG) to discover the characteristic scales of the data manifold and produce a coarse geometric "scaffold." Stage 2 uses this scaffold to initialize a high-fidelity refinement process at each discovered scale. This stage builds a full simplicial complex and uses a novel geometric audit mechanism, the Torsion Ladder, to measure and correct local curvature errors in the manifold approximation. The final result is a generative model with a simplicial piecewise-linear core (optionally composed with invertible non-linear warps), complete with a continuous probability gradient field derived from a "dual flow" of probability across the complex. This allows for high-fidelity data synthesis, robust probability density queries, and a deeply interpretable representation of the data's structure.

1 Introduction

Real-world data rarely conforms to simple structures. It often exists on a manifold with varying intrinsic dimensionality, significant non-linearity, and meaningful structures at multiple, nested scales. A comprehensive understanding of such data requires a system that can:

1. Efficiently and robustly discover the "natural" scales of the data without prior knowledge.
2. Explicitly model non-linear geometry and variable dimensionality.
3. Produce a final model that is not just descriptive, but generative, interpretable, and provides a rich basis for downstream tasks.

Proteus is designed to be such a system. Its two-stage architecture is built to resolve the classic trade-off between speed and fidelity. The first stage is a fast exploratory engine that performs a principled search for the data's characteristic scales, delivering a structural "scaffold." The second stage is a powerful refinement engine that takes this scaffold and builds a high-fidelity, generative model. This is achieved by introducing a simplicial complex representation and a novel

geometric auditing process based on a discrete measure of torsion, which allows the system to identify and correct failures in its own piecewise-linear approximation of the data manifold. This paper describes the architecture of the Proteus learning engine, from its theoretical grounding in scale-space analysis to the specific mechanisms of the Torsion Ladder. We will conclude with an empirical analysis of the resulting model's topological and generative quality.

1.1 Related Work

Proteus intersects manifold learning, scale-space analysis, discrete geometry, and generative modeling. Classical manifold learning methods (e.g., Isomap, LLE, t-SNE, UMAP) provide embeddings but not generative, scale-aware manifold models. Scale-space theory (Lindeberg) motivates our normalized response and grid design in $\log \tau$. Discrete exterior calculus and vector calculus underpin our torsion 2-form and the dual-flow divergence constraints. Persistent homology offers topology diagnostics complementary to our simplicial audit. Generative models (VAEs, GANs, normalizing flows) inform our warp components; we use flows minimally, as upgrades when geometry alone cannot resolve curvature. Our two-stage, scaffold-then-refine architecture differs by separating fast scale discovery from high-fidelity generative modeling at a fixed characteristic scale.

2 The Proteus Framework

The Proteus framework is built upon a heavily augmented Growing Neural Gas (GNG) architecture, a form of competitive Hebbian learning. In the limit, such Hebbian procedures produce a maximum-entropy tiling of the underlying probability density function (PDF), where each element of the tiling—typically a node's Voronoi cell—contains an equal share of probability mass. Proteus is founded on this same max-ent principle. Its novelty lies in a series of augmentations for robustness and multi-scale analysis: it replaces heuristic error counters with first-principles statistical tracking (local moments, variance, and incoherence), uses data-dependent growth thresholds derived from scale-space theory, employs a recursive architecture for hierarchical decomposition, and upgrades the geometric representation to a full simplicial complex in its refinement stage. Despite these modifications, the fundamental goal remains to learn a mesh that faithfully represents the data's PDF through a maximum-entropy partition.

The GNG learning process is grounded in a physical analogy of thermal equilibration. Each node maintains robust statistical trackers for the first two moments of its local error distribution: the mean ($\mathbf{m}_i$) and the squared error ($\mathbf{s}_i$). These are not mere heuristics; they have direct physical interpretations where $\mathbf{m}_i$ represents a **local heat current** (a measure of coherent drift) and the derived variance σ_i^2 acts as **local temperature**. The entire mesh is considered to have converged at a given scale only when it reaches **thermal equilibrium**—a state where these heat currents are spatially uniform, measured by the coefficient of variation of neighbor-normalized incoherence scores falling below a small threshold (CV ¡ 0.01).

2.1 Stage 1: Fast Exploration and Scaffolding

The goal of this stage is a fast and scalable implementation of the mass-partition principle using a highly-optimized Growing Neural Gas (GNG) architecture. The fundamental element is the node's Voronoi cell, with a variance-based splitting rule ($\sigma_i^2 > \tau_{local,i}$) acting as an efficient proxy for detecting cells with excess probability mass. This process is coupled with a Bayesian optimizer in a meta-learning loop to discover the characteristic scales at which the partitioned structure is most salient.

The GNG learning process is grounded in robust statistical tracking. Each node i maintains exponential moving averages of the first two moments of its local error distribution: the mean ($\mathbf{m}_i$) and squared error ($\mathbf{s}_i$), from which local variance (σ_i^2) and signal incoherence (ρ_i) are derived.

Additionally, each node tracks the principal error direction ($\mathbf{u}_i$) via Oja's rule, providing a principled, data-driven direction for topological growth. The EWMA weight is set to $\alpha = \ln 2/k$, tying the estimator memory to the neighborhood size (derivation in SI S2.2).

The algorithm processes data points by identifying the k nearest nodes and distributing the learning signal using rank-ordered geometric weights $(1/2, 1/4, 1/8, \ldots)$. This weighted distribution acts as a discrete exponential neighborhood kernel while enabling fractional statistics for robust pruning via Wilson bounds. Node topology evolves through principled splitting (when $\sigma_i^2 > \tau_{local,i}$, with new nodes positioned using the principal error direction) and statistical pruning, with mandatory linearity checks (GLCE and distance correlation tests) to handle potential non-linearities. The process continues until thermal equilibrium is reached, detected when the coefficient of variation of neighbor-normalized incoherence scores falls below 0.01. Complete derivations for the dual-rate motion scheme (η_{GNG} and η_{cent}) are provided in SI S2.3, with pruning details in SI S3.

The per-sample computational cost is $\mathcal{O}(k\,d)$ for moment updates, with expensive $\mathcal{O}(d^2)$ operations occurring only on deferred moves/splits. ANN queries average $\tilde{\mathcal{O}}(\log N)$ with HNSW. Complete algorithmic details and pseudocode are provided in SI S11.

2.1.1 Principled Scale Representation

We employ a four-tier representation of scale to decouple the optimizer's search from the GNG's local, adaptive physics (see SI S2.4 for the control-to-threshold mapping).

1. **Control Parameter ($s_{control}$):** A normalized parameter $\in [0, 1)$ proposed by the optimizer.

2. **Subspace Dimensionality ($D_{subspace}$):** The dimensionality of the current sub-problem.

3. **Global Growth Threshold (τ_{global}):** A physical scale derived from the control parameter via $\tau_{global} = -D_{subspace} \log(1 - s_{control})$, where $D_{subspace}$ is the dimensionality of the current problem. This global threshold is used to evaluate the overall quality of the learned structure at a given scale.

4. **Local Growth Threshold ($\tau_{local,i}$):** A per-node threshold, $\tau_{local,i} = -d_{final,i} \log(1 - s_{control})$, where $d_{final,i}$ is the smoothed, local intrinsic dimension at node i. This allows the GNG to adapt its growth conditions to regions of varying manifold dimensionality.

2.1.2 Scale-Space Grid, Optimizer, and Recursive Search

Classical scale-space theory studies how structures emerge and dissolve under Gaussian smoothing at scale σ, using scale-normalized operators to make responses comparable across scales. In Proteus we learn a mesh whose growth is governed by a variance cap τ that plays the role of the smoothing scale (with $\tau{=}\sigma^2$). The structural signal is obtained from rank-geometric fractional hits and a local k-NN volume proxy; under local stationarity this behaves like a KDE up to a constant. With a one-time calibration $\sigma = \sqrt{\tau}/c_{d,k}$ and choosing $\gamma{=}1$ for scale-covariance, the normalized-response principle transfers verbatim: maxima of $\Phi(\log \tau)$ identify characteristic scales.

We seek to maximize a γ-normalized scale-space response $\Phi(\tau)$ over a geometric grid of thresholds $\{\tau_i\}$. The grid spacing follows the scale-space resolution limit: adjacent samples in $\log \sigma$ are separated by $\frac{1}{2}$ of the FWHM of a normalized Gaussian response, yielding a geometric ratio $r \approx 1/\sqrt{2} \approx 0.71$ (derivation in SI S2.1). At each τ_i, the Stage 1 loop is run to thermal equilibrium (CV< 0.01), and $\Phi(\tau_i)$ is recorded. Local maxima are bracketed by sign flips of the second difference in $\log \tau$, then refined with a Gaussian-Process Bayesian optimizer restricted to the bracket to obtain τ^*.

This procedure recurses on data partitions induced by the converged mesh at τ^*, discovering nested structure only where sufficient routed data exists (minimum sample threshold as in SI S9). The high-level control is summarized below.

Algorithm 1 Recursive Scale-Space Search (meta-level controller)

procedure FINDSCALES(X)
  Choose geometric grid $\mathcal{T} = \{\tau_0 r^j\}_{j=0}^{J}$ with $r{=}1/\sqrt{2}$
  for each $\tau \in \mathcal{T}$ **do**
    Initialize mesh $\mathcal{M} \leftarrow \varnothing$; **repeat** $\mathcal{M} \leftarrow$ UpdateWithSample($\mathbf{x}, \mathcal{M}, \tau$) until CV$< 0.01$
    Record response $\Phi(\tau)$
  end for
  $\mathcal{B} \leftarrow$ BracketsFromSecondDifference($\Phi, \mathcal{T}$)
  $\tau^* \leftarrow$ BayesianOptimize($\Phi, \mathcal{B}$)
  $\mathcal{M}^* \leftarrow$ RunStage1ToEquilibrium(X, τ^*); $\{X_k\} \leftarrow$ PartitionByMesh($X, \mathcal{M}^*$)
  for each partition X_k **do**
    if $|X_k| \geq n_{min}$ **then**
      FINDSCALES(X_k) ▷ Recurse on sufficiently large sub-regions
    end if
  end for
end procedure

The result of Stage 1 at a single scale is a scaffold: a node–edge graph that captures the manifold structure at the chosen threshold. The full hierarchical scaffold is obtained by recursively re-applying Stage 1 to the sub-regions identified by the scale-space search (recursion described below).

2.2 Stage 2: High-Fidelity Refinement and Generative Modeling

The second stage of the Proteus framework transitions from fast exploration to high-fidelity refinement. It takes the geometric scaffold and characteristic scale τ^* provided by Stage 1 and constructs a rigorous, generative model of the data manifold. This transition is founded on a key architectural upgrade: the representation is lifted from a node-edge graph to a full simplicial complex. This change elevates the framework's core mass-partitioning principle, shifting the fundamental element from a node's Voronoi cell to the simplex volume. The learning objective thus becomes the attainment of simplex equilibrium, a state where each simplex contains an equal share of probability mass. This move to a simplicial structure, which can be viewed as the dual of the Stage 1 Voronoi tessellation, also provides the natural setting for subsequent generative modeling; the dual graph of face-adjacent simplices is where the conservation laws governing probability flow are enforced. Beyond equilibrium, the mismatch between simplex- and node-level objectives produces a stable, multi-faceted statistical signature at structural junctions (see SI S12.2), strengthening the model's interpretability claims.

Stage 2 operates as a single, intensive refinement pass at the fixed scale τ^*. It begins by warm-starting from the converged Stage 1 results: node positions and statistics are carried forward, with EWMA moments shrunk by a decay factor $\gamma_0 \approx 0.7$ to accommodate the richer neighborhood model ($k \approx d$ neighbors instead of $k = 8$), while hit and link counts are preserved unchanged (see SI S9). The nudge accumulator $\mathbf{a}$ is reset to zero to ensure stable Stage 2 dynamics. An initial simplicial complex is then constructed using a fast, quadratic-cost greedy chaining heuristic that discovers the majority of d-simplices from the existing node-edge graph. The learning algorithm subsequently runs on this dynamic complex, employing simplex-native updates where each data point identifies its containing simplex as the exclusive neighborhood for statistical updates, with vertex updates weighted by rank-ordered distance within the simplex. Throughout this process, link creation proceeds GNG-style between the two best-matching

units (BMU_1, BMU_2), and the framework continuously grows and refines the complex topology, healing gaps through automatic simplex discovery whenever new links complete $d+1$ cliques.

A simplicial complex, however, provides only a piecewise-linear (PL) approximation of the manifold, which is insufficient for data with significant curvature. To ensure geometric fidelity, Stage 2 introduces a novel auditing mechanism, the torsion ladder, which provides a discrete measure of local curvature and triggers a principled response to correct for it. To make the model generative, a dual flow mechanism tracks the net flow of probability across simplex faces, allowing for the reconstruction of a continuous probability gradient field over the entire manifold. This gradient field is essential for high-fidelity data synthesis and robust density queries. In cases of extreme curvature where geometric corrections are insufficient, this PL core can be composed with invertible non-linear warps; the representation remains piecewise-linear in the warped coordinates, while in the original data space it is the pullback of this core by the learned warp.

2.2.1 The Learning Objective: Simplex Equilibrium

The learning rule in Stage 2 is based on a physical analogy of energy partition, which directly optimizes for simplex equilibrium, where the probability mass contained within each simplex is approximately equal. The error from a data point is treated as an "impulsive force." The system's response is determined by its local rigidity: in *plastic* regions (low rigidity), error acts like kinetic energy and results in geometric nudges; in *rigid* regions (high rigidity), error accumulates as potential stress that is resolved via topological change (splits or local warps).

Define Node Equilibrium as the state where the probability mass in each node's Voronoi cell is approximately equal, $P(V_k) \approx 1/N$ for N nodes. In contrast, Simplex Equilibrium requires equal mass in each simplex, $P(C_i) \approx 1/M$ for M simplices. While not identical, optimizing for Simplex Equilibrium drives the system toward Node Equilibrium via a negative feedback loop. Consider a node i with excess mass $P(V_i) \gg 1/N$. This biases data toward w_i, inflating structural stress in adjacent simplices S_i. The accumulated error vector E_{C_k} grows radially away from w_i, triggering a split that inserts a new node j, redistributing mass from V_i to V_j. Divergences occur at dimensionality junctions (e.g., 1D filament meeting 2D sheet), where bimodal activity and asymmetric links signal complex topology. This "informative mismatch" emerges naturally, providing robust signatures for local structure. A concise correspondence argument, showing that simplex equilibrium induces effective node equilibrium under mild regularity (local stationarity and nondegenerate simplex stars), is provided in SI S12.2. The proof sketches how persistent node over-mass necessarily produces simplex-level stresses whose resolution reduces Voronoi imbalance.

2.2.2 Geometric Audit: Torsion Formalism and the Torsion Ladder

The core innovation of Stage 2 is a rigorous geometric audit designed to address the limitations of the piecewise-linear approximation on a curved manifold. This audit is based on the concept of **torsion**. In our framework, the local residual mean field, m, should ideally be conservative (irrotational). Any rotational component in this field indicates a geometric mis-fit that cannot be resolved by simple node translation, signaling a failure in the linear approximation.

We formalize this concept with a discrete 2-form. For a d-simplex $S = \{v_0, \ldots, v_d\}$ with vertex positions $\{\mathbf{w}_{v_i}\}$ and residual means $\{\mathbf{m}_{v_i}\}$, we define edge matrices relative to the barycenter: $E = [\,\mathbf{w}_{v_1} - \bar{\mathbf{w}} \;\cdots\; \mathbf{w}_{v_d} - \bar{\mathbf{w}}\,]$ and $M = [\,\mathbf{m}_{v_1} - \bar{\mathbf{m}} \;\cdots\; \mathbf{m}_{v_d} - \bar{\mathbf{m}}\,]$. The torsion tensor is then given by:

$$\Omega_S \;=\; M^\top E - E^\top M\,.$$

This **simplex 2-form** Ω_S is the direct discrete analogue of the curl ($\nabla \times m$). Its Frobenius norm, $\kappa_S = \|\Omega_S\|_F$, provides a single scalar value for the "torsional stress" within simplex S. A

Placeholder schematic: Torsion Ladder triage with thresholds on $R_S = \kappa_S/\tau^*$ and actions (keep, split, mini-NSF).

Figure 1: Torsion Ladder triage with thresholds on $R_S = \kappa_S/\tau^*$ and resulting actions (keep, split, mini-NSF).

high value of κ_S indicates that the piecewise-linear approximation is failing and that a corrective intervention is necessary.

This audit is operationalized by the **Torsion Ladder**, a principled triage system that decides how to act based on the ratio of torsional stress to the scale threshold, $R_S = \kappa_S/\tau^*$. Ratios $R_S < 0.05$ are treated as numerical noise and ignored. For $0.05 \leq R_S < 0.30$, curvature is tolerable and the simplex is kept as-is. For intermediate stress levels ($0.30 \leq R_S < 0.60$), the system concludes that the misfit is significant enough to be resolved geometrically. It performs a torsion-aligned split, inserting a new node at the midpoint of the simplex's longest edge projected onto the principal axis of the torsion tensor Ω_S, directly resolving the primary rotational stress. When the local curvature is too strong for a geometric fix ($R_S \geq 0.60$), the ladder escalates to a more powerful intervention, attaching a small, local Normalizing Spline Flow (a "mini-NSF") to the simplex patch to explicitly learn the non-linear transformation.

The geometric correction of a torsion-aligned split must not degrade the numerical stability of the mesh. To guard against this, any proposed split is evaluated using a simplex shape metric, $Q_S = d\, r_{\text{in}}/R_{\text{circ}} \in (0, 1]$, where r_{in} and R_{circ} are the inradius and circumradius. If any child simplex created by a split has poor shape quality ($Q_S < 0.25$), the split is rejected, and the system falls back to a more conservative centroid insertion (a "star split"). This safeguard ensures that all elements in the mesh maintain a high quality, which is crucial for stable face normal calculations and the overall integrity of the model (details in SI S3).

In Stage 2, node pruning is arbitrated by the incident simplices: deletions require a formal vote across the simplex star of a node, with a geometric veto (reconstruction error test) to protect structurally critical "tent-pole" nodes (see SI S11).

2.2.3 Generative Modeling with the Dual Flow

The framework's generative capability is enabled by the tracking of a **dual flow** of probability across the faces of the simplices. The error vector from each data point is projected onto the face normals of its containing simplex, creating a measure of "pressure" on each face. By tracking the net pressure (flux) on every face and enforcing a global conservation law, the system reconstructs the full, continuous probability gradient field within each simplex. By the Divergence Theorem, this is sufficient to define a valid generative model.

For a winning simplex S and sample $\mathbf{x}$, a fractional face pressure is distributed to each facet $f \in \partial S$ proportional to the projection of the residual onto the outward unit normal $\mathbf{n}_f$:

$$\Delta p_f \propto \max\left(0,\, (\mathbf{x} - \bar{\mathbf{w}}_S)^\top \mathbf{n}_f\right); \qquad p_f \leftarrow p_f + \Delta p_f\,.$$

Let $\mathcal{G}_{\text{dual}}$ be the dual graph whose nodes are faces and whose edges connect adjacent faces. Each primal simplex induces a local flow-consistency constraint on this graph. We seek face pressures $\{p_f\}$ that satisfy discrete conservation (no net source or sink within each simplex, up to sampling noise) while staying close to the accumulated evidence. This yields a quadratic objective over the dual graph:

$$\min_{\{p_f\}} \sum_f \lambda\,(p_f - \hat{p}_f)^2 \;+\; \sum_S \mu\,\left\|A_S\,\mathbf{p}_S\right\|_2^2,$$

where $\hat{p}_f$ are the empirical tallies, $\mathbf{p}_S$ is the vector of face pressures incident to simplex S, and A_S is a small stencil enforcing local conservation. Belief Propagation (or Gauss–Seidel on the normal equations) over $\mathcal{G}_{\text{dual}}$ provides an efficient solver, propagating constraints until convergence. The

resulting $\{p_f\}$ define a piecewise-linear probability gradient field consistent with the Divergence Theorem. (Details in SI S4.)

2.2.4 Out-of-Sample Queries and Sampling

Given a query $\mathbf{x}$, locate the containing simplex (or nearest via barycentric projection), compute barycentric coordinates, and evaluate density from face pressures and the piecewise-linear gradient field (see SI S13). For generation, sample a simplex proportional to its mass, draw barycentric coordinates, and map to data space; apply local warps if attached. This enables likelihood queries, anomaly scoring, and synthesis using the same dual-flow field used in training. Boundary faces use Neumann-like treatment (no exterior flux) as in the SI. The procedure is linear in simplex dimension and reuses cached factors.

For evaluation, we report generation metrics (MMD) and log-likelihood as detailed in Section 3 and SI S5.

2.2.5 Non-Linear Warp Strategy (Global vs. Patch-wise)

When residual curvature persists after geometric refinement, Proteus applies a minimal non-linear warp based on the torsion coverage P_κ (the fraction of simplices with $R_S \geq 0.30$). If curvature is widespread ($P_\kappa > 50\%$), a shallow global Glow is trained, using invertible 1×1 convolutions to mix dimensions efficiently. If curvature is patchy ($P_\kappa \leq 25\%$), compact mini-NSFs (2-3 layers of rational-quadratic spline couplings) are attached only to the highest-torsion patches, identified via Leiden community detection on a facet-adjacency graph. In the intermediate regime ($25\% < P_\kappa \leq 50\%$), a moderate global Glow is combined with mini-NSFs on the worst patches. If a mini-NSF fails to reduce R_S below threshold, a torsion-aligned split is attempted. Complete details on patch identification, parameter budgets, and training procedures are in SI S7.

3 Model Quality Analysis

To validate the claims of the Proteus framework, we conduct a series of experiments designed to measure the fidelity of the generated model, rather than its performance on a specific downstream task.

3.1 Experimental Setup

3.1.1 Datasets

We use a suite of synthetic datasets with known ground-truth topologies, including nested spheres of different dimensions, linked tori, and manifolds with variable density and dimensionality junctions. We also use benchmark real-world datasets to demonstrate performance on complex, unknown structures.

3.1.2 Evaluation Metrics

Our analysis focuses on metrics that directly assess the quality of the learned manifold model:

- **Topological Accuracy:** We compare the Betti numbers (characterizing holes of different dimensions) of the learned simplicial complex to the ground truth of the synthetic datasets using persistent homology.
- **Reconstruction Error:** We measure the average distance between original data points and their projection onto the learned manifold.

- **Log-Likelihood:** As Proteus produces a generative model, we evaluate the average log-likelihood of held-out test data under the model. This provides a direct measure of its predictive power, a capability absent in non-generative manifold learning methods.
- **Generative Quality:** We draw samples from the learned model and compare their distribution to the original data's distribution using metrics such as Maximum Mean Discrepancy (MMD) to assess the fidelity of the generative process.

We follow standardized visualization and protocol defaults detailed in SI S15 (visuals) and SI S5 (evaluation settings).

3.2 Comparative Analysis

We will present results (tables, visualizations) comparing the Proteus model's performance on the above metrics to models produced by other state-of-the-art algorithms, such as UMAP (for embedding quality) and a Variational Autoencoder (for generative quality).

3.2.1 Protocol Details

- **Datasets:** Synthetic: nested spheres, linked tori, variable-density manifolds; Real-world: standard benchmarks (e.g., MNIST subset, tabular with mixed scales).
- **Splits & seeds:** 80/20 train/test; 5 seeds per experiment; report mean $\pm$std.
- **Scales grid:** Geometric grid in τ: $\tau_i = \tau_0\, r^i$, r=0.71; 7–9 points per decade; GP refinement within first bracketed maximum.
- **Stopping:** Stage 1: CV<0.01; Stage 2: CV<0.01 and R_S<0.30 for all simplices.
- **Persistent Homology:** Vietoris–Rips with max dim ≤ 2; filtration up to $1.5\,\sigma_\star$; bottleneck/Wasserstein distances to truth.
- **MMD:** RBF kernel with bandwidth median heuristic; 1 000 samples; 1 000 test points.
- **Ablations:** (i) No Stage 1 scaffold; (ii) No Torsion Ladder; (iii) No Dual Flow.

3.3 Ablation Studies

We conduct targeted ablation studies to demonstrate the necessity of the core components of the Proteus architecture:

- **Ablating Stage 1 Scaffolding:** We run the Stage 2 refinement process from a random initialization instead of the Stage 1 scaffold. We expect to show that this leads to significantly slower convergence, higher final reconstruction error, and frequent failure to identify the correct global topology, demonstrating the critical importance of the coarse-to-fine, scale-aware approach.
- **Ablating the Torsion Audit:** We disable the Torsion Ladder, forcing the model to rely only on variance-based splitting. On datasets with non-trivial curvature, we expect the model to produce topologically incorrect structures and exhibit high reconstruction error in curved regions, validating the need for a direct geometric audit.
- **Ablating the Dual Flow:** We disable the dual flow mechanism and evaluate the model's generative capabilities. We expect to show that without the explicit modeling of the probability gradient, the model's ability to produce high-fidelity samples and provide accurate density estimates is severely degraded.

Table 1: Placeholder results: topological accuracy (PH distance), reconstruction error, log-likelihood, and MMD (lower is better where applicable).

Method	PH Dist.	Recon Err	Log-Lik	MMD
Proteus (full)	–	–	–	–
No Stage 1	–	–	–	–
No Ladder	–	–	–	–
No Dual Flow	–	–	–	–
Baseline (UMAP+KDE)	–	–	–	–
VAE	–	–	–	–

Placeholder ablation plots: effects of removing Stage 1 scaffold, Torsion Ladder, or Dual Flow on PH distance, reconstruction error, and MMD across datasets.

Figure 2: Ablations (placeholders): effects of removing Stage 1 scaffold, Torsion Ladder, or Dual Flow on PH distance, reconstruction error, and MMD across datasets.

4 Limitations and Complexity/Scaling

Limitations: (i) Extremely high ambient dimensions without an initial projection can inflate ANN and per-sample costs; (ii) Highly anisotropic densities may require a deeper grid in τ to separate overlapping scales; (iii) In very sparse data, torsion estimates on large simplices can be noisy—guarded by our Q_S fallback and minimum-sample thresholds but still a practical consideration.

Typical envelopes (indicative):

- $N \in [10^5, 10^7]$, $d \in [64, 2048]$ after optional projection; per-level wall-time dominated by HNSW queries and moment updates.
- HNSW: $M \in [16, 32]$, `efConstruction`$\in [100, 400]$, `efSearch`$\in [50, 200]$; increase `efSearch` near convergence for stability.
- Memory: nodes+links+simplexes typically within $\sim 10\%$ growth per recursion level due to Ladder and prune rules.

Asymptotic envelopes: Stage 1 per-epoch time is $\mathcal{O}\big(N\, d\, k\, \log N_{\text{nodes}}\big)$ with HNSW queries; topology/pruning is amortized. Space is $\mathcal{O}(N_{\text{nodes}}\, d + N_{\text{nodes}}\, \bar{d})$, independent of N. Stage 2 seeding via Greedy Chaining costs $\mathcal{O}\big(N_c\, d_{\text{cand}}^2\, d_c\big)$; periodic torsion audits cost $\mathcal{O}\big(N_{\text{simplices}}\, (d{+}1)^2\, d\big)$. Dual-flow belief propagation operates on the dual graph of faces of size $\mathcal{O}(N_{\text{simplices}}\, d)$ and converges in near-linear time per pass.

Complexity/Scaling: Per-sample compute is $\mathcal{O}(k\, d)$; deferred moves/splits trigger occasional $\mathcal{O}(d^2)$ work. Stage 1's grid uses 7–9 points per decade with GP refinements (few evaluations), leading to wall-time dominated by ANN queries and moment updates; HNSW scales well sub-linearly with N. Stage 2 adds simplex-native operations; torsion and shape metrics are $\mathcal{O}(d^2)$ but applied sparsely on topology events. Overall, memory grows with nodes/links/simplex count; empirical growth stays within a $\tilde{1}0\%$ budget per recursion level due to the Torsion Ladder and prune rules.

5 Conclusion

Building on competitive Hebbian learning's maximum-entropy partition of the data PDF, Proteus augments this core with a two-stage, scale-aware architecture that resolves the classic trade-off between speed and fidelity in manifold learning. By using a fast exploratory engine to first

discover the structural "scaffold" of the data, it can then deploy a more powerful simplicial model to build a final, high-fidelity, generative representation. Our analysis shows that this hybrid approach produces models that are not only topologically accurate but also have strong predictive power.

A key advantage of Proteus is that many quantities are *derived from first principles* rather than tuned by hand: the geometric grid ratio r, the EWMA weight α, and the dual motion rates η_{cent} and η_{GNG} all follow from scale-space resolution, statistical half-life, and variance-correction arguments (see SI S2). Together with the statistical gauntlets for pruning (SI S3), these choices make the system robust, interpretable, and largely parameter-free in practice.

This work establishes a robust method for modeling complex data. The rich, generative model produced by the Proteus pipeline serves as a foundational layer for a wide array of subsequent applications, from high-performance search to automated pattern discovery. These applications, along with architectural enhancements for streaming data, will be detailed in subsequent work.

As shown in ablations, the two-stage design and torsion audit are necessary for both topological fidelity and generative quality; the added discussion on Simplex vs. Node Equilibria explains why our simplex-level objective still drives node-level balance in practice.

6 References Prep

This section collects potential references to be formalized and cited.

Manifold Learning: Isomap (Tenenbaum et al., 2000); Locally Linear Embedding (Roweis & Saul, 2000); t-SNE (van der Maaten & Hinton, 2008); UMAP (McInnes, Healy, Melville, 2018).

Hierarchical and Density Clustering: HDBSCAN (Campello, Moulavi, Sander, 2013/2015).

Scale-Space Theory: Lindeberg, Scale-Space Theory (1994, 1998).

Growing Neural Gas and Variants: GNG (Fritzke, 1995) and node-centric extensions.

Discrete Geometry and Topology: Discrete exterior calculus / discrete differential forms (Desbrun et al.); Divergence Theorem references (standard vector calculus texts); Persistent homology and Betti numbers (Edelsbrunner & Harer).

Generative Models: VAEs (Kingma & Welling, 2013/2014); GANs (Goodfellow et al., 2014); Normalizing Flows (Dinh et al., 2014/2016; Papamakarios et al., 2021).

Appendix: Notation

We summarize the core symbols used throughout the paper.

- $m_i \in \mathbb{R}^d$: EWMA residual mean at node i.
- $s_i \in \mathbb{R}^d$: EWMA of squared residual at node i; variance $\sigma_i^2 = s_i - m_i^2$.
- $\rho_i = \|m_i\|/(\sigma_i + \varepsilon)$: incoherence ratio at node i.
- k: neighborhood size (Stage 1: k=8; Stage 2: $k \approx d$).
- $D_{subspace}$: current working dimensionality; $d_{final,i}$: smoothed local intrinsic dimension at node i.
- $s_{control} \in [0, 1)$: optimizer control; $\tau_{global} = -D_{subspace} \log(1-s_{control})$.
- $\tau_{local,i} = -d_{final,i} \log(1-s_{control})$: per-node growth threshold.
- $\Phi(\tau)$: scale-space response evaluated at threshold τ.

- Ω_S: simplex 2-form (discrete torsion) on simplex S; $\kappa_S = \|\Omega_S\|_F$.
- $R_S = \kappa_S/\tau^*$: torsion ratio used by the Torsion Ladder at the fixed scale τ^*.
- $\tilde{\rho}$: neighbor-normalized incoherence; CV $= \mathrm{std}(\tilde{\rho})/\mathrm{mean}(\tilde{\rho})$.