STRUCTURED KNOWLEDGE ACCUMULATION: GEODESIC LEARNING PATHS AND ARCHITECTURE DISCOVERY IN RIEMANNIAN NEURAL FIELDS

Bouarfa Mahi Quantiota
Université Joseph Fourier
Grenoble, Auvergne-Rhône-Alpes, FR
info@quantiota.org

May 12, 2026

ABSTRACT

The Structured Knowledge Accumulation (SKA) framework views neural learning as an entropy-guided process through which knowledge organizes itself. In this work, we extend SKA beyond standard discrete architectures and into spatially continuous neural fields by adopting a Riemannian geometric perspective. This leads to the notion of *Riemannian SKA Neural Fields*. Here, the learning substrate is treated as an information manifold whose metric tensor reflects local entropy, information density, and structural gradients. Under this formulation, knowledge travels along geodesic flows that naturally balance global organization with local variability. This viewpoint yields a continuous, entropy-based least-action principle that unifies architecture discovery, representation shaping, and learning dynamics within a single variational framework. To make these ideas computationally practical, we employ finite element methods and discrete exterior calculus, introducing a scalable discretization scheme suitable for high-dimensional settings. The resulting paradigm provides a biologically inspired and computationally efficient mechanism for real-time adaptive learning. This work brings together ideas from information theory, differential geometry, and neural computation, opening a promising direction for developing resource-efficient continuous-space AI models.

Keywords SKA · Riemannian Geometry · Forward-Only Learning · Local Entropy · Information Density · Structural Gradients · Least Action Principle · Finite Element Method

MSC (2020): Primary 68T07; Secondary 68T05, 68Q32.

1 Introduction

The SKA framework, established in prior works [1, 2], proposed a novel notion of learning in neural systems. SKA distinguishes itself by defining learning as not merely a set of changes to parameters of a system, but an iterative refining process, guided by entropy, in which knowledge organizes itself in a forward-only dynamics. This unique perspective provided a solid, viable, plausible, and computationally simple way of framing the accumulation of information in artificial and biological neural systems.

The initial version of SKA and its subsequent temporal extension has been successful in analyzing systems with a layered structure and forward evolution. Unfortunately, these systems' scope and applicability is limited. Systems with a layered architecture are too simplistic and miss the essence of many continuous spatial systems. Such systems have a wide range of characteristics, including, but not limited to, variable neuron density, biased directional connectivity, and functionally differing structural roles dependent on tasks. We are therefore interested in moving beyond discrete systems. We examine the SKA framework in relation to the neural fields and continuous systems. This provides a greater degree of freedom in describing learning processes in spatially elaborate and functionally organized systems.

In this paper, we present the *Riemannian SKA Neural Fields* – a generalized version of SKA that fills the gap we described. Here, the neural field is considered as an information manifold equipped with a Riemannian metric that captures both entropy gradients and structural features such as varying density. Knowledge accumulation is believed to be represented as movement along the manifold's geodesic, which captures an ideal equilibrium between the the underlying architecture and information organization. This version of SKA is free of the ad hoc probabilistic synaptic assumptions SKA and others used, and provide a consistent way to understand learning in neural systems with a complex spatial and/or functional configuration.

The Riemannian SKA Neural Fields demonstrate a nascent potential for adaptive learning by bringing together differential geometry, information theory, and modeling theory of biology. This integration of Riemannian geometry and SKA theory is an extension of the entropic least action principle, and, for the first time, includes time, space, and function. The potential of this combination to derive scalable solutions for efficient neural computation is profound.

1.1 Organization and Structure of the Paper

This paper is organised as follows. After the introduction Section 1, a thorough review of associated work is established in Section 2. This section covers neural field models (Section 2.1), information geometry in learning (Section 2.2), Riemannian geometry in neural networks (Section 2.3), architecture discovery (Section 2.4), discrete exterior calculus in neural systems (Section 2.5), biological neural organization (Section 2.6), and a positioning of the contemporary contribution within the present literature (Section 2.7). After the literature review, Section 3 examines the research gap and summarizes the inspiration for the suggested framework.

Section 4 formalizes the evolution from discrete SKA framework to continuous neural representations. This section opens with an assessment of the SKA framework (Section 4.1), followed by the neuron density introduction and continuous fields (Section 4.2), and the development of knowledge and decision fields as tensor valued quantities (Section 4.3). Local computation procedures and the notions governing knowledge propagation are discussed in Section 5, including local neural computation (Section 5.1), spatial connectivity (Section 5.2), and temporal evolution of the knowledge field (Section 5.3).

The learning 's entropic foundations are presented in Section 6. This contains local entropy density discussed in Section 6.1, field equations obtained from the principle of entropic least action (Section 6.2), and universal learning equations (Section 6.3) are thoroughly framed. Computational and numerical aspects of the suggested formulation are established in Section 7. This section discusses spatial discretization (Section 7.1), local SKA representations (Section 7.2), discretized dynamics (Section 7.3), and the numerical computation of spatial entropy gradients (Section 7.4), including scalar field construction (Section 7.4.1), element-wise gradient computation (Section 7.4.2), nodal gradient recovery (Section 7.4.3), and entropy gradient initialization (Section 7.4.4).

The geometric construction of learning is given in Section 8, which introduces Riemannian SKA neural fields. This comprises the understanding of Riemannian manifolds as natural learning spaces (Section 8.1), geodesic learning paths (Section 8.2), continuous knowledge propagation along geodesics (Section 8.3), boundary conditions for learning paths (Section 8.4), and the duality between local temporal and spatial learning processes (Section 8.5), with detailed discussions of local temporal learning paths (Section 8.5.1) and spatial propagation learning paths (Section 8.5.2). The realization of learning paths (Section 8.6), computational implementation via discrete exterior calculus (Section 8.7), and dimensional integration with functional modulation (Section 8.8) are also presented. At the end, Section 9 provides conclusion of the paper and future research works.

2 Literature Review

The construction of the Riemannian SKA Neural Fields integrates a number of the most promising related research areas. Below, we outline the contextual background of the research to the neural field modeling, information geometry, Riemannian neural networks, architecture search, and discrete computational geometry.

2.1 Neural Field Models

The modeling of spatially distributed neural systems has typically been built upon continuous neural field theories. The neural field models can be summarized as follows:

- **Amari (1977)** [5]: Pioneered the field of neural modeling with the introduction of differential and lateral-inhibitory neural fields. Amari's field of neural dynamics demonstrates the role of differential equations modeling the neural field in the formation of patterns and the emergence of field behaviors.

- **Kohonen (1990)** [10]: Introduced self-organizing maps and showed that, through competitive learning, topological maps can be realized with local interactions on a continuous neural sheet.
- **Gap in existing work**: While most of the proposed models have been useful in the simulation of the activity dynamics in continuous space, the rules of learning have been externally incorporated. A common thread through most of the classical models is the failure to consider learning as a continuous process constrained by the principles of entropy reduction or knowledge accumulation of a forward-only nature.

2.2 Information Geometry as a Tool for Learning

Information geometry serves as a field of mathematics that can be utilized to understand learning systems:

- **Amari & Nagaoka (2000)** [6] found the field of information geometry and how Fisher information metric can be used to create natural gradient techniques which preserve the geometric structure of probability distributions. Nagaoka and Amari used "geometry" as an approach to "improve" learning from other perspectives.
- **Nielsen (2020)** [18] focuses on information geometry for machine learning and statistical inference and provides one of the best introductory pieces on the subject.
- **Primary difference**: While Amari uses Fisher metric in *parameter space*, we *neural physical space* and use gradients of density and entropy to create a metric. Amari uses the Fisher metric to represent the statistical geometry of parametric models. We consider the entropy-value metric to represent the information-based geometry related to knowledge accumulation in the neural media distributed in space.

2.3 Riemannian geometry in neural networks

There is an increasing number of numerical studies on neural learning and geometry, which are stated as follows.

- **Bronstein et al. (2017)** [8]: proposed the first attempt of geometric deep learning and developed the first deep networks with graphs, manifolds, and non-Euclidean structures. They also emphasized the inductive bias that can be obtained with non-Euclidean structures.
- **Benfenati & Marta (2022)** [12]: applied Riemannian geometry to deep neural networks and provided some tools to analyze learning in curved spaces.
- **Katsman et al. (2023)** [34]: presented Riemannian residual networks and showed that architectural design with Riemannian structures can produce better results.
- **Simon et al. (2021)** [35]: studied interpolation based on geodesic for incremental learning and proved it is useful for some continual learning problems.
- **Chen et al. (2018)** [38]: proposed the first Neural ODE, redefining neural networks in the context of a dynamical system with continuous depth and providing a new perspective on neural network computations.
- **Our contribution**: Other geometric frameworks attempt to *impose* Riemannian geometry in a system to improve optimization or generalization. Our framework is the first to Riemannian geometry that derives from the system's own entropy and density landscapes. Here, geometry is not designed—it is an emergent property of entropy-driven organization of knowledge.

2.4 Architecture Discovery

The automated discovery of neural architectures has become a notable area of study. This includes:

- **Neural Architecture Search (NAS)** [14, 15, 16, 17]: Many NAS strategies utilize reinforcement learning, evolutionary methodologies, and differentiable relaxations to analyze architectural spaces. Although these methodologies are largely effective, they often depend on a prior defined search space and can be very resource intensive.
- **Our Approach**: Rather than conducting a search across defined architectures, we allow optimal connectivity to *emerge continuously* via geodesic computation on the information manifold. Rather than being the result of an externally driven search, architecture is the result of an entropy minimization process. The neural field has the ability to self-organize its connectivity along information-optimal pathways, and, therefore, does not depend on a predefined search space or discrete optimization.

2.5 Neural Systems and Discrete Exterior Calculus

Several existing works have constructed the foundation of using discrete exterior calculus (DEC) as a computational tool to understand the geometric structure of learning systems.

A few of these works include:

- **PyDEC (Bell & Hirani, 2011)** [13]: DEC on simplicial complexes is framed as a software tool. This has made computational differential geometry more accessible to the research community.
- **Desbrun et al. (2005)** [33]: Founded research on DEC. They showed that a certain class of continuous differential forms can be discretized while retaining certain geometrical and topological properties of the structure, and that given a geometry, a topological structure can be associated.
- **Applications** [32, 36]: The most recent works integrate DEC with physics-informed machine learning and scientific computing, showcasing the potential for DEC in complex systems with the ability to impose geometric constraints.
- **Zhang et al. (2023)** [31]: Proposed scaled representations in neural networks for the efficient computation of geodesic distances and paths, indicating the potential of neural systems for the computation of geometric objects.
- **Our Use**: To the best of our knowledge, this is the first time DEC has been applied to *neural learning path computation*. DEC is employed to compute geodesic paths for the optimal knowledge propagation routes in continuous neural fields. Thus, DEC provides a grounded framework for this approach.

2.6 Biological Neural Organization

Biological neural systems suggest valuable metrics for the geometric constructs we propose:

- **Collins et al. (2010)** [3]: Reported significant discrepancies in the density of neurons in different regions of the primate cortex, which is a direct motivation for our application of neural fields with variable density.
- **Hebb (1949), Bi & Poo (1998)** [19, 20]: Pioneered the study of activity-dependent plasticity. In our case, the entropy minimization process is described, from which plasticity in our formulation arises, without external learning rules being imposed.
- **Brain-inspired architectures** [21, 22, 23, 28]: Neuromorphic and brain-inspired systems emphasize the role of spatial arrangement and the architecture of interconnectivity. Our framework provides a theoretical justification for a plethora of these empirical results.

In addition to being consistent with various biological phenomena, Riemannian SKA Neural Fields also, from first principles, recover some of the neural economy principles. The framework illustrates and creates the geodesic pattern in the neural architecture which coincides with a well-known principle of optimization in neural wiring [39], providing an underlying reason for the observation that biological neural systems are optimally costed for neural wiring. Similarly, the economically efficient small-world organization present in real neural systems [40] is the output of a steady-state process brought about by geodesic flow, thus integrating functional efficiency and the metabolic economy under a coherent information-theoretic approach.

2.7 Positioning the Contribution

Bridging various fields of research, the Riemannian SKA Neural Fields framework has several notable original contributions, some examples include:

1. **Intrinsic geometry**: Constraints of a geometric structure are circumvented as the system's own entropy and neuron-density gradients underpin the emergent metric.
2. **Geodesic knowledge propagation**: For the first time, knowledge propagation along geodesics is established, extending the entropic least-action principle of Papers I–II [1, 2] to geodesic knowledge propagation. Every point employs self-updating based on local entropy changes, facilitating efficient processing on dynamic information manifolds.
3. **Continuous architecture discovery**: Where discontinuous search of architecture is a norm, the geodesic computation of the SKA Neural Field is a first for fluid and continuous dispersion of connection patterns.

4. **Unified temporal/spatial principles**: Local learning obey temporal entropic least action, while global information flow follows spatial least action on the information manifold. This creates a unified view of learning across space and time.
5. **Dimensional scalability**: Higher dimensional spaces (4D, 5D, etc.) can be accommodated by the formulation without altering the core equations.
6. **Computational implementation**: By incorporating DEC [13, 33], FEM [7], and tools of Riemannian geometry [4], we present a practical and reproducible pathway for implementation of the theoretical framework.

Some recent research tends to indicate that learning in biological and artificial neural systems involves more than just an optimization process; it also includes learning metrics and learning architectures. Friston's free energy principle is one such approach. It locates learning and inference as entropy minimizing phenomenon and provides a unified variational picture of neural dynamics, perception, and plasticity [9]. From a geometric point of view, Riemannian perspectives suggest that learning is not likely to stay on a flat trajectory; therefore, curvature, and geometric structures of a manifold critically determine how neural representations change and how efficiently information is organized [11].

These ideas also appear in the study of neural systems that change in time; in particular, how transient dynamics and adaptive filtering reveal relationships that cannot be obtained from solely analyzing the static connections [24, 26]. Also, hierarchical organization remains a main theme, with recent work investigating that how deep networks parallel the layered architecture found in biological brains [25]. Manifold-based studies also show that intricate neural activity usually evolves on unexpectedly low-dimensional spaces and structures [27]. Meanwhile, entropy minimization continues to anchor many successful learning strategies [30].

These establishments, along with practical studies which range from structured noise models to adaptive attention mechanisms [29, 37], strengthen the view that effective learning systems benefit from dynamics that are both structured and geometrically grounded.

3 Research Gap and Motivation

The literature presented above provides that structure, geometry, and entropy each plays a significant role in modern concepts of neural learning. Information geometry offers principled learning directions, continuous neural field models capture spatial organization, and Riemannian approaches provide helpful tools for understanding curved optimization strategies. Also, biological evidence always points to learning processes that are efficient, self-organizing, and controlled by the effective use of resources. Despite these developments, current approaches tend to treat these elements separately: geometry is often implemented as an external design choice, entropy performs as an auxiliary objective or regularizer, and network architecture is typically uncovered through discrete search or heuristic optimization.

This division places a significant gap in our understanding of how learning dynamics, connectivity patterns and geometric structures might appear jointly from a single organizing principle. In particular, current frameworks rarely model learning itself as an intrinsically continuous, spatially distributed process in which geometry and architecture arise from the same underlying mechanism. As a result, connections between entropy-driven adaptation, geodesic information flow, and the emergence of functional network structure remain largely implicit rather than explicitly modeled.

The motivation of this work is to address this gap by introducing a unified framework in which learning, geometry, and architecture are not prescribed in advance but instead emerge together. By expressing learning as entropy-driven knowledge accumulation on a Riemannian information manifold, we model neural systems as continuous fields where metric structure, information density, and geodesic learning paths co-evolve. In this context, learning dynamics obey a principled least-action framework, and connectivity patterns come naturally via geodesic propagation rather than discrete architectural search.This perception provides a coherent bridge between geometric theory, biological observations,and practical computation, and offers a basis for interpretable, resource-efficient, and scalable continuous neural models.

4 From Discrete to Continuous Neural Representations

4.1 Review of the SKA Framework

In the SKA framework [1, 2], learning occurs through the alignment of knowledge tensors with decision probability shifts, governed by the principle of entropic least action. The layer-wise entropy is defined as:

$$H^{(l)} = -\frac{1}{\ln 2} \sum_{k=1}^{K} \mathbf{z}_k^{(l)} \cdot \Delta \mathbf{D}_k^{(l)}. \tag{1}$$

This formulation captures how structured knowledge accumulates as neural networks learn to reduce uncertainty through forward-only information flow.

4.2 Neuron Density and Continuous Fields

In our continuous neural field model, to simulate the spatial distribution described in our theory, we utilize a computational model using Simplex noise from the `noise` library, which generates smoothly transitioning density structures in 3D and higher dimensional models [41].

The first step we take in extending SKA to a continuous field theory is to define the concept of neuron density $\rho(\mathbf{r})$, which denotes the number of neurons per unit volume at position $\mathbf{r} = (x, y, z)$. As noted by Collins (2010), in the primate brain, the density of tissue changes across the various regions and can vary between 30,000 and 180,000 neurons in 1 mm^3 [3]. For each infinitely small volume element, dv, at the position $\mathbf{r}$, the number of neurons is defined as:

$$n(\mathbf{r}) = \rho(\mathbf{r}) \cdot dv. \tag{2}$$

This *n* value (number of neurons) indicates a specific dimension the knowledge and decision probability tensors can take at that point in the space.

In our continuous neural field model, to simulate the spatial distribution described in our theory, we utilize a computational model using Simplex noise from the `noise` library, which generates smoothly transitioning density structures in 3D and higher dimensional models. As stated in Figure 1, noise function gives neuron distributions that are of localized variations and biologically-inspired neuron. The statistical properties of Simplex noise allows us to create realistic density fields that resemble the natural variability seen in biological neural tissues across different dimensionalities.

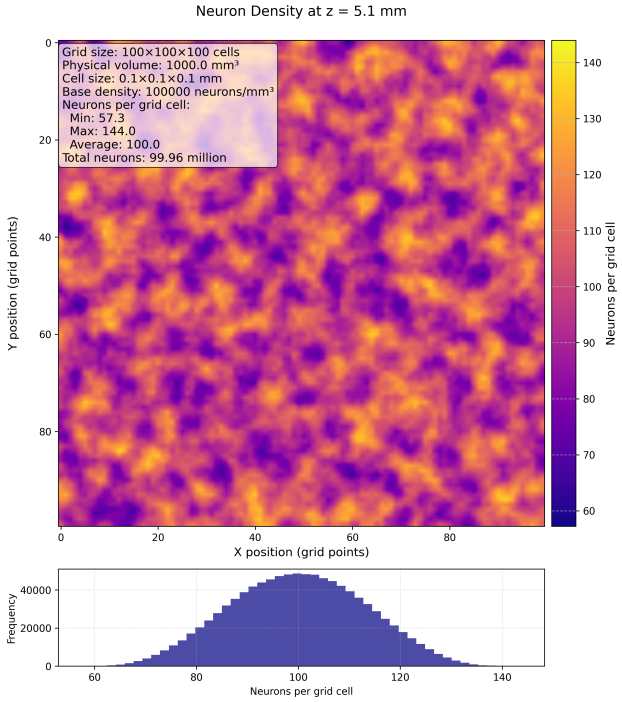

Figure 1: Neuron density per grid cell in a 100×100×100 cubic grid (cell volume = 0.001 mm³) with a baseline density of 100,000 neurons/mm³ modulated by Simplex noise (±50% variability). A colorbar shows the neuron count per cell, and a histogram below illustrates the distribution across the grid.

4.3 Knowledge and Decision Fields as Tensors

For our continuous neural medium, we frame the following tensor fields:

- $\mathbf{W}(\mathbf{r}, t)$: Weight tensor field with dimensions $n(\mathbf{r}) \times d$

- $\mathbf{X}(\mathbf{r}, t)$: Input tensor field of dimension d
- $\mathbf{b}(\mathbf{r}, t)$: Bias tensor field of dimension $n(\mathbf{r})$
- $\mathbf{D}(\mathbf{r}, t)$: Decision probability tensor field of dimension $n(\mathbf{r})$
- $\mathbf{Z}(\mathbf{r}, t)$: Knowledge tensor field of dimension $n(\mathbf{r})$
- $\mathbf{\Phi}(\mathbf{r}, t)$: Knowledge flow tensor field of dimension $n(\mathbf{r})$

Here, $\mathbf{r}$ is a position vector in the D-dimensional space (D=3 for initial implementation, but also potentially higher dimensions D=4, D=5, etc.). These quantities have dual dimensions: they occupy D-dimensional physical space, and at each point in space, they have representational dimensions that depend on the local neuron density $n(\mathbf{r})$. The resulting structure is mathematically rich, as tensors must integrate spatial coordinates and representational dimensions.

The entropy gradient tensor $\nabla \mathbf{h}(\mathbf{r}, t)$ is an exception. The knowledge and decision tensors have representational dimensions that vary with neuron density, but the entropy gradient is an exception, as it has a fixed dimension equal to the spatial dimensions D due to it being derived from scalar entropy differences between two adjacent locations. This property of dimensional invariance makes it a default for guiding how connectivity should flow through the neural field, irrespective of the dimension of the underlying physical space.

The tensor formalism makes it possible to track the interactions and evolutions of tensors with dual nature in an arbitrary number of dimensions. For instance, as knowledge tensors move through spaces with varying neuron density, the interactions comply with the least action and geodesic principles that enable flow of information along a path that is least in entropy.

5 Knowledge Propagation and Local Computation

5.1 Local Neural Computation

The initial knowledge state is obtained as under for each volume element at position $\mathbf{r}$:

$$\mathbf{Z}_0(\mathbf{r}) = \mathbf{W}(\mathbf{r}) \cdot \mathbf{X}(\mathbf{r}) + \mathbf{b}(\mathbf{r}). \tag{3}$$

This gives a direct parallel with the discrete SKA framework, where each volume element functions as a complete neural computation unit that:

- Processes them through local weights $\mathbf{W}$ and biases $\mathbf{b}$
- Takes in external inputs $\mathbf{X}$
- creates an initial knowledge state $\mathbf{Z}_0$
- Evolves this knowledge in accordance with the entropic least action principle

5.2 Spatial Connectivity

The spatial structure of the neural field comes from local interactions between adjacent volume elements. The output from one region becomes the input to neighbouring regions through:

$$\mathbf{X}(\mathbf{r} + d\mathbf{r}) = \mathbf{D}(\mathbf{r}) = \sigma(\mathbf{Z}(\mathbf{r})). \tag{4}$$

This generates a forward-only flow of information via the neural medium and preserves the unidirectional property of the SKA framework.

5.3 Temporal Evolution

The knowledge field changes over time according to:

$$\mathbf{Z}(\mathbf{r}, T) = \mathbf{Z}(\mathbf{r}, 0) + \int_0^T \mathbf{\Phi}(\mathbf{r}, t)\, dt, \tag{5}$$

where at point $\mathbf{r}$, the $\mathbf{\Phi}(\mathbf{r}, t)$ is the knowledge flow field and time t. The decision probability field is obtained from the knowledge field through:

$$\mathbf{D}(\mathbf{r}, T) = \sigma(\mathbf{Z}(\mathbf{r}, T)). \tag{6}$$

This time-dependent framework allows us to:

- Model the propagation of knowledge waves via the neural medium
- Track that how knowledge evolves with respect to time in each volume element
- Study the appearance of structured knowledge patterns with respect to time
- Identify characteristic timescales for knowledge accumulation in various regions

6 Entropy Density and the Principle of Entropic Least Action

6.1 Local Entropy Density

We give the definition of the local entropy density, for each volume element as under:

$$h(\mathbf{r}) = -\frac{1}{\ln 2}\mathbf{z}(\mathbf{r}) \cdot d\mathbf{D}(\mathbf{r}). \tag{7}$$

This shows the entropy contribution from the volume element at position $\mathbf{r}$. The dimension of tensors $d\mathbf{D}$ and $\mathbf{z}$ is associated with the number of neurons in the infinitesimal volume element at that point.
It need to be noted that since the length of $\mathbf{z}(\mathbf{r})$ is proportional to the local neuron density $\rho(\mathbf{r})$, this definition implicitly integrates neuron density—regions of higher ρ yield proportionally larger $\mathbf{z}$, so for a fixed change in $d\mathbf{D}$ and an increase in density, the entropy per volume naturally decreases.

6.2 Field Equations from Entropic Least Action

The principle of entropic least action governs how the knowledge field evolves over time. The weight update equations in the continuous field are:

$$\frac{\partial \mathbf{W}(\mathbf{r}, t)}{\partial t} = \frac{1}{\ln 2}\mathbf{z}(\mathbf{r}, t) \odot \mathbf{D}'(\mathbf{r}, t) \otimes \mathbf{X}(\mathbf{r}, t). \tag{8}$$

For regions receiving input from neighbors:

$$\frac{\partial \mathbf{W}(\mathbf{r}, t)}{\partial t} = \frac{1}{\ln 2}\mathbf{z}(\mathbf{r}, t) \odot \mathbf{D}'(\mathbf{r}, t) \otimes \mathbf{D}(\mathbf{r} - d\mathbf{r}, t). \tag{9}$$

Where:

- $\odot$ denotes element-wise multiplication
- $\mathbf{D}'(\mathbf{r}, t) = \mathbf{D}(\mathbf{r}, t) \odot (1 - \mathbf{D}(\mathbf{r}, t))$ is the derivative of the decision field w.r.t the knowledge $\mathbf{z}(\mathbf{r}, t)$
- $\otimes$ represents the outer product operation

6.3 Universal Learning Equations

Neural networks of the SKA type, which are discrete, are governed by the same field equations due to the entropic least action principle [1, 2]. In the case of discrete networks, the coordinates, $\mathbf{r}$, represent indices of layers instead of positions in continuous space. However, the equation is essentially the same, and this is due to the alterations being made because of the matrix (tensor) dimensions being changed due to modifications in the density of the neurons. The changes with respect to neuron density are usually transparent in the equations and occur due to the structure that the mathematical operations possess, which are quite invariant.

Entropic least action appears to produce learning laws that are independent of the underlying architecture. Also, learning laws can cross the discrete and continuous domains. The equations that describe the behavior of layered networks in information processing are the same equations that describe continuous neural fields, which is perhaps the most fundamental SKA continuous system. The approximations to continuous information manifolds are the discrete neural networks, and it justifies the working of discrete networks in a theoretical sense. The discrete networks are approximations of continuous fundamental processes.

7 Numerical Foundations of 3D SKA Neural Fields

For implementation of the continuous neural field notion in a computational formulation, the spatial domain is discretized, while preserving the theoretical principles of SKA. Each position in the neural medium is represented via a grid-based system that maintains local computational properties and enable efficient numerical simulation.

7.1 Spatial Discretization

We develop a three-dimensional computational grid where, through a scaling factor s, position indices (i, j, k) map to physical coordinates (x, y, z) :

$$x = i \cdot s \tag{10}$$
$$y = j \cdot s \tag{11}$$
$$z = k \cdot s. \tag{12}$$

Each grid point (i, j, k), with $\rho(i, j, k)$ representing the neuron density at the position, denotes a local volume element with dimensions s^3 which contains $n(i, j, k) = \rho(i, j, k) \cdot s^3$ neurons.

7.2 Local SKA Representation

At each grid position (i, j, k), we define a local SKA computation unit characterized by:

- Bias tensor $\mathbf{b}(i, j, k)$ with dimension $n(i, j, k)$
- Weight tensor $\mathbf{W}(i, j, k)$ with dimensions $n(i, j, k) \times d$, where d is the input dimension
- Decision probability tensor $\mathbf{D}(i, j, k)$ with dimension $n(i, j, k)$
- Knowledge tensor $\mathbf{Z}(i, j, k)$ with dimension $n(i, j, k)$

The dimension of these tensors varies across the grid in accordance to the neuron density, which creats a discretized approximation of the continuous tensor fields described in the theoretical framework.

7.3 Discretized Dynamics

The continuous field equations are discretized in both space and time, giving update rules as under:

$$\mathbf{W}(i, j, k, t + \Delta t) = \mathbf{W}(i, j, k, t) + \Delta t \cdot \frac{1}{\ln 2}\mathbf{Z}(i, j, k, t) \odot \mathbf{D}'(i, j, k, t) \otimes \mathbf{X}(i, j, k, t) \tag{13}$$

Also, the local entropy density is calculated as:

$$h(i, j, k) = -\frac{1}{\ln 2}\mathbf{Z}(i, j, k) \cdot \Delta\mathbf{D}(i, j, k). \tag{14}$$

7.4 Numerical Computation of Spatial Entropy Gradient

In the SKA Neural Fields framework, the spatial entropy gradient is essential, as it quantifies the flow of information between the connected neural units. The computation of the entropy gradient is complicated because of the differing neural density of the neural fields, which causes discontinuities in the knowledge and decision tensors. The discontinuities present a challenge which we address using computational physics and engineering's FEM (Finite Element Method) which is robust, mesh based, and usable over fields with spatially varying computational attributes. This enables us to generate a neural field with tetrahedral elements, and allows us to locate entropy in regions with different numbers of neurons. The FEM is unlike analytical equations because they are incapable of dimensional transitions. The FEM is legally used to describe the mathematical nature of the neural medium, and the ways and the correlations of entropy in dimensions throughout.

The aforementioned methodologies applied to neural fields of dimensions greater than 3 (4D, 5D, etc) by substituting tetrahedral elements with D-simplexes (4-simples in 4D, 5-simples in 5D) , computing gradients as D-dimensional tensors, and employing hypervolume weighted averaging for nodal recovery. All assembly steps, shape functions, and Jacobian transformations generalize directly to arbitrary dimension D.

7.4.1 Scalar Field at Mesh Nodes

Let us denote by h_h the finite-element approximation of the continuous entropy field h, where the subscript h refers to the mesh discretization (mesh size parameter). Let each node n carry

$$\mathbf{z}_n \in \mathbb{R}^d, \quad \mathbf{D}_n(t) \in \mathbb{R}^d,$$

where d is the dimension of the decision/knowledge tensors. Defining the one-step change

$$\Delta\mathbf{D}_n = \mathbf{D}_n(t + \Delta t) - \mathbf{D}_n(t),$$

at each global node, we evaluate the scalar entropy value:

$$H_n \;=\; -\frac{1}{\ln 2}\,\mathbf{z}_n \cdot \Delta\mathbf{D}_n. \tag{15}$$

Then interpolating onto the continuous field through the FEM type functions $\{\phi_n(\mathbf{r})\}_{n=1}^{N}$:

$$h_h(\mathbf{r}) \;=\; \sum_{n=1}^{N} H_n\,\phi_n(\mathbf{r}), \tag{16}$$

where N represents the total number of mesh nodes.

7.4.2 Element-wise Gradient Computation

For linear tetrahedral elements, $\nabla\phi_a$ is constant within each element K [7]. Let $I[a]$ map the ath local node of K to its global index. At element K,

$$\nabla_\xi\phi_a \quad\longrightarrow\quad \nabla\phi_a = J_K^{-T}\,\nabla_\xi\phi_a,$$

with J_K the Jacobian of the element mapping and $|K| = \frac{1}{6}|\det J_K|$. The element-wise entropy gradient is then

$$\nabla h_K \;=\; \sum_{a=1}^{4} H_{I[a]}\,\nabla\phi_a, \quad \text{constant on } K. \tag{17}$$

7.4.3 Nodal Gradient Recovery

Having ∇h_K for each element, we recover a nodal gradient $\mathbf{g}_n \in \mathbb{R}^3$ by volume-weighted averaging:

$$\mathbf{g}_n = \frac{1}{\sum_{K\ni n}|K|}\sum_{K\ni n}|K|\,\nabla h_K, \tag{18}$$

where the sum is over all tetrahedra K containing node n.

7.4.4 Initialization of Entropy Gradients in Neural Fields

The learning process begins with a single activated SKA unit at node n_0. We set

$$H_{n_0} \;=\; -\frac{1}{\ln 2}\,\mathbf{z}_{n_0} \cdot \Delta\mathbf{D}_{n_0},$$

and for all other nodes $n \neq n_0$, $H_n = 0$. The scalar field $h_h(\mathbf{r})$ is then assembled via FEM shape functions.

Applying the element-wise and nodal-recovery steps yields

$$\mathbf{g}_n \;=\; \frac{1}{\sum_{K\ni n}|K|}\sum_{K\ni n}|K|\,\nabla h_K,$$

where only tetrahedra incident on n_0 contribute nonzero ∇h_K. As a result, exactly the immediate mesh neighbors of n_0 have nonzero $\mathbf{g}_n$; all others are zero.

The initial set of gradient components designs the first pattern of connectivity, guiding the information flow from the activated unit to its neighbouring units. Each activated unit builds its own entropy in the next learning steps. Since the entropy landscape is developing in greater complexity as a result of the non-uniform gradients in the grid, it is the self-organizing landscape that is continuously evolving a pattern of connectivity that does not rely on predefined connectivity. The entire learning process from a single point is elegantly inspired with this initialization technique as the connectivity of the network in the pattern of the entropy reduction that is set to be realized.

8 Riemannian SKA Neural Fields

8.1 Riemannian Manifolds as Natural Learning Spaces

Riemannian geometry offers a solution to the issues that are a byproduct of the lack a layered model. The lack of spatial, heterogeneous neuron density and function anisotropy are problems that overly layered models exhibit. The

learning space is reframed as a 'curved manifold'. It is this 'manifold' that exhibits its own notion of distance that is not Euclidean but is information-theoretic. A basic property of the SKA formulation presents that uniform density regions ($\nabla\rho = 0$) that they naturally lead to uniform entropy ($\nabla h = 0$). This relationship appears directly from SKA's entropy-driven learning dynamics and offers the theoretical foundation for when Riemannian geometry becomes essential compared to simpler Euclidean approaches suffice.

The learning space in a Riemannian setting becomes a manifold M, with metric tensor g_{ij}, which may vary from point to point depending on the local information properties:

$$g_{ij}(\mathbf{r}) = \alpha \cdot (\nabla h)_i (\nabla h)_j + \beta \cdot (\nabla \rho)_i (\nabla \rho)_j + \gamma \cdot \delta_{ij}. \tag{19}$$

Where:

- $(\nabla h)_i$ and $(\nabla h)_j$ are components of the entropy gradient
- $(\nabla \rho)_i$ and $(\nabla \rho)_j$ are components of the density gradient
- α, β, and γ are positive weighting parameters
- δ_{ij} is the Kronecker delta, ensuring positive-definiteness of the metric.

Justification of the Metric Structure. The metric structure in Equation (19) follows directly from three principles:

1. **Scalar invariance.** Both $h(\mathbf{r})$ and $\rho(\mathbf{r})$ are scalars; therefore their gradients ∇h and $\nabla \rho$ are the lowest-order covariant objects that encode spatial variation. Any Riemannian metric constructed from these fields must depend on tensors of rank 2. The outer products $(\nabla h)_i (\nabla h)_j$ and $(\nabla \rho)_i (\nabla \rho)_j$ are the unique symmetric rank-2 tensors formed from first derivatives.
2. **Minimality of order.** Higher-order derivatives (e.g., Hessians $\partial_i \partial_j h$) would introduce curvature-like information into the metric itself, conflating geometry with its derivatives. SKA requires that curvature emerges from the metric, not inside it. Therefore the metric must depend only on first-order quantities.
3. **Linearity and independence.** The metric must remain positive-definite and must separate entropic variation from density variation. A linear combination with coefficients α, β, γ is the minimal structure that:
   - preserves symmetry
   - preserves positive-definiteness
   - treats h and ρ as independent sources of geometric anisotropy
   - reduces to Euclidean geometry when $\nabla h = \nabla \rho = 0$

Under these constraints , the above expression is a *minimal first-order* Riemannian metric compatible with entropy-driven SKA learning. This formulation creates a space where "distance" is defined not by physical separation but by information difference: points with similar entropy and density properties become informationally "closer" regardless of their Euclidean positions.

In this Riemannian formulation, learning paths follow geodesics—locally distance-minimizing curves—according to the geodesic equation:

$$\frac{d^2 x^i}{dt^2} + \Gamma^i_{jk} \frac{dx^j}{dt} \frac{dx^k}{dt} = 0 \tag{20}$$

where:

- x^i are coordinates in the information manifold,
- Γ^i_{jk} are Christoffel symbols derived from the metric tensor $g_{ij}(\mathbf{r})$,
- t parametrizes the learning progression

8.2 Geodesic Learning Paths

In Riemannian geometry, the shortest path between two points is a geodesic, which generalizes the concept of straight lines to curved spaces. For SKA, geodesics represent optimal learning paths that minimize information distance.

The geodesic distance between points a and b is given by:

$$d(a, b) = \min_{\gamma} \int_a^b \sqrt{g_{ij} \frac{dx^i}{dt} \frac{dx^j}{dt}} dt \tag{21}$$

Where the minimum is taken over all paths γ connecting a and b. This represents the information distance in our learning space.

8.3 Geodesic Paths for Continuous Knowledge Propagation

The Riemannian metric

$$g_{ij}(\mathbf{r}) = \alpha \cdot (\nabla h)_i (\nabla h)_j + \beta \cdot (\nabla \rho)_i (\nabla \rho)_j + \gamma \cdot \delta_{ij}$$

describes geodesics that support continuous knowledge propagation by discouraging motion along strong entropy or density gradients. As a result , geodesic paths naturally follow regions of relatively stable entropy and neuron density. These geodesics, explaining the shortest possible trajectories in relation to the information distance, are said to be in accordance with the smooth trajectories of the gradients in the (∇h), computed through the FEMs (Section 7.4), and the gradient of density, $\nabla \rho$, which is said to be generated by the Simplex noise (Section 4.2). This property of minimal gradients bypasses changes in the information manifold (or topology), suggesting that the flow of knowledge is uninterrupted across varying levels of entropy and computational power. This flow is also observed in biological neural systems [3].

The flow of geodesics exemplifies the core information theoretic principles of the structure of neurons in the brain. The weight update rule arising from the least action principle of entropy in the previous SKA framework [1, 2]:

$$\frac{\partial \mathbf{W}(\mathbf{r}, t)}{\partial t} = -\nabla_{\mathbf{z}} H(\mathbf{r}, t) \otimes \mathbf{X}(\mathbf{r}, t)$$

where the entropy gradient with respect to knowledge is given by:

$$\nabla_{\mathbf{z}} H = -\frac{1}{\ln 2} \mathbf{z}(\mathbf{r}, t) \odot \mathbf{D}'(\mathbf{r}, t)$$

with $\mathbf{D}'(\mathbf{r}, t) = \mathbf{D}(\mathbf{r}, t) \odot (1 - \mathbf{D}(\mathbf{r}, t))$ being the sigmoid derivative. Keep in mind that $\nabla_{\mathbf{z}} H$ has the same dimension as the decision probability tensor $\mathbf{D}$, which denotes the change in entropy rate per unit of knowledge observed within each representational dimension. There is an outer product $\otimes$ structure here that looks like an analogy with correlation-based learning systems. Nevertheless, the driving force differs from correlation-based learning systems. In contrast to correlation-based learning systems, which strengthen the connections of co-active components, the outer product in SKA is the alignment of the entropy gradient, $(\nabla_{\mathbf{z}} H)$, with the input information $\mathbf{X}$. The gradient $(\nabla_{\mathbf{z}} H)$ is not simply the neural activity. Rather, it is the direction of information $(\mathbf{X})$ that facilitates the optimal arrangement of knowledge through the reduction of entropy.

This distinction accounts for the fact that the apparent Hebbian-like connectivity in biological systems could be an indication of profound information-theoretic optimization. The neural systems empirically observed to reduce entropy show patterns of connectivity that correlate with the necessary responses to reduce entropy, providing the appearance of activity-based correlation but actually conforming to the principles of information geometry. The biological phenomenon of correlated activity strengthening the connections [19] is a phenomenon of information systems that minimize entropy, not a rule of learning that operates independently.

The time coherency of gradient-driven paths produces effects similar to spike-timing-dependent effects [20]. In SKA Riemannian Neural Fields, the sequential activation of units along geodesic paths generates time correlations that amplify the optimally informative connections via the entropy minimization process. This temporal ordering is a result of the geodesic structure – the principle of least information distance produces temporal relations while still preserving a forward-only learning mechanism. This approach is applied computationally via the discrete exterior calculus (Section 8.6), where edge weights

$$w_{ij} = \sqrt{g_{uv}\, \Delta x_{ij}^u\, \Delta x_{ij}^v}$$

model local metric variations, allowing robust geodesic computation utilising `pydec` and `FEniCS`. By giving priority to information-optimal paths which minimize entropy and density gradients, the metric tensor frames a biologically computationally scalable and plausible mechanism for self-organizing connectivity in continuous neural fields.

8.4 Boundary Condition for the Learning Path

The evolution of SKA Neural Fields is impacted by its learning initialization and boundary conditions. While the initialization sets the first differential entropy gradient within the area of influence of the seed, the boundary conditions specify how field knowledge is captured, and how it is distributed and amassed as learning takes place. The learning path possesses two defining boundary conditions:

1. **Starting Point Condition**: The learning path initiates from the most neuron-dense areas within the field. This boundary condition is about harnessing the processing potential of the neuron-assembled mass to compute

and initiate SKA. By starting from the neuron-dense maxima, the system is able to focus on the areas of higher representational potential to form and create the primary structures of knowledge, which can then be transferred toward the lower-capacity regions.

2. **Endpoint Density Condition**: Instead of setting particular positional endpoint constraints, we introduce the characteristic of endpoint density. This allows the system to determine autonomously the density endpoints of the neural field. The learning path organizes itself between the high-end starting points and the density-defined endpoint to form the most efficient structures for achieving other objectives, such as classification.

Under these boundary conditions, the learning SKA Neural Field is a *geodesic* in the info manifold defined by the Riemannian metric tensor *g ij(r)*. This shows that knowledge is disseminated through the least local entropy minimizing (most informative) pathways through the densely populated origin zone to the emergent endpoint zones and that the pathway is adjusted to the entropy and density contours.
With these boundary conditions, the continuous accumulation of knowledge facilitates the updating of the spatial entropy gradient, prompting the flow of information to the geodesics that minimize entropy, both locally and globally. This approach, which organizes itself, upholds the principle of forward learning while introducing the freedom to create pathways that are adaptive toward specific tasks.
Applied to classification tasks, this framework allows the system to not only locally minimize entropy but also develop geodesic pathways toward high-density processor zones and desirable classification output zones. Perhaps a more substantial contribution is the system's ability to bypass pre-defined pathways and utilize optimal geodesic learning pathways to the defined boundary zones, as opposed to traditional neural architectures with fixed connectivity patterns.

8.5 The Duality of Local and Spatial Learning Paths

The SKA Neural Fields framework, in its simplest form, captures a fundamental duality in the accumulation of knowledge. Locally, knowledge 'grows' in time as per the `first law of entropic least action`; globally, it 'spreads' in time and space, as per the `second law of spatial least action` across the entropy and density landscapes. The temporally bound and spatially related learning principles, function in conjunction, leading to the said knowledge accumulation along the orthogonal time and space dimensions.

8.5.1 Local Temporal Learning Path

Knowledge processes temporally at each fixed spatial position $\mathbf{r}$, in accordance *locally* with the entropic least action principle (the first law of SKA learning). The action integral for local learning is defined as follows:

$$H(\mathbf{r}) = \frac{1}{\ln 2} \int L(\mathbf{r}, t)\, dt \tag{22}$$

where $L(\mathbf{r}, t)$ is the local Lagrangian (e.g., local entropy). The trajectory obtained describes the process of knowledge accumulation at each position (the learning process that is time-optimally entropic).

8.5.2 Spatial Propagation Learning Path

In accordance with the second law of SKA learning, the accumulation of knowledge propagates spatially across the field along geodesic paths in the information manifold. These paths represent trajectories of optimal information efficiency. This spatial propagation is characterized by the Riemannian metric $g_{ij}(\mathbf{r})$ and follows the geodesic equation:

$$d(a, b) = \min_{\gamma} \int_a^b \sqrt{g_{ij} \frac{dx^i}{dt} \frac{dx^j}{dt}}\, dt \tag{23}$$

Importantly, the integrand

$$L = \sqrt{g_{ij} \frac{dx^i}{dt} \frac{dx^j}{dt}}$$

serves as the **Lagrangian** for the spatial learning path. The geodesic trajectory γ is thus obtained by minimizing the action functional

$$S[\gamma] = \int_a^b L\, dt.$$

Minimizing this action using the Euler–Lagrange equations yields the geodesic equation:

$$\frac{d^2 x^i}{dt^2} + \Gamma^i_{jk} \frac{dx^j}{dt} \frac{dx^k}{dt} = 0,$$

where Γ^i_{jk} are the Christoffel symbols derived from the metric g_{ij}. This makes clear that the spatial propagation of knowledge in Riemannian SKA neural fields is governed by a variational principle that is conceptually analogous to the temporal least action principle at each point. Thus, the learning path is not merely an optimal route by construction, but is dynamically discovered through the minimization of the action functional defined by the information geometry of the neural field.

The spatial learning path γ connects regions (for example, from a high-density starting point to an endpoint defined by boundary conditions) and represents an optimal route that minimizes information distance—i.e., a **geodesic** in the information manifold.

8.6 Learning Path Realization in Riemannian SKA Neural Fields

The construction and execution of learning paths in Riemannian SKA Neural Fields happens in the following ways:

1. **Initialization of Entropy Fields:** The learning process starts with the construction of the entropy field across the neural domain. In 7.4.4, a seed node (usually located in the region of maximum neuron density) is activated. The entropy values for the first time and the first of such landscapes (entropy gradients) is computed for the first time. This will form the informational scaffolding for the subsequent knowledge propagation.
2. **Specification of Boundary Conditions:** The learning path starts and finishes with predefined boundary conditions. As mentioned in subsection 8.4, the beginning (starting point) is usually fixed at a maximum in the density. Meanwhile, the end point is either a neuron density threshold or something defined by the task (classification boundary, for instance). *If there are multiple possible positions in the field for the end point density criterion, then the end point is defined by the entire set of valid positions.*
3. **Computation of the Geodesic Path:**The geodesic (the information-optimal path) is calculated on the Riemannian manifold given the entropy field and boundary conditions. The entropy field at this point is usually initiated with a single activated seed node so it is quite sparse. Given that the last destination is not known a priori, some candidate destinations are delineated. The candidate endpoints are usually in some areas of similar neuron density. A geodesic is calculated from the seed to each candidate, and the one with the least information distance is chosen. This step determines the endpoint in the first instance, and then it sets the initial path for learning. The path selected has the least distance in terms of information according to the local metric $g_{ij}(\mathbf{r})$ and is in accordance with the entropy and density gradients.
4. **Forward Pass ($K = 1$):** A forward pass is done over the geodesic path. As the geodesic traverses nodes, knowledge and decision probability tensors are updated via the entropic least action principle. This forward pass represents the first step ($K = 1$) of the learning path and exemplifies the SKA forward-only update principle.
5. **Update of Entropy Fields:** After the forward pass, knowledge is incorporated and a new entropy field is calculated to reflect the current state of knowledge. As a result, the entropy field is updated and the available gradients are changed to facilitate the formation of new geodesic paths and enable further learning.

This step by step method shows how the SKA Neural Field effectively self-organizes and collects knowledge in a way that minimizes entropy in a local region and optimizes information in the system as a whole. The SKA paradigm's efficiency is in the balance of local updates and the routing of gradients. Convergence occurs when, for some optimal K^*, after a finite number of forward passes, the geodesic path becomes stable and is invariant under further iterations. At this point, the local entropy values $h_i^{K^*}$ along the geodesic also converge, signifying that the knowledge gain and entropy decrease along the optimal information path have reached a stasis. Therefore, the learning geodesic is completed when the geodesic becomes invariant and the local entropy values along the geodesic converge.

The Riemannian SKA Neural Fields framework provides optimal local learning path architecture. It enables the system to discover the most efficient learning connectivity pattern unshackled from fixed architectural designs, guided by the entropy and density gradient. This is the most significant leap compared to previous methodologies where the architectural design was done before the learning commenced.

Figure 2 illustrates how learning paths evolve over successive forward passes as the entropy landscape is updated, demonstrating the dynamic nature of geodesic formation in the Riemannian SKA Neural Field. The temporal separation between successive geodesics is governed by the characteristic time step Δt, which represents the fundamental weight update interval that allows the entropy field to reorganize and establish new optimal pathways for knowledge propagation. As the entropy evolves and converges to equilibrium, the difference between successive geodesics vanishes, indicating that the system has completed its architecture discovery process and identified the optimal neural connectivity pattern for information flow— marking the convergence of both the temporal and spatial learning paths established in the duality framework above (Section 8.5).

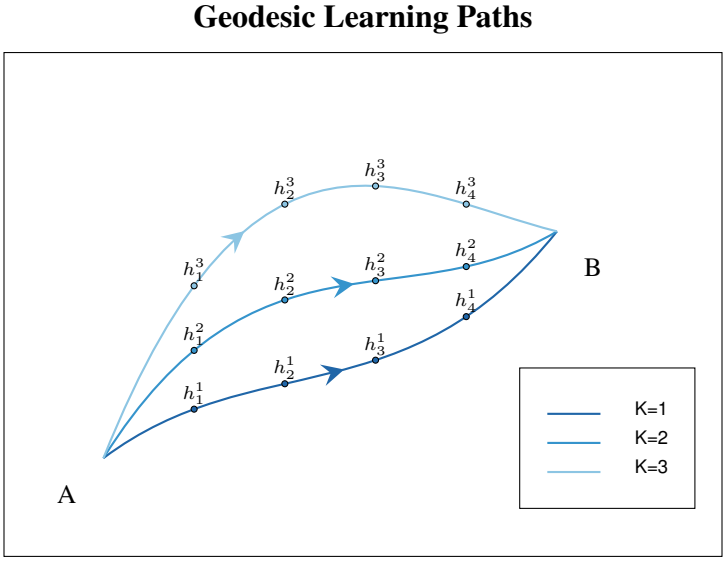

Figure 2: Three elevated geodesic learning paths connecting points A and B for different forward passes ($K = 1, 2, 3$). Each path shows local entropy values h_i^K at intermediate points. As K increases, the paths rise and curve more to optimize information flow according to the evolving entropy landscape. Arrows indicate mid-path knowledge direction.

8.7 Computational Implementation via Discrete Exterior Calculus

For the SKA Neural Fields framework, fully integrating the continuous Riemannian geometry comes with a considerable degree of difficulty, owing to the manifold's complexity, high dimensions, and heterogeneous neuron density. Discrete Exterior Calculus (DEC), realized through the Python library `pydec`, offers an approach that is geometrically differential while remaining computationally feasible. This subsection describes the DEC-centered pipeline that describes integration of `pydec` for the computation of discrete geodesics and differential forms, `FEniCS` for the FEM, `TorchManifolds` for the differentiable Riemannian metric and geodesic, `noise` for the neuron density, and `PyTorch` for the tensor calculations.

The main features of the DEC implementations are as follows.

1. **Simplicial Complex**: A tetrahedral mesh (or higher dimensional D-simplex mesh) that is adaptable to the information landscape is made using `pydec`'s simplicial complex features, with the vertices symbolizing the computational units.
2. **Discrete Differential Forms**: Decompositions of gradients (e.g. entropy gradient ∇h) and tensor fields (e.g. knowledge tensor $\mathbf{Z}$, decision probability tensor $\mathbf{D}$) are treated as discrete differential forms, using `pydec`'s cochain framework.
3. **Hodge Star Operator**: The Hodge star operator that is included in `pydec` to help with the calculation of gradients and divergence is the same as the one used to compute the entropy gradients (Section 7.4.3) and does not require continuous representations.
4. **Discrete Geodesics**: The computation of the shortest paths in the information manifold (i.e., the paths that represent the most optimal learning trajectories) is performed using `pydec`'s weighted graph utilities.

To find the geodesic path between each pair of units a and b:

1. **Initialization**: Using `pydec` create a tetrahedral mesh of the neural field, where the vertices are the computational units. For the neuron density $\rho(\mathbf{r})$, use `noise` Simplex noise as described in Section 4.2.
2. **Metric Assignment**: For each edge (i, j), assign a weight:

$$w_{ij} = \sqrt{g_{uv} \Delta x_{ij}^u \Delta x_{ij}^v} \tag{24}$$

where Δx_{ij}^u is the coordinate difference in dimension u, and g_{uv} is the metric tensor (Equation 19) evaluated at the edge midpoint using `TorchManifolds`. The metric incorporates entropy gradients (∇h) from `FEniCS` (Section 7.4) and density gradients ($\nabla \rho$) from the noise-generated field.

3. **Path Computation**: Apply Dijkstra's algorithm in `pydec` to find the shortest path from a to b, representing the information-optimal learning path (Section 8.2).
4. **Gradient Flow**: Update knowledge and decision probability tensors $(\mathbf{Z}, \mathbf{D})$ along the geodesic using the entropic least action principle (Equations 8, 13), with tensor operations (e.g., $\odot$, $\otimes$) performed using `PyTorch`. Recompute the entropy field $h_h(\mathbf{r})$ using `FEniCS`'s FEM shape functions (Section 7.4.1).

The pipeline employs:

- `FEniCS`: To perform the interpolation of the entropy values `PyTorch` calculated as H_n, into the continuous entropy field $h_h(\mathbf{r})$ and then compute ∇h_K on the tetrahedral elements, thus creating the gradient input for `pydec` to perform its discrete steps.
- `TorchManifolds`: To facilitate the Riemannian metric tensor calculation along with the Christoffel coefficients as $g_{ij}(\mathbf{r})$ (see Equation 19) using the PyTorch autograd, thus enabling the verification of continuous geodesics.
- `pydec`: To compute discrete geodesics and perform differential steps, using the gradient from `FEniCS` to connect the continuous and discrete spaces.

Computational dependencies include:

- `pydec`: Required for performing DEC (a simplicial mesh, discrete forms, and geodesics).
- `FEniCS`: Required for the calculation of the centroid entropy gradients using the FEM.
- `TorchManifolds`: Required for manipulating the Riemannian metric tensor.
- `noise`: Required for generating the neuron density based on Simplex noise.
- `PyTorch`: Required for performing tensor calculations (for example, $\mathbf{Z} \odot \mathbf{D}'$, $\mathbf{Z} \cdot \Delta\mathbf{D}$).

Using only `FEniCS` and `TorchManifolds` is not possible since `pydec` is a prerequisite for the computation of discrete geodesics on simplicial meshes, an option that `FEniCS` (FEM-centered) and `TorchManifolds` (continuous geometry) do not provide. In the case of elevated dimensions (e.g. 4D, 5D), `pydec` extends its focus to D-simplex complexes, `FEniCS` modifies to the FEM, and `TorchManifolds` adapts a generalized structure for the metric tensor. For `TorchManifolds` we reference Chattopadhyay [4] and for `FEniCS` we reference Hughes [7].

8.8 Dimensional Integration and Functional Modulation

The Riemannian method has a notable strength in that it combines physical space with other properties like information or functionality in a single geometric representation. Other models segregate such properties—be it attention, abstraction, or context—and treat them as separate computational layers or modules. The Riemannian SKA Neural Field model, by contrast, embeds them within the metric itself, enabling these properties to modulate geometry without the traveling along additional spatial axes.

Let $f_k(\mathbf{r})$ be a scalar functional field representing local information states or modulatory field factors affecting learning. These fields influence metric coefficients as follows:

$$g_{ij}(\mathbf{r}) = \alpha\,(\nabla h)_i(\nabla h)_j + \beta\,(\nabla \rho)_i(\nabla \rho)_j + \sum_{k=1}^{n} \gamma_k\,(\nabla f_k)_i(\nabla f_k)_j + \gamma_0\,\delta_{ij}. \tag{25}$$

In this formula, $(\nabla f_k)_i$ is the i-th component of the gradient of the k-th functional field, affecting the local curvature as well as the manifold's connectivity. These fields do not increase the neural manifold's coordinate dimension; instead, they function as geometric modulators to modify the information metric.

This particular formulation addresses the complexity and challenges that arises from needing to define a new higher-dimensional representation for each new property by redirecting the focus to the Riemannian framework. Complex functionalities encapsulated by this framework are through metric modulation on a fixed-D manifold, where the computation complexity increases linearly in proportion to the number of modulatory fields, as opposed to exponentially scaling with the number of new properties.

Clarification on Dimensionality. In this work, the ambient dimension of the manifold where the neural field is defined is $D \in \{3, 4, 5\}$. In this regard, introducing additional scalar fields f_k does not increase the dimension of the coordinates; these fields merely serve to modify the coefficients of the Riemannian metric $g_{ij}(\mathbf{r})$, thus altering the geometry on a fixed-D manifold. Consequently, for any given D, our FEM/DEC formulations pertain to D-simplices (for instance, tetrahedra in 3D, 4-simplices in 4D) without the need for additional coordinates.

9 Conclusion and Future Research Works

In this article, we presented Riemannian SKA Neural Fields, a geometric extension of the SKA formulation to continuous neural systems. By modeling the neural medium as a Riemannian manifold with a metric tensor encoding entropy and density gradients, we enable knowledge propagation along geodesics, balancing information structuring with spatial heterogeneity. This approach generalizes the entropic least action principle from discrete layered networks to spatially continuous fields, providing a unified, biologically plausible paradigm for adaptive learning in complex architectures.

Our framework bridges differential geometry, information theory, and neural computation, and addresses limitations in already existing SKA works by integrating anisotropic connectivity and functional organization. Through discrete geometric computation and finite element discretization, the framework provides scalable techniques for higher-dimensional simulations, and opens novel pathways for efficient AI design. While theoretical in scope, this foundation paves the way for practical applications in neuromorphic computing and self-organizing systems, with empirical validations to follow in future studies. The implementation and evaluation of these principles constitute the next step toward realizing self-organizing, entropy-guided architectures in practice.
While this paper establishes the theoretical foundations of Riemannian SKA Neural Fields, extending the discrete SKA framework to continuous, geometrically principled systems, several avenues remain for practical validation and enhancement. Future research will focus on implementing the proposed framework using the outlined computational pipeline, including FEM and discrete exterior calculus (DEC) via libraries such as `pydec`, `FEniCS`, and `TorchManifolds`.

In addition, we aim to investigate higher-dimensional generalizations ($D = 4, 5$) for modeling complex spatial spaces and to extend the methodologies of entropy-driven architecture emergence to more heterogeneous and dynamic fields. Because the SKA entropy expression is independent of spatial dimension, these extensions remain mathematically natural. A subsequent paper will detail these implementations, experimental results, and performance metrics, demonstrating scalability for real-time, resource-efficient AI systems.

These directions aim to empirically validate the theoretical claims presented here and demonstrate the practical viability of Riemannian SKA Neural Fields in large-scale learning systems.

References

[1] Mahi, B., "Structured Knowledge Accumulation: An Autonomous Framework for Layer-Wise Entropy Reduction in Neural Learning," arXiv preprint, arXiv:2503.13942 [cs.LG], 2025.

[2] Mahi, B., "Structured Knowledge Accumulation: The Principle of Entropic Least Action in Forward-Only Neural Learning," arXiv preprint, arXiv:2504.03214 [cs.LG], 2025.

[3] Collins, C. E., Airey, D. C., Young, N. A., Leitch, D. B., & Kaas, J. H., "Neuron densities vary across and within cortical areas in primates," Proceedings of the National Academy of Sciences, 107(36), 15927–15932, 2010.

[4] Chattopadhyay, S. (2025). TorchManifolds: High-Performance Differential Geometry and Topology (Version 0.1.0). GitHub Repository: `https://github.com/shirso22/TorchManifolds`.

[5] Amari, S. (1977). Dynamics of pattern formation in lateral-inhibition type neural fields. *Biological Cybernetics*, 27(2), 77–87.

[6] Amari, S., & Nagaoka, H. (2000). Methods of information geometry. *Translations of Mathematical Monographs*, 191, American Mathematical Society & Oxford University Press.

[7] Hughes, T. J. R. (2000). The FEM: Linear Static and Dynamic Finite Element Analysis. *Dover Publications*.

[8] Bronstein, M. M., Bruna, J., LeCun, Y., Szlam, A., & Vandergheynst, P. (2017). Geometric deep learning: Going beyond Euclidean data. *IEEE Signal Processing Magazine*, 34(4), 18–42.

[9] Friston, K. (2010). The free-energy principle: a unified brain theory?. *Nature Reviews Neuroscience*, 11(2), 127-138.

[10] Kohonen, T. (1990). The self-organizing map. *Proceedings of the IEEE*, 78(9), 1464-1480.

[11] Hauser, M., & Ray, A. (2017). Principles of Riemannian Geometry in Neural Networks. In Advances in Neural Information Processing Systems 30 (NIPS 2017).

[12] Benfenati, A., & Marta, A. (2022). A singular Riemannian geometry approach to Deep Neural Networks I. Theoretical foundations. arXiv preprint arXiv:2201.09656.

[13] Bell, Nathan and Anil N. Hirani. "PyDEC: Software and Algorithms for Discretization of Exterior Calculus." ACM Trans. Math. Softw. 39 (2011): 3:1-3:41.

[14] Zoph, B., & Le, Q. V. (2017). Neural architecture search with reinforcement learning. *International Conference on Learning Representations (ICLR)*.

[15] Liu, H., Simonyan, K., & Yang, Y. (2018). DARTS: Differentiable architecture search. *International Conference on Learning Representations (ICLR)*.

[16] Pham, H., Guan, M. Y., Zoph, B., Le, Q. V., & Dean, J. (2018). Efficient neural architecture search via parameter sharing. *International Conference on Machine Learning (ICML)*.

[17] Elsken, T., Metzen, J. H., & Hutter, F. (2019). Neural architecture search: A survey. *Journal of Machine Learning Research*, 20(55), 1-21.

[18] Nielsen, F. (2020). An elementary introduction to information geometry. *Entropy*, 22(10), 1100.

[19] Hebb, D. O. (1949). *The Organization of Behavior: A Neuropsychological Theory*. Wiley.

[20] Bi, G. Q., & Poo, M. M. (1998). Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type. *Journal of Neuroscience*, 18(24), 10464–10472.

[21] Suárez, L. E., Richards, B. A., Lajoie, G., & Misic, B. (2021). Learning function from structure in neuromorphic networks. *Nature Machine Intelligence*, 3(9), 771-786.

[22] Susin, E., & Destexhe, A. (2022). Brain connectivity meets reservoir computing. *PLoS Computational Biology*, 18(11), e1010639.

[23] Zeng, Y., Zhao, D., Zhao, F., Shen, G., Dong, Y., Lu, E., ... & Huang, T. (2023). BrainCog: A spiking neural network based, brain-inspired cognitive intelligence engine for brain-inspired AI and brain simulation. *Patterns*, 4(8), 100789.

[24] Pascucci, D., Rubega, M., & Plomp, G. (2020). Modeling time-varying brain networks with a self-tuning optimized Kalman filter. *PLoS Computational Biology*, 16(8), e1007566.

[25] Nonaka, S., Majima, K., Aoki, S., & Kamitani, Y. (2021). Brain hierarchy score: Which deep neural networks are hierarchically brain-like? *iScience*, 24(9), 103013.

[26] Bondanelli, G., & Ostojic, S. (2020). Coding with transient trajectories in recurrent neural networks. *PLoS Computational Biology*, 16(2), e1007655.

[27] Gallego, J. A., & Perich, M. A. (2017). Neural manifolds for the control of movement. *Neuron*, 94(5), 978-984.

[28] Maass, W. (2021). Essential principles for cortical microcircuits that enable brain-like computation and energy efficiency. *National Science Review*, 8(5), nwab120.

[29] Perlin, K. (2002). Improving noise. In Proceedings of the 29th annual conference on Computer graphics and interactive techniques (SIGGRAPH '02), pp. 681–682. Association for Computing Machinery.

[30] Grandvalet, Y., & Bengio, Y. (2004). Semi-supervised Learning by Entropy Minimization. In Advances in Neural Information Processing Systems (Vol. 17), pp. 529–536.

[31] Zhang, Q., Hou, J., Adikusuma, Y., Wang, W., & He, Y. (2023). NeuroGF: A Neural Representation for Fast Geodesic Distance and Path Queries. In Advances in Neural Information Processing Systems (Vol. 36).

[32] Manti, S., & Lucantonio, A. (2024). Discovering interpretable physical models using symbolic regression and discrete exterior calculus. Machine Learning: Science and Technology, 5(1), 015005.

[33] Desbrun, M., Hirani, A. N., Leok, M., & Marsden, J. E. (2005). Discrete Exterior Calculus. arXiv preprint arXiv:math/0508341.

[34] Katsman, I., Chen, E. M., Holalkere, S., Lou, A., Lim, S.-N., & De Sa, C. M. (2023). Riemannian Residual Neural Networks. In Advances in Neural Information Processing Systems (Vol. 36).

[35] Simon, C., Koniusz, P., & Harandi, M. (2021). On Learning the Geodesic Path for Incremental Learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1591–1600.

[36] Trask, N., Huang, A., & Huynh, T. (2022). Enforcing exact physics in scientific machine learning: a data-driven exterior calculus on graphs. Journal of Computational Physics, 456, 110969.

[37] Simon, C., Koniusz, P., Nock, R., & Harandi, M. (2021). Adaptive-Attentive Geolocalization from Few Queries: A Hybrid Approach. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 291–300.

[38] Chen, R. T. Q., Rubanova, Y., Bettencourt, J., & Duvenaud, D. (2018). Neural Ordinary Differential Equations. In Advances in Neural Information Processing Systems (Vol. 31).

[39] Cherniak, C. (1994). Component placement optimization in the brain. *Journal of Neuroscience*, 14(4), 2418–2427.

[40] Bullmore, E., & Sporns, O. (2012). The economy of brain network organization. *Nature Reviews Neuroscience*, 13(5), 336–349.

[41] Mahi, B. (2025). neuron-density-fields: 3D 4D 5D coherent neuron density fields using Simplex noise. GitHub Repository: `https://github.com/quantiota/neuron-density-fields`.