
STRUCTURED KNOWLEDGE ACCUMULATION: AN AUTONOMOUS FRAMEWORK FOR LAYER-WISE ENTROPY REDUCTION IN NEURAL LEARNING

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

February 8, 2026

ABSTRACT

We introduce the Structured Knowledge Accumulation (SKA) framework, which reinterprets entropy as a dynamic, layer-wise measure of knowledge alignment in neural networks. Instead of relying on traditional gradient-based optimization, SKA defines entropy in terms of knowledge vectors and their influence on decision probabilities across multiple layers. This formulation naturally leads to the emergence of activation functions such as the sigmoid as a consequence of entropy minimization. Unlike conventional backpropagation, SKA allows each layer to optimize independently by aligning its knowledge representation with changes in decision probabilities. As a result, total network entropy decreases in a hierarchical manner, allowing knowledge structures to evolve progressively. This approach provides a scalable, biologically plausible alternative to gradient-based learning, bridging information theory and artificial intelligence while offering promising applications in resource-constrained and parallel computing environments.

Keywords Structured Knowledge Accumulation · Layer-wise Entropy Measurement · Knowledge Alignment in Neural Networks · Gradient-free Optimization · Biologically Plausible Learning.

1 Introduction

The role of entropy in intelligent systems is crucial in how one interprets the underlying structure and functioning of AI. Deep learning, especially the deep superposition of gradients, has been aided by traditional learning paradigms that have proven extremely effective, such as the ones based on gradient backpropagation. Unfortunately, these methods are computationally expensive, biologically unrealistic, and impenetrably opaque. There needs to be an attempt to overcome the divide between the theoretical tenets of information theory and its actual engineering implementation in neural networks. In view of this problem, we present a novel framework that treats the redefinition of entropy as a continuous process of structured knowledge accumulation, which enables a different approach to the internal organization of AI learning system.

Entropy, as classically defined by [21], is given as:

$$H = - \sum p_i \log_2 p_i, \quad (1)$$

which measures uncertainty in a discrete, static probabilistic system. While this framework serves as the basis of information theory, it falls short of getting the continuous and dynamic structuring of intelligent systems knowledge such as neural networks. The sigmoid function which is defined as:

$$\sigma(z) = \frac{1}{1 + e^{-z}}, \quad (2)$$

*Use footnote for providing further information about author (webpage, alternative address)—*not* for acknowledging funding agencies.

has been a keystone of AI. This function is commonly used for activation in neural networks. However, its theoretical justification beyond empirical utility is an open question.

Conventional training through backpropagation spreads errors backward via the network, which requires additional computational means while lacking limiting scalability, biological plausibility, and real-world utilization. Tackling these limitations, this article establishes the SKA framework, which redefines entropy as a hierarchical and continuous process of knowledge accumulation.

We propose:

1. Entropy as a dynamic measure, expressed in layers as:

$$H^{(l)} = -\frac{1}{\ln 2} \sum_{k=1}^K \mathbf{z}_k^{(l)} \cdot \Delta \mathbf{D}_k^{(l)}, \quad (3)$$

approximating the continuous entropy formulation:

$$H = -\frac{1}{\ln 2} \int z dD. \quad (4)$$

2. Cumulative knowledge \mathbf{z} serves as the building block for literacy.
3. Learning rules involving minimization of local entropy and lack of backpropagation.

We formulate the sigmoid function as an emergent phenomenon of continuous entropy flow and transfer this paradigm to fully connected networks, showing how learning emerges from the micromatching of knowledge with the decision-making dynamics. This framework offers a biologically accurate, scalable explanation of deep learning that displacement methods, increases the optimization effectiveness, and adds optimization transparency.

The creation of AI systems, neuroscience, and computational intelligence would all benefit from understanding knowledge accumulation through entropy dynamics. With this information, SKA is able to streamline neural network training in resource-limited environments, promote concurrent learning, and aid in the formulation of more interpretable AI systems. The framework further motivates the development of energy-efficient and biologically inspired computing systems, making it a fundamental step toward the next generation of intelligent architectures.

2 Literature Review

The study of AI and neural networks has greatly benefitted from the research done on knowledge-based entropy systems. Early works of [1] proposed the concept of entropy in information theory, which has since been a driving force for many subsequent learning paradigms in neural networks. Modern research developed an entropy-based framework to improve neural networks and create knowledge alignment. The assessment of entropy has been studied in several different ways. [9] studied a method of neural joint entropy approximation which contributes to how entropy can be leveraged for learning efficacy. [11] analyzed integrated information from the perspective of decoding, which proposes a way to assess the level of complexity present in neural representations. Additionally, some of this research concentrates on learning theories with biological motivation. [3] proposed notion networks that are inspired by the brain, demonstrating learning from watching scenes unfold in a disorganized fashion, emulating human perception. [13] examined the collapse of integrated information mechanisms due to anesthetic loss which sheds light on the processing of neural information.

Multiple difficulties stem from using backpropagation-based learning, including the lack of biological plausibility as well as computational inadequacy. These concerns have driven the exploration of alternative optimization methods. [12] proposed Generative ConvNets, which are energy-based models capable of synthesizing dynamic patterns and investigates into local learning mechanisms. Similarly, [19] illustrate the relationship between contrastive and backpropagation Hebbian learning, which offer a more believable biological framework. [5], regarding the encoding of innate ability via a genomic bottleneck, investigated how biological concepts can be integrated into machine learning systems. [17] built on this idea using biologically-informed neural networks to guide mechanistic modeling based on sparse experimental data.

The last several years have witnessed considerable progress in optimizing the training processes for deep neural networks. [20] presented SAGA, an incremental gradient optimization technique that enhances convergence and lowers variance. [22] pioneered acceleration catalysts also known as first-order convex optimization acceleration in the context of machine learning. The vanishing gradients problem remains unresolved in deep neural network architectures, and [23] proposed ResNets for addressing this enduring challenge. [10] analyzed the use of contrastive divergence for training energy-based latent variable models, which enhances representation learning in neural networks. Adding

entropy-based learning yields results beyond those achievable with standard approaches in machine learning. In their study, [6] introduced a neural network-based approach for estimating joint entropy by leveraging mutual information neural estimation techniques. In addition, Hess et al. [15] studied how continual learning models accumulate knowledge and confront feature forgetting, highlighting that while absolute forgetting may be minimal, relative forgetting can significantly impede the development of robust general representations. In the same context, as discussed by [7], knowledge distillation techniques have evolved significantly. Furthermore, [8] introduced a main/subsidiary network framework to simplify binary neural networks by employing a learning-based approach to filter pruning, thereby enhancing efficiency without compromising performance.

3 Our Obtained Results

3.1 Redefining Entropy in the SKA Framework

Entropy traditionally quantifies uncertainty in probabilistic systems, but its classical form is static and discrete, limiting its applicability to dynamic learning processes like those in neural networks. In the SKA framework, we redefine entropy as a continuous, evolving measure that reflects knowledge alignment over time or processing steps. This section contrasts Shannon’s discrete entropy with our continuous reformulation, enabling the use of continuous decision probabilities and supporting the derivation of the sigmoid function through entropy minimization.

3.1.1 Classical Shannon Entropy

For a binary system with decision probability D , Shannon’s entropy is:

$$H = -D \log_2 D - (1 - D) \log_2(1 - D). \quad (5)$$

Its derivative with respect to D is:

$$\frac{dH}{dD} = \log_2 \left(\frac{1 - D}{D} \right). \quad (6)$$

This formulation assumes D is a fixed probability, typically associated with discrete outcomes (e.g., 0 or 1). While foundational, it does not capture the continuous evolution of knowledge in a learning system, where D may vary smoothly as the network processes inputs. To address this, we seek a continuous entropy measure that accommodates dynamic changes in D , aligning with the SKA’s focus on knowledge accumulation.

3.1.2 Entropy as a Measure of Knowledge Accumulation

In SKA, we redefine entropy for a single neuron as a continuous process:

$$H = -\frac{1}{\ln 2} \int z dD. \quad (7)$$

Here, z represents the neuron’s structured knowledge, and dD is an infinitesimal change in the decision probability, treated as a continuous variable over the range $[0, 1]$. The factor $-\frac{1}{\ln 2}$ ensures alignment with base-2 logarithms, consistent with Shannon’s information units. Unlike the static snapshot of Equation (1), this integral captures how entropy accumulates as z drives changes in D , reflecting a dynamic learning process.

For a layer l with n_l neurons over K forward steps, we approximate this continuous form discretely:

$$H^{(l)} = -\frac{1}{\ln 2} \sum_{k=1}^K \mathbf{z}_k^{(l)} \cdot \Delta \mathbf{D}_k^{(l)}, \quad (8)$$

where $\mathbf{z}_k^{(l)} = [z_1^{(l)}(k), \dots, z_{n_l}^{(l)}(k)]^T$ is the knowledge vector, $\Delta \mathbf{D}_k^{(l)} = [\Delta D_1^{(l)}(k), \dots, \Delta D_{n_l}^{(l)}(k)]^T$ is the vector of decision probability shifts, and the scalar product is:

$$\mathbf{z}_k^{(l)} \cdot \Delta \mathbf{D}_k^{(l)} = \sum_{i=1}^{n_l} z_i^{(l)}(k) \Delta D_i^{(l)}(k). \quad (9)$$

The total network entropy sums over all layers:

$$H = \sum_{l=1}^L H^{(l)}. \quad (10)$$

Equation (4) is the core theoretical construct, with Equation (3) as its practical discrete approximation. As $K \rightarrow \infty$ and $\Delta \mathbf{D}_k^{(l)}$ becomes infinitesimally small, Equation (3) approaches the continuous integral, enabling us to model D as a smooth function of z . This continuous perspective is essential for deriving the sigmoid using dynamics in later sections, while the discrete form supports implementation in neural architectures.

3.1.3 Accumulated Knowledge

Knowledge accumulates over steps:

$$z_k = z_0 + \sum_{f=1}^k \Delta z_f. \quad (11)$$

In a layer, $\mathbf{z}_k^{(l)}$ evolves, reducing $H^{(l)}$ as it aligns with $\Delta \mathbf{D}_k^{(l)}$.

3.2 Deriving the Sigmoid Function

The SKA framework posits that the sigmoid function emerges naturally from continuous entropy minimization, linking structured knowledge to decision probabilities. This section demonstrates that when D follows $D = \frac{1}{1+e^{-z}}$, the SKA entropy H_{SKA} equals the classical Shannon entropy H_{Shannon} , differing by a constant (zero). By leveraging the continuous formulation from Section 2, we derive this equivalence, reinforcing the framework's theoretical grounding.

3.2.1 Key Definitions

Shannon Entropy (for binary decisions): For a binary system with continuous decision probability D :

$$H_{\text{Shannon}} = -D \log_2 D - (1 - D) \log_2 (1 - D). \quad (12)$$

SKA Entropy (layer-wise, for a single neuron): The SKA entropy, defined continuously as in Section 3.1.2, is:

$$H_{\text{SKA}} = -\frac{1}{\ln 2} \int z dD, \quad (13)$$

where $z = -\ln(\frac{1-D}{D})$ relates knowledge to D , consistent with $D = \frac{1}{1+e^{-z}}$ as shown in Section 3.2.1.

3.2.2 Equivalence Proof

Substituting $z = -\ln(\frac{1-D}{D})$ (or equivalently, $z = \ln(\frac{D}{1-D})$) into H_{SKA} :

$$H_{\text{SKA}} = -\frac{1}{\ln 2} \int \ln\left(\frac{D}{1-D}\right) dD. \quad (14)$$

Evaluate the integral with substitution $u = D$, $du = dD$:

$$\int \ln\left(\frac{D}{1-D}\right) dD = D \ln\left(\frac{D}{1-D}\right) + \ln(1-D). \quad (15)$$

Substituting back:

$$H_{\text{SKA}} = -\frac{1}{\ln 2} \left[D \ln\left(\frac{D}{1-D}\right) + \ln(1-D) \right]. \quad (16)$$

Rewrite $\ln\left(\frac{D}{1-D}\right) = \ln D - \ln(1-D)$:

$$H_{\text{SKA}} = -\frac{1}{\ln 2} [D \ln D - D \ln(1-D) + \ln(1-D)]. \quad (17)$$

Factorize:

$$H_{\text{SKA}} = -\frac{1}{\ln 2} [D \ln D + (1-D) \ln(1-D)]. \quad (18)$$

Thus:

$$H_{\text{SKA}} = H_{\text{Shannon}}. \quad (19)$$

3.2.3 Implications

- **Zero Difference:** The SKA and Shannon entropies are identical (differing by zero) when $D = \frac{1}{1+e^{-z}}$, confirming the sigmoid as an emergent property of continuous entropy reduction.
- **Knowledge Alignment:** This equivalence stems from z structuring D to minimize uncertainty, as defined in Sections 2 and 3.

3.2.4 Significance

1. **Theoretical Consistency:** SKA extends Shannon entropy into a continuous, dynamic context while preserving its core properties for sigmoidal outputs.
2. **Backpropagation-Free Learning:** Since $H_{\text{SKA}} = H_{\text{Shannon}}$, layer-wise entropy minimization aligns with classical uncertainty reduction, achieved via forward dynamics alone.
3. **Biological Plausibility:** The continuous, local alignment of z with D mirrors plausible neural learning mechanisms.

3.2.5 Summary

When D is the sigmoid function, H_{SKA} matches H_{Shannon} exactly, with a difference of zero. This result, derived from the continuous entropy $H_{\text{SKA}} = -\frac{1}{\ln 2} \int z dD$, validates SKA's foundation and its seamless integration with information theory, leveraging continuous dynamics for neural learning with classical information theory.

3.3 The Fundamental Law of Entropy Reduction

The SKA framework establishes a fundamental law governing how entropy decreases as structured knowledge evolves. This section derives this law using continuous dynamics, reflecting the continuous nature of decision probabilities and knowledge accumulation introduced in Sections 2 and 3. We then provide a discrete approximation for practical implementation, ensuring the framework's applicability to neural networks while rooting it in a continuous theoretical foundation.

3.3.1 Continuous Dynamics

For a single neuron, the rate of entropy change with respect to structured knowledge z is derived from the continuous entropy $\bar{H} = -\frac{1}{\ln 2} \int z dD$. Taking the partial derivative:

$$\frac{\partial \bar{H}}{\partial z} = -\frac{1}{\ln 2} z D(1 - D). \quad (20)$$

This follows from $D = \frac{1}{1+e^{-z}}$ (as derived in Section 4), where $\frac{dD}{dz} = D(1 - D)$, and reflects the neuron's local contribution to entropy reduction. For a layer l with n_l neurons at step k , this extends to each neuron i :

$$\frac{\partial H^{(l)}}{\partial z_i^{(l)}(k)} = -\frac{1}{\ln 2} z_i^{(l)}(k) D_i^{(l)}(k) \left(1 - D_i^{(l)}(k)\right). \quad (21)$$

Equation (21) governs the continuous reduction of layer-wise entropy $H^{(l)}$, driven by the alignment of $z_i^{(l)}(k)$ with the sigmoidal decision probability $D_i^{(l)}(k)$. This dynamic, localized process underpins the SKA's forward-only learning mechanism, leveraging the continuous evolution of D over time or input processing.

3.3.2 Discrete Dynamics

In practice, neural networks operate over discrete forward steps. For a single neuron at step k , the entropy gradient approximates the continuous form, incorporating the change in decision probability $\Delta D_k = D_k - D_{k-1}$:

$$\frac{\partial \bar{H}}{\partial z} \Big|_k = -\frac{1}{\ln 2} [z_k D_k (1 - D_k) + \Delta D_k]. \quad (22)$$

For layer l at step k , this becomes:

$$\frac{\partial H^{(l)}}{\partial z_i^{(l)}(k)} = -\frac{1}{\ln 2} z_i^{(l)}(k) \left[D_i^{(l)}(k) \left(1 - D_i^{(l)}(k)\right) + \Delta D_i^{(l)}(k) \right]. \quad (23)$$

Equation (23) adapts continuous law to discrete steps where $\Delta D_i^{(l)}(k)$ denotes the change in $D_i^{(l)}(k)$. While (21) captures the ideal case of continuous dynamics, and Equation (23) provides a computable approximation, aligning knowledge adjustments with observed changes in decision probabilities across discrete iterations.

3.4 Generalization to Fully Connected Networks

The SKA framework builds from single neurons to fully connected neural networks without losing continuity, utilizing earlier defined principles of continuous entropy reduction. In a L -layer network, knowledge and decision probabilities develop in a stratified manner, decreasing the overall entropy through local, one-way movements. In this part, I describe the workings of SKA at different levels while keeping the network's biology-inspired scalability in mind.

For a neural network having L layers:

- $\mathbf{D}_k^{(l)} = \sigma(\mathbf{z}_k^{(l)})$, the decision probabilities derived via the sigmoid function,
- $\mathbf{z}_k^{(l)}$ the knowledge vector at layer l and step k is defined as,
- \mathbf{D}^l is defined as decision probabilities of l -th layer of neural network,

Note: In this form, the neural network is used for static systems where information does not change with time.

The continuous formulation provides the basis for layer-wise entropy which is defined discretely as:

$$H^{(l)} = -\frac{1}{\ln 2} \sum_{k=1}^K \mathbf{z}_k^{(l)} \cdot \Delta \mathbf{D}_k^{(l)}. \quad (24)$$

The coherence relation describing the knowledge and decision shift as measured is given by:

$$\mathbf{z}_k^{(l)} \cdot \Delta \mathbf{D}_k^{(l)} = \|\mathbf{z}_k^{(l)}\| \|\Delta \mathbf{D}_k^{(l)}\| \cos(\theta_k^{(l)}). \quad (25)$$

Total network entropy across layers is defined as:

$$H = \sum_{l=1}^L H^{(l)}. \quad (26)$$

Progress is made by adjusting $\mathbf{z}_k^{(l)}$ to $\Delta \mathbf{D}_k^{(l)}$ at each layer, enabling local reductions of $H^{(l)}$ without necessitating backpropagation of errors. In the limit of continuity, this smoothing aligns with evolution of knowledge's alignment, in this case represented with gaps for ease of computation.

3.5 Learning Without Backpropagation

The SKA model achieves learning through localized entropy minimization, allowing no backpropagation in favor of forward-only dynamics. This subsection breaks down the weight update procedure and the relevant metrics framed within continuously reducing entropy, with adjustments made for discretized entropy minimization in fully-connected networks.

The driving force behind the entropy minimization at layer l is:

$$\frac{\partial H^{(l)}}{\partial w_{ij}^{(l)}} = -\frac{1}{\ln 2} \sum_{k=1}^K \frac{\partial (\mathbf{z}_k^{(l)} \cdot \Delta \mathbf{D}_k^{(l)})}{\partial w_{ij}^{(l)}}. \quad (27)$$

The update rule adjusts weights forward:

$$w_{ij}^{(l)} \leftarrow w_{ij}^{(l)} - \eta \frac{\partial H^{(l)}}{\partial w_{ij}^{(l)}}. \quad (28)$$

Here, $\Delta D_i^{(l)}(k)$ is computed directly from forward passes, bypassing the need for error backpropagation. This local adjustment aligns with the continuous dynamics of knowledge evolution, approximated over discrete steps.

Step-wise Entropy Change

To quantify knowledge accumulation, the step-wise entropy change at layer l and step k is:

$$\Delta H_k^{(l)} = H_k^{(l)} - H_{k-1}^{(l)} = -\frac{1}{\ln 2} \mathbf{z}_k^{(l)} \cdot \Delta \mathbf{D}_k^{(l)}. \quad (29)$$

This measures uncertainty reduction as $\mathbf{z}_k^{(l)}$ aligns with $\Delta \mathbf{D}_k^{(l)}$, with total layer entropy as:

$$H^{(l)} = \sum_{k=1}^K \Delta H_k^{(l)}. \quad (30)$$

Entropy Gradient

The gradient of $H^{(l)}$ with respect to $\mathbf{z}_k^{(l)}$ at step k is:

$$\nabla H^{(l)} = \frac{\partial H^{(l)}}{\partial \mathbf{z}_k^{(l)}} = -\frac{1}{\ln 2} \mathbf{z}_k^{(l)} \odot \mathbf{D}'_k^{(l)} - \Delta \mathbf{D}_k^{(l)}, \quad (31)$$

where $\mathbf{D}'_k^{(l)} = \mathbf{D}_k^{(l)} \odot (1 - \mathbf{D}_k^{(l)})$ is the sigmoid derivative. This gradient guides updates to minimize $H^{(l)}$, aligning knowledge with decision shifts.

Knowledge Evolution Across Layers

The gradient $\nabla H^{(l)} = -\frac{1}{\ln 2} \mathbf{z}_k^{(l)} \odot \mathbf{D}'_k^{(l)} - \Delta \mathbf{D}_k^{(l)}$ captures the change in entropy for a given layer l during the k^{th} iteration. $\mathbf{D}_k^{(l-1)}$ feeds to $\mathbf{z}_k^{(l)}$ so each layer can flexibly self-adjust – so the model interprets wide features first, then narrows down on details to refine decision making. This self-driven adjustment resembles a flow of knowledge that has been sampled at discrete intervals.

Governing Equation of SKA

The network evolves according to:

$$\nabla H^{(l)} + \frac{1}{\ln 2} \mathbf{z}_k^{(l)} \odot \mathbf{D}'_k^{(l)} + \Delta \mathbf{D}_k^{(l)} = 0, \quad (32)$$

where $\nabla H^{(l)}$ reduces entropy for each layer, with updates done by $-\nabla H^{(l)}$ to steer $\mathbf{z}_k^{(l)}$ towards $\Delta \mathbf{D}_k^{(l)}$.

Inter-Layer Entropy Change

The entropy change between layers l and $l+1$ at step k is:

$$\Delta H_k^{(l,l+1)} = -\frac{1}{\ln 2} [\mathbf{z}_k^{(l+1)} \cdot \Delta \mathbf{D}_k^{(l+1)} - \mathbf{z}_k^{(l)} \cdot \Delta \mathbf{D}_k^{(l)}]. \quad (33)$$

This describes the movement of knowledge over space and complements the temporal flow provided by $\nabla H^{(l)}$, while entropy is flowing down the network.

4 Applications of the Obtained Results

4.1 Application to Neural Networks

SKA organizes information into different layers, weakening total entropy H through flow-like processes that can be simulated in steps. A multilayer perceptron (MLP) can be trained to minimize H , where $\cos(\theta_k^{(l)})$ stands as a suitable indicator for the level of alignment between $\mathbf{z}_k^{(l)}$ and $\Delta \mathbf{D}_k^{(l)}$. This method takes advantage of the system's scalability and self-management, which can be utilized in various networks, hence using a forward-only approach.

Layer-wise Entropy Reduction in SKA

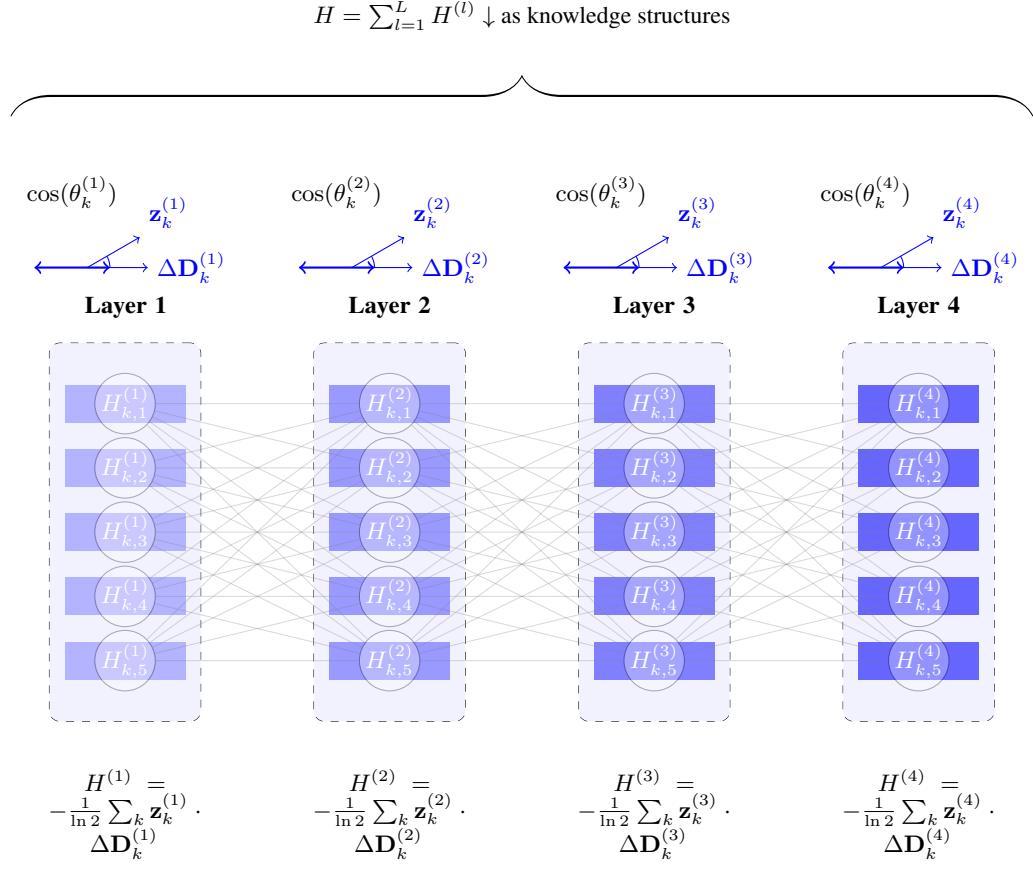


Figure 1: Layer entropy reduction for a given step k . The entropy level is represented by color intensity, darker blue corresponds to lower entropy. For illustration purpose only, entropy progressively decreases from Layer 1 to Layer 4. Each layer lowers entropy on a local level by coordinating knowledge vector $\mathbf{z}_k^{(l)}$ with decision change vector $\Delta \mathbf{D}_k^{(l)}$ as measured by $\cos(\theta_k^{(l)})$.

4.2 Tensor-Based Implementation of SKA

In order to improve computing performance and scalability, **tensor-based implementation** is introduced to the SKA framework. This approach preserves the theory of SKA and uses tensor operations to allow for efficient and parallelizable learning. With a tensor approach to SKA’s mathematical formulation, we optimize **knowledge accumulation, entropy reduction, and decision updates** computation on several layers and step K .

Tensor Definitions

The operation of a neural network under SKA can be expressed using the following tensors: The following tensors can be used to represent a neural network functioning under SKA:

- **Knowledge Tensor (Z):** Represents structured information within each neuron at different **layers (L)**, **steps (K)** and **number of neurons (n_{\max})**-dimension.
- **Decision Probability Tensor (D):** Keeps neuron activations as knowledge values transformed using sigmoid function.
- **Shift Tensor ΔD :** Represents **local probability shifts** of change between steps, aligning with SKA’s entropy-based learning mechanism.

- **Weight Tensor (W) and Bias Tensor (b):** Set knowledge parameters that change over time for enhancing knowledge retention.

Forward-Only Learning and Knowledge Update

SKA revises knowledge using only the forward path as shown in below SKA update formula.

$$\mathbf{Z} = \mathbf{W} \cdot \mathbf{X} + \mathbf{b}$$

\mathbf{X} refers to the input tensor. As opposed to backpropagation, SKA comes with a notable advantage; SKA does not require a back propagation of gradients, which means less computational resources are consumed.

Entropy Computation and Learning

SKA's entropy formulation can be naturally expressed using tensor operations:

$$\mathbf{H} = -\frac{1}{\ln 2} \sum_{k=1}^{K-1} \mathbf{Z}_{k+1} \cdot \Delta \mathbf{D}_k,$$

where **entropy decreases layer by layer** as structured knowledge accumulates.

To optimize learning, entropy gradients are computed as:

$$\nabla \mathbf{H} = -\frac{1}{\ln 2} \mathbf{Z} \odot \mathbf{D}' - \Delta \mathbf{D},$$

where \mathbf{D}' is the derivative of the sigmoid function.

Weight Updates and Learning Stability

Weight changes occur according to an entropy minimization rule:

$$\mathbf{W} \leftarrow \mathbf{W} - \eta \frac{\partial \mathbf{H}}{\partial \mathbf{W}}.$$

An **alignment tensor** determines if the knowledge updates deviate from their expected paths:

$$\Theta_{l,k} = \cos(\theta_{k+1}^{(l)}) = \frac{\mathbf{Z}_{l,k+1} \cdot \Delta \mathbf{D}_{l,k}}{\|\mathbf{Z}_{l,k+1}\| \|\Delta \mathbf{D}_{l,k}\|}.$$

This alignment mechanism ensures that SKA remains **structured** and **controlled**.

Advantages of Incorporating Tensors in Mathematical Modeling

- **Simultaneous Execution of Tasks:** The structure of tensors allows for parallel processing within a single layer or across multiple layers and neurons, optimizing resource usage.
- **Flexible Connectivity Patterns:** The arrangement allows designs with additional hidden layers to be included without major alterations to the design process.
- **Analogous Low Level Operation:** Structures which only compute the forward pass are compatible with design constraints for power-oriented low-complexity real-time operations.

This tensor-based model solidifies SKA's claim as a **computationally efficient, biologically plausible, and scalable learning paradigm**. Further work will assess its performance when compared to model versions utilizing traditional backpropagation, investigating applications in edge computing, real-time AI, and neural network optimizations.

4.3 Entropy Evolution in SKA

One of the most prominent observations within the SKA framework is the distinctive movement of entropy across layers during learning processes. Unlike standard deep networks with non-uniformly varying dynamics of entropy across layers, SKA shows an extraordinary phenomenon: **all layers are observed tending towards a singular entropy equilibrium value, converging at a predefined entropy level**. This suggests that entropy is not simply decreasing but organizing itself in some orderly manner.

4.3.1 Empirical Observation: Layer-Wise Entropy Convergence

The various layers' entropy evolution over several forward learning steps is showcased in Figure 2. The main highlights are:

- **Layer-Specific Minima:** Local minima for every layer is associated with local minimum entropy value at particular forward steps (K) after which the value increases gently followed by stabilization.
- **Convergence Toward a Common Equilibrium:** The entropy for layers 2, 3, and 4 approaches a constant value approximately at step $K = 49$ which appears to play a critical equilibrium role.
- **Slow Convergence of Layer 1:** This layer seems to behave similarly when it comes to reducing entropy however, clearly more gradually than the other layers suggesting that they are positioned deeper in the hierarchy of knowledge structuring.

All things taken into consideration, structured entropy SKA systems are able to achieve indicate that systems have more to SKA than simply minimizing the total levels of entropy, but rather showing a clear attempt to redistribute it across every layer efficiently.

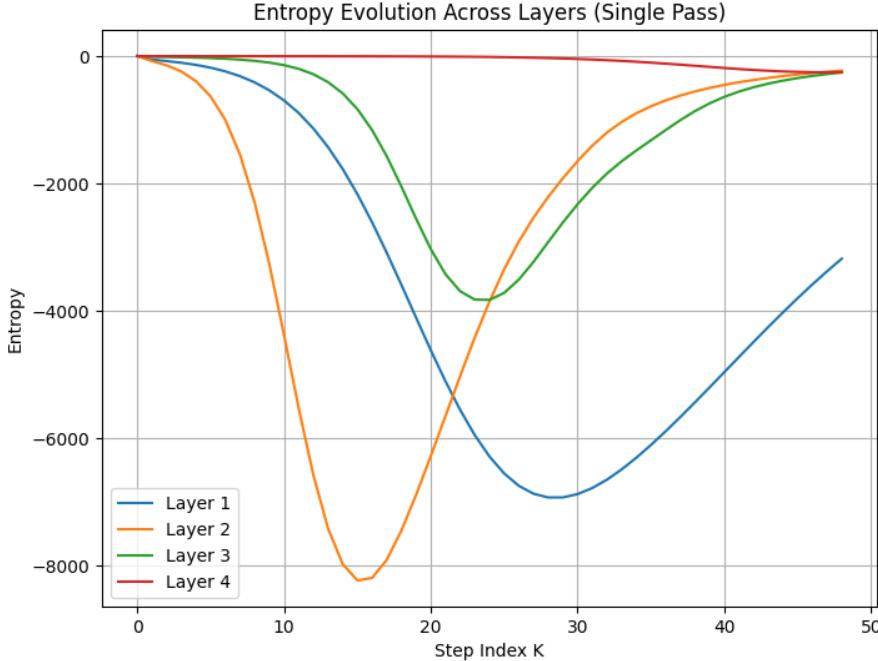


Figure 2: Entropy transformation over layers in an SKA neural network. Layers 2, 3, and 4 achieve a common entropy equilibrium $K = 49$. Layer 1 converges to this value, albeit at a more gradual rate.

4.3.2 Cosine Alignment Evolution

Besides studying entropy evolution, SKA also exhibits a remarkable phenomenon known as **cosine alignment evolution**. As stated before, Figure 3 depicts how the alignment between knowledge vectors $\mathbf{z}_k^{(l)}$ and decision probability shift $\Delta \mathbf{D}_k^{(l)}$ grows over time.

- **Cosine Alignment Convergence:** Across layers, $\cos(\theta_k^{(l)})$ converges toward a stable value which signals a growing alignment between knowledge accumulation and decision making updates.
- **Parallel Behavior to Entropy:** The exporting of cosine value consistency occurs simultaneously with entropy equilibrium, further confirming the coherent learning strategies of SKA.
- **Layer-Wise Synchronization:** Alignment convergence across layers hints towards the existence of self-organization phenomenon which would optimize knowledge updates via entropy reduction.

This provides further evidence that SKA exhibits a learning behavior independent of traditional gradient approaches within learning systems that evolve in a structured, sequential manner.

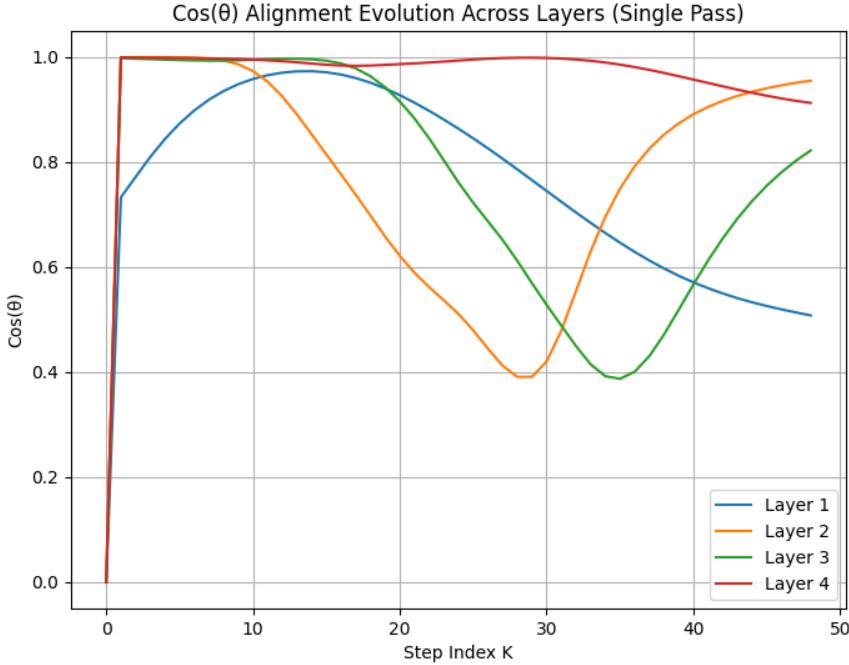


Figure 3: Cosine alignment evolution in SKA. The stabilization of the $\cos(\theta_k^{(l)})$ over steps indicates organized knowledge alignment across the layers.

The organized form of the evolution of entropy and cosine alignment offers a new framework concerning learning in neural networks which might be even more useful when attempting to explain how biological and artificial learning systems self-structure their internal representations.

4.3.3 Output Decision Probability Evolution

Alongside entropy and cosine alignment, one vital visualization in SKA is the **evolution of output decision probabilities** through forward steps. In Figure 4, we show how the mean decision probability of all 10 classes is distributed as the learning proceeds.

- **Gradual Separation of Classes:** The decision probabilities SKA achieves at the end of the learning period show that the system improves class distinguishability, which is indicative of class separability refinement, even in the absence of explicit gradient updates.

- **Emergent Stability:** The increase in the number of steps results in a stabilization of the decision probabilities, which signals the ability to reliably reach a systematized form of classification.
- **No Drastic Separation:** In contrast to classical models, SKA does not create sharp decision boundaries. SKA gradually refines knowledge build-up which incorporates soft boundaries in probabilistic terms.

This provides additional proof for SKA’s self-organizing, entropy-driven classification process, emphasizing the main difference from learning that relies on backpropagation.

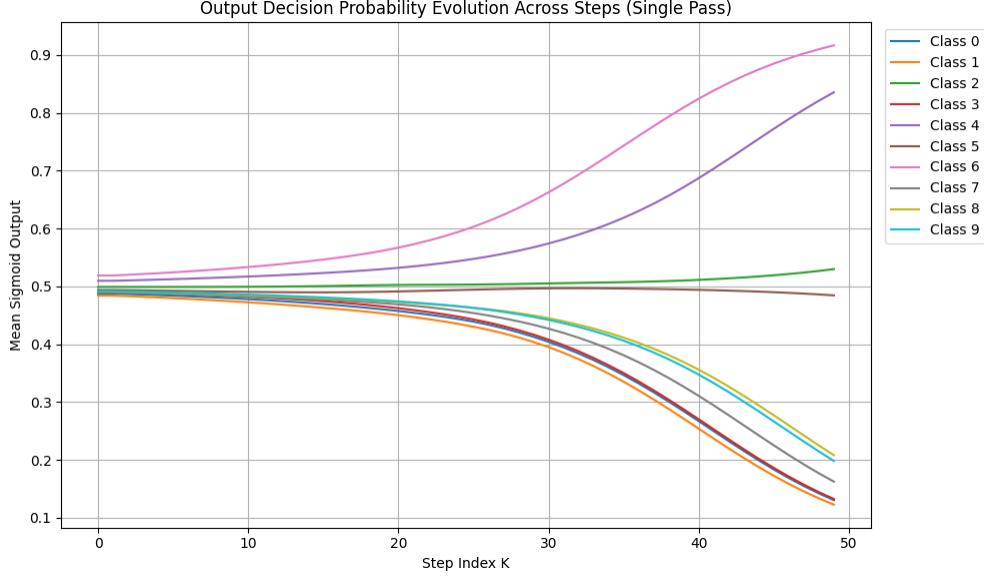


Figure 4: Change in output decision probabilities over forward steps in SKA. Distinctively from other models, SKA improves class distinctiveness incrementally without disrupting defined probability structures.

4.3.4 Frobenius Norm Evolution of the Knowledge Tensor

Apart from tracking entropy, cosine alignment, and output decision probabilities, an informative perspective in SKA is the **Frobenius norm** of the knowledge tensor $\mathbf{z}^{(l)}$. For a layer l , the definition of the Frobenius norm is:

$$\|\mathbf{z}^{(l)}\|_F = \sqrt{\sum_{i,j} (z_{ij}^{(l)})^2},$$

where $z_{ij}^{(l)}$ is the knowledge value of neuron j in sample i (before the sigmoid activation).

- **Layer-Specific Magnitude Growth:** Each layer’s Frobenius norm reflects the overall magnitude of its knowledge tensor $\mathbf{z}^{(l)}$. A larger norm indicates that the pre-sigmoid activations are more extreme, suggesting stronger or more polarized responses. Interestingly, while some layers may exhibit rapid increases in their norms, our observations show that the final layer (Layer 4) tends to grow more slowly. This gradual increase in Layer 4’s Frobenius norm suggests that, despite its role in driving the output logits, its activations remain relatively moderate—possibly indicating an early stabilization of the output during the SKA learning process.
- **Single-Pass Dynamics:** Under a single-pass, forward-only scheme, some layers may exhibit steadily increasing norms, reflecting the absence of a backward error signal that would typically constrain large activations. This effect can be particularly pronounced in the final layer, where classification logits may grow larger as the model strives to minimize local entropy.
- **Relationship to Entropy and Alignment:** While entropy and cosine alignment measure how well the knowledge tensor $\mathbf{z}^{(l)}$ aligns with the decision shifts $\Delta\mathbf{D}^{(l)}$, the Frobenius norm focuses solely on the *magnitude* of $\mathbf{z}^{(l)}$. Thus, a layer may have a large norm yet still maintain low entropy if its knowledge vectors are well-aligned with the decision shifts.

As shown in Figure 5, the knowledge tensors' Frobenius norms change across several forward steps. It's particularly interesting that layers may grow or converge at different rates, which provides information on how much each layer is "pushing" its logits to minimize local uncertainty.

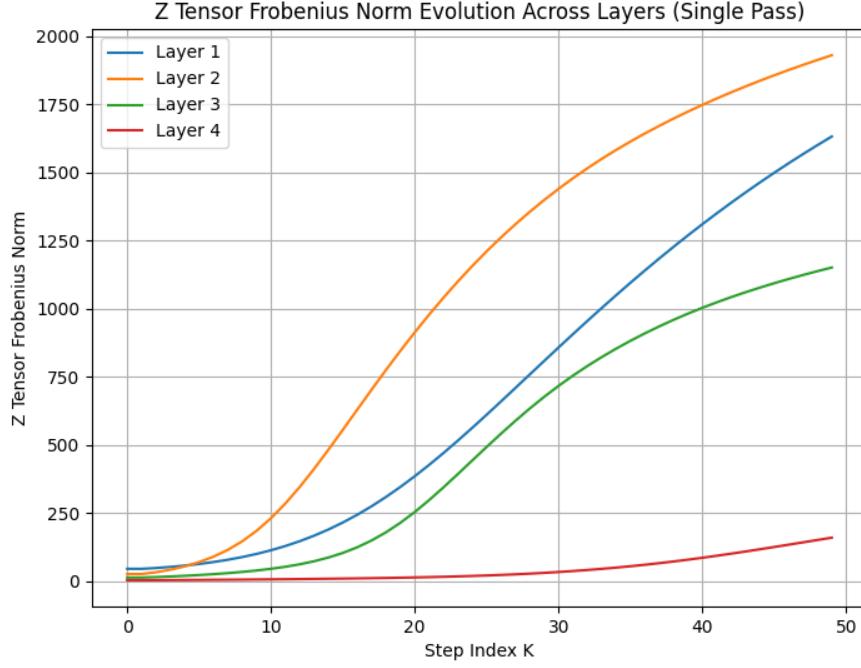


Figure 5: Frobenius norm evolution of the knowledge tensors $\mathbf{z}^{(l)}$ throughout the layers during single-pass SKA training. It is often the case that the layer with the greater norm will have stronger activations because the updates are only being done in a forward manner and are more focused on local entropy minimization.

In general, monitoring the Frobenius norm offers a new perspective on **how much magnitude** volumetric factorization captures knowledge at each layer. Combined with entropy, cosine alignment, and evolution of decision probabilities, it adds yet another dimension to how these metrics portray the self-organization of internal representations within SKA networks.

4.3.5 Entropy Trajectories

An important observation in SKA is the existence of organized entropy trajectories, especially when graphed against knowledge magnitude. Figure 6 demonstrates how reduction of entropy relates to the Frobenius norm of the knowledge tensor at different layers.

- **U-Shaped Relationship:** Every layer displays a typical U-shaped pattern, with entropy reduction exhibiting decrease with respect to knowledge, followed by an increase after reaching a minimum.
- **Progressive Shift of the Minimum:** The entropy minimum occurs at a progressively lower knowledge magnitude as we move from Layer 1 to Layer 4. This indicates that the lower layers are less efficient in utilizing knowledge magnitude to achieve entropy minimization.
- **Monotonic Knowledge Growth:** Unlike most models that impose active bounds or regularization on knowledge, SKA allows these mechanisms to arise intrinsically without any tuning needed.

This kind of behavior suggests that entropy minimization is an outcome of control over the dynamics of knowledge accumulation, which strengthens the claim of SKA being a self-organizing system.

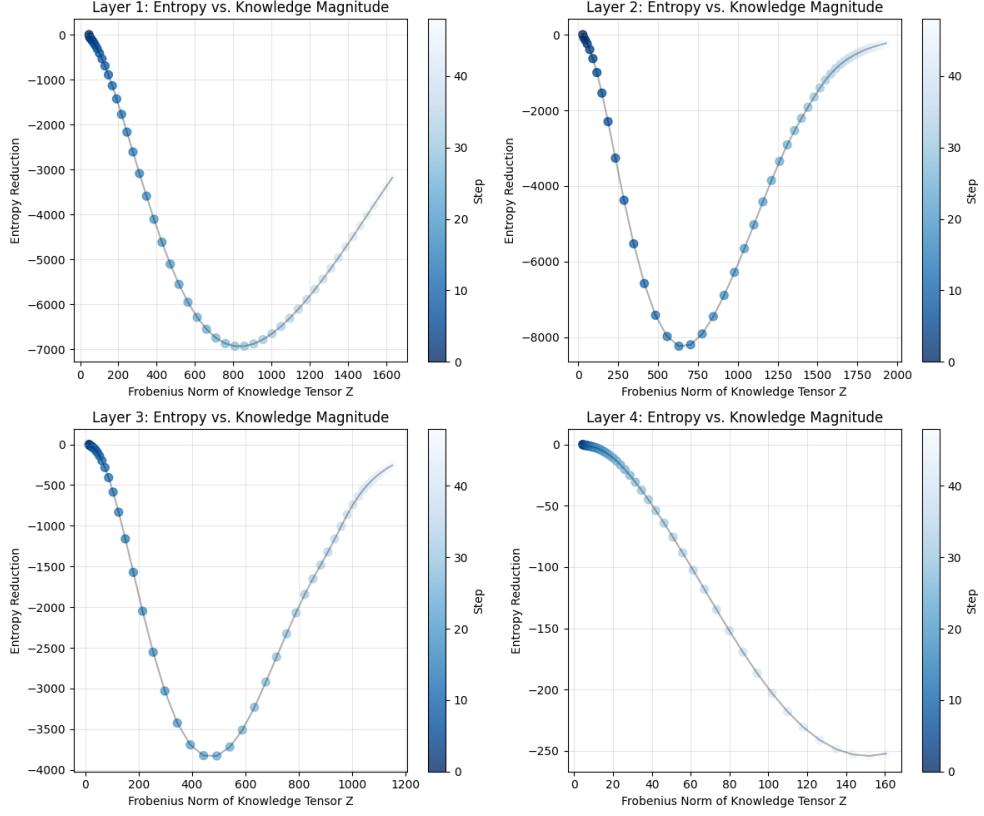


Figure 6: Entropy trajectories over the layers, tracked with respect to the Frobenius norm of the knowledge tensor. Each layer displays a U-shaped curve of entropy, with the minimum shifting towards lower values of knowledge magnitude as depth increases.

The finding demonstrates a new facet on how entropy-driven learning unfolds in SKA: each layer organizes knowledge on its own, without any external molding.

4.3.6 Theoretical Interpretation

The equilibrium of the entropy as we observed it is consistent with the SKA principle of knowledge alignment drives learning SKA inference. The structure of learning is self-organized in the following sense: the layers of knowledge are driven towards equilibrium, while accumulation of knowledge is balanced.

This indicates the possible existence of a fundamental law governing SKA-based neural networks.

In an SKA neural network, layer-wise entropy converges to an equilibrium state where knowledge accumulation stabilizes across hierarchical representations.

5 Conclusion and Future Works

The SKA framework redefines neural learning as a process of entropy-guided knowledge organization. This approach presents a foundational shift from conventional gradient-based training. By formulating entropy as a continuous, dynamic accumulation of knowledge, given as

$$H = -\frac{1}{\ln 2} \int z dD. \quad (34)$$

We have demonstrated that the sigmoid activation function emerges naturally from entropy minimization principles which provide a biologically plausible and mathematically elegant basis for learning without backpropagation.

The SKA framework enables layer-wise optimization, where each layer independently aligns knowledge vectors with decision probability shifts. This local learning dynamic not only decentralizes the training process but also enhances interpretability through angular alignment metrics such as $\cos(\theta_k^{(l)})$. Entropy progressively decreases across layers, compressing knowledge representations while maintaining information fidelity, and offering a scalable architecture suitable for real-time and resource-constrained applications.

As research on the SKA framework progresses, it holds the potential to transform the landscape of artificial intelligence by aligning learning mechanisms more closely with natural information systems. The interplay between entropy, structure formation, and local decision dynamics hints at a deeper organizational principle underlying both artificial and biological learning. In this light, SKA becomes not merely a training methodology but a paradigm through which the self-organization of intelligent behavior can be understood and replicated.

Looking ahead, future research will focus on extending the SKA framework to real-time domains such as visual and auditory processing, where continuous adaptation and forward-only computation offer tangible advantages. Comparative studies with traditional gradient-based methods on benchmark datasets will further elucidate its performance and generalization capabilities. The inherent interpretability of entropy-guided alignment also opens promising directions for transparent AI, where knowledge flow within networks can be tracked and understood. Furthermore, exploring its applicability in distributed systems, neuroscience-inspired architectures, and unsupervised learning could unlock new dimensions in scalable and interpretable machine intelligence.

Data Availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Competing Interests

The authors declare that there are no competing interests.

Author Contributions

The author confirms sole responsibility for all aspects of this work, including conceptualization, methodology, analysis, writing, and manuscript preparation.

Ethics Declaration

Not applicable.

References

- [1] Claude E. Shannon, "A Mathematical Theory of Communication," *Bell System Technical Journal*, vol. 27, pp. 379–423, 623–656, 1948. Available at: <https://people.math.harvard.edu/ctm/home/text/others/shannon/entropy/entropy.pdf>
- [2] Y. Shalev, A. Painsky, and I. Ben-Gal, "Neural Joint Entropy Estimation," *IEEE Transactions on Neural Networks and Learning Systems*, 2022. Available at: <https://ieeexplore.ieee.org/document/9308347>
- [3] Juyang Weng and Matthew D. Luciw, "Brain-Inspired Concept Networks: Learning Concepts from Cluttered Scenes," *IEEE Intelligent Systems*, 2022. Available at: <https://ieeexplore.ieee.org/document/6979240>
- [4] Seungsu Kim, Ashwini Shukla, and Aude Billard, "Catching Objects in Flight," *IEEE Transactions on Robotics*, 2014. Available at: <https://ieeexplore.ieee.org/document/6875958>
- [5] Alexei A. Koulakov, Sergey Shubaev, Divyansha Lachi, and Anthony Zador, "Encoding Innate Ability through a Genomic Bottleneck," *bioRxiv*, 2023. Available at: <https://www.biorxiv.org/content/10.1101/2023.01.15.524099v1>
- [6] Gal Shalev et al., "Neural estimation of joint entropy using mutual information neural estimation," *arXiv preprint arXiv:2203.11345*, 2022.
- [7] Himel Das Gupta, "From Symbolic Reasoning to Object Embeddings: Advanced Approaches of Knowledge Distillation in Compacted Neural Networks," *PhD Thesis*, 2024.

- [8] Yinghao Xu, Xin Dong, Yudian Li, and Hao Su, "A main/subsidiary network framework for simplifying binary neural networks," *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, pp. 7154–7162, 2019.
- [9] Y. Shalev, A. Painsky, and I. Ben-Gal, "Neural Joint Entropy Estimation," *IEEE Trans. on Neural Networks and Learning Systems*, vol. 33, no. 2, pp. 728–738, 2022. doi: 10.1109/TNNLS.2021.3056276.
- [10] Yixuan Qiu, Lingsong Zhang, and Xiao Wang, "Unbiased Contrastive Divergence Algorithm for Training Energy-Based Latent Variable Models," *Int. Conf. on Learning Representations*, 2019. Available at: <https://arxiv.org/abs/1909.07973>
- [11] Masafumi Oizumi, Shun-ichi Amari, Toru Yanagawa, Naotaka Fujii, and Naotsugu Tsuchiya, "Measuring Integrated Information from the Decoding Perspective," *PLOS Computational Biology*, vol. 11, no. 12, pp. e1004464, 2015. doi: 10.1371/journal.pcbi.1004464.
- [12] Jianwen Xie, Yang Lu, Song-Chun Zhu, and Ying Nian Wu, "A Theory of Generative ConvNet," *Int. Conf. on Machine Learning*, 2017. Available at: <https://arxiv.org/abs/1608.04211>
- [13] Angus Leung, Dror Cohen, Bruno van Swinderen, and Naotsugu Tsuchiya, "Integrated Information Structure Collapses with Anesthetic Loss of Conscious Arousal in *Drosophila melanogaster*," *PLOS Computational Biology*, vol. 17, no. 2, pp. e1008722, 2021. doi: 10.1371/journal.pcbi.1008722.
- [14] Daniel Toker, Friedrich T. Sommer, and Daniele Marinazzo, "Information Integration in Large Brain Networks," *PLOS Computational Biology*, vol. 15, no. 2, pp. e1006807, 2019. doi: 10.1371/journal.pcbi.1006807.
- [15] Timm Hess, Eli Verwimp, Gido M. van de Ven, and Tinne Tuytelaars, "Knowledge Accumulation in Continually Learned Representations and the Issue of Feature Forgetting," *arXiv preprint arXiv:2304.00933*, 2023.
- [16] A. Eguchi, J. Isbister, N. Ahmad, and S. Stringer, "The Emergence of Polychronization and Feature Binding in a Spiking Neural Network Model," *Psychological Review*, vol. 125, no. 4, pp. 545–568, 2018. doi: 10.1037/rev0000108.
- [17] J. H. Lagergren, J. T. Nardini, R. E. Baker, M. J. Simpson, and K. B. Flores, "Biologically-Informed Neural Networks Guide Mechanistic Modeling from Sparse Experimental Data," *PLOS Computational Biology*, vol. 16, no. 12, pp. e1008462, 2020. doi: 10.1371/journal.pcbi.1008462.
- [18] Yixuan Qiu, Lingsong Zhang, and Xiao Wang, "Unbiased Contrastive Divergence Algorithm for Training Energy-Based Latent Variable Models," *Int. Conf. on Learning Representations*, 2019. Available at: <https://arxiv.org/abs/1909.07973>
- [19] Xiaohui Xie and H. Sebastian Seung, "Equivalence of Backpropagation and Contrastive Hebbian Learning in a Layered Network," *Neural Computation*, vol. 15, no. 2, pp. 441–454, 2003. doi: 10.1162/089976603762552988.
- [20] Aaron Defazio, Francis Bach, and Simon Lacoste-Julien, "SAGA: A Fast Incremental Gradient Method," *Advances in Neural Information Processing Systems*, vol. 27, pp. 1646–1654, 2014. Available at: <https://proceedings.neurips.cc/paper/2014/file/ede7e1b7e17a4a7e2c4d4bf1c8455e3a-Paper.pdf>
- [21] Claude E. Shannon and Warren Weaver, "The Mathematical Theory of Communication," *University of Illinois Press*, 1949.
- [22] Hongzhou Lin, Julien Mairal, and Zaid Harchaoui, "Catalyst Acceleration for First-order Convex Optimization," *Journal of Machine Learning Research*, vol. 17, no. 1, pp. 1–54, 2015. Available at: <http://jmlr.org/papers/v17/15-535.html>
- [23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep Residual Learning for Image Recognition," *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 770–778, 2016. doi: 10.1109/CVPR.2016.90.