Delving Deeper Into Astromorphic Transformers

Md Zesun Ahmed Mia, *Student Member, IEEE*, Malyaban Bal, Abhronil Sengupta, *Senior Member, IEEE*

***Abstract*—Preliminary attempts at incorporating the critical role of astrocytes—cells that constitute more than 50% of human brain cells—in brain-inspired neuromorphic computing remain in infancy. This paper seeks to delve deeper into various key aspects of neuron-synapse-astrocyte interactions to mimic self-attention mechanisms in Transformers. The cross-layer perspective explored in this work involves bioplausible modeling of Hebbian and presynaptic plasticities in neuron-astrocyte networks, incorporating effects of non-linearities and feedback along with algorithmic formulations to map the neuron-astrocyte computations to self-attention mechanism and evaluating the impact of incorporating bio-realistic effects from the machine learning application side. Our analysis on sentiment and image classification tasks (IMDB and CIFAR10 datasets) highlights the advantages of Astromorphic Transformers, offering improved accuracy and learning speed. Furthermore, the model demonstrates strong natural language generation capabilities on the WikiText-2 dataset, achieving better perplexity compared to conventional models, thus showcasing enhanced generalization and stability across diverse machine learning tasks.**

***Index Terms*—Astromorphic Transformer, Tripartite Synapse, Hebbian Plasticity, Presynaptic Plasticity, Self-Attention**

I. Introduction

ASTROCYTES, a type of glial cell, play a critical role in brain function, encompassing various processes such as homeostasis, metabolism, and synaptic regulation [1]. Astrocytes detect and regulate synaptic activity in the tripartite synapse through interactions with pre- and postsynaptic neurons. Investigating their impact on neural computation is currently an active research field in neuroscience and underscores the critical need to move beyond the neuro-synaptic perspective of current Artificial Intelligence (AI) systems. Recent experimental findings on neuron-astrocyte interactions and modulation have led to significant progress in computational neuroscience, enabling the development of models that incorporate neuron-astrocyte interactions within neural networks [2], [3]. Astrocytes have been found to modulate bursting in neural circuitry through the release of gliotransmitters, which have an impact on neuronal excitability and synaptic plasticity [4], [5]. Astrocytes possess the ability to encode information through calcium signaling and regulate information processing, thereby actively engaging in neural computation at the tripartite synapse level. Additionally, astrocytes possess inherent capacity as memory components [6], [7] and plasticity regulators that are capable of facilitating local sequential learning [8], [9].

Preliminary attempts at incorporating the critical role of astrocytes in brain-inspired neuromorphic computing have started in earnest. Neuromorphic systems [10] are bioplausible AI models that attempt to emulate neuronal and synaptic operations with a higher degree of bio-fidelity to tap into the huge energy efficiency of the brain. Such systems are temporal and event-driven, where computation occurs through transmission of sparse spiking events between neurons, thereby offering the potential of orders of magnitude power efficiency when implemented on neuromorphic platforms [11]. However, astromorphic computing that encapsulates the features of astrocytic regulation in neuromorphic systems still remains in infancy, with works focusing primarily on the integration of astrocyte functionality with Spike-Timing-Dependent-Plasticity (STDP) mechanisms [12]–[15]—an unsupervised learning framework that remains limited to shallow networks and simple tasks [16]. Other attempts also suffer from the pressing issue of scalability to complex tasks [17].

This work is based on a recent study [18] that has shown promise at emulating the self-attention mechanism in transformers [19] using a simple two-layered network involving neuron-astrocyte interactions. The ability to design Astromorphic Transformers is therefore highly relevant in the current era of Large Language Models (LLMs), such as BERT [20] and GPT [21], paving the pathway for designing explainable transformers from a neuroscience perspective as well as addressing scaling challenges for utilizing the computational role of astrocytes in neuromorphic computing. However, as we illustrate in this work, prior work has neglected various key aspects of *bi-directional astrocytic signalling, feedback, and non-linear temporal behavior*—all of which are key to addressing the fundamental mapping of self-attention mechanism to astrocyte-neuron interactions along with providing insights on whether such intrinsic non-linear dynamics and plasticity modulation enable better learning capabilities from the application side. The distinguishing feature of this work, therefore, lies in the **neuroscience-algorithm-application co-design** perspective seeking to answer the following questions: **(i) Theoretical neuroscience:** What aspects of astrocyte functionality should be included in neural architectures to better explain the self-attention mechanism in transformers? **(ii) Algorithm**: Can we map key algorithmic modules, like the relative positional information encoding of tokens, to astrocytic interactions by drawing qualitative inspiration from theoretical neuroscience? **(iii) Application:** What is the impact of incorporating such dynamic temporal bioplausible transformers from the machine learning application perspective?

II. Related Works and Main Contributions

Spiking neural networks (SNNs), inspired by biological neurons, have garnered attention for their ability to process temporal data efficiently through sparse, event-driven computation and communication [22], [23]. Recent works have

MZA Mia, M. Bal and A. Sengupta are with the School of Electrical Engineering and Computer Science, The Pennsylvania State University, University Park, PA 16802, USA. E-mail: zesun.ahmed@psu.edu, mjb7906@psu.edu, sengupta@psu.edu.

explored enhancing the computational efficiency of SNNs, including incorporation of advanced learning techniques [25], improving network robustness through hybrid models [26], and addressing energy efficiency challenges [27], [28]. While these advances contribute to energy-efficient and robust learning systems, they remain focused on spike-based computation and do not fully leverage neuron-astrocyte interactions, which is the core focus of our work.

Understanding the role of astrocytes in neuromorphic computing, especially from the algorithm perspective, is relatively nascent. While hardware implementation of astrocyte functionality in CMOS [29], [30] as well as post-CMOS technologies [31] have been explored, algorithm-level works exploiting the role of astrocytes and leveraging their computational role have largely remained limited to self-repair functionality of faulty neural network hardware [12], [14], [32], [33]. These works are mainly inspired by the fact that astrocytes, through their synaptic plasticity regulation and feedback mechanisms, enable neurons to fire at a baseline ideal frequency. This property can be utilized to mitigate neuromorphic hardware non-idealities with stuck-at-faults and drift effects [14]. However, such approaches are still limited to single-layered unsupervised networks where astrocyte functionality is integrated with STDP learning mechanisms. Recent studies have also examined the influence of astrocytes on learning [9], working memory [6], energy minimization in neural networks [34], robotic locomotion control [35], structure learning [17], among others.

Building upon recent works [18] at emulating self-attention module in transformers through astrocyte-neuron interactions, we develop a computational framework that encapsulates key aspects of temporal non-linearities and feedback enabled by astrocytes and underscore the importance of incorporating such bioplausible effects through a comprehensive evaluation of Astromorphic Transformers on a benchmark evaluation set. Our work expands the computational capabilities of astrocyte-neuron network architectures equipped with plasticity, for the first time, to a broad category of sequential tasks.

III. Methodology

A. Transformers

Vanilla transformers [19] for classification tasks usually consist of an encoder block processing the provided *key*(K), *query*(Q), and *value*(V) generated from the input token embeddings X (words in language tasks, pixels in vision tasks, etc.). Appendix A in the supplementary material illustrates the conventional configuration of a transformer, featuring a block referred to as "Multi-Head Attention" that produces the contextual association among the provided inputs. The multi-head attention block in transformers uses a self-attention mechanism for multiple input heads, allowing the model to allocate varying importance to individual tokens, focusing on salient features for precise output in specific contexts.

The transformer implements auto-regressive attention by computing the dot product of *key* and *query* and then multiplying the resulting scalar with the corresponding *value* derived from the input token. Implementing the softmax function (a type of distribution function) atop the dot product of *key* and *query* (to enforce attention over the tokens) increases the order of complexity, a predicament that can be alleviated by the use of linearized attention [36]. Linearized attention exploits the distributive property of matrix multiplication, resulting in lower complexity for self-attention computation. More details can be found in Appendix B of the supplementary material. From the linear attention-based transformer (linear transformer) perspective [37], the transformer equations can be defined in the following form:

$$\mathrm{SA}(X) = \frac{\phi(Q)(\phi(K)^T V)}{\phi(Q)[\sum_{t=1}^{t=N} \phi(k_t)]^T} \tag{1}$$

$$Y = \mathrm{LayerNorm}(\mathrm{SA}(X) + X) \tag{2}$$

$$\mathrm{logits} = \mathrm{Softmax}(\mathrm{Linear}(\mathrm{LayerNorm}(\mathrm{FFN}(Y) + Y))) \tag{3}$$

From Equations 2 and 3, we can see that the output of self-attention is Y, which undergoes layer normalization after being augmented with residual connections from the input layer. The transformer determines the final output probabilities, denoted by logits. FFN stands for a Feed Forward Network and there is a $\phi(.)$ feature map that corresponds to a positive similarity function. When two such functions are multiplied, the normalized dot product maps to a distribution function [38] (*i.e.*, softmax). According to [37], $\phi(.)$ can be defined by Equation 4. Here, the activation function includes $elu(x)$, which stands for exponential linear unit and is utilized for efficient and precise learning [39].

$$\phi(x) = \mathrm{elu}(x) + 1 \tag{4}$$

In the next section, we discuss the tripartite synapse model and aspects that will be necessary in constructing Astromorphic Transformers.

B. Tripartite Synapse Dynamics

Astrocytes play a crucial role in the modulation of synaptic plasticity by releasing chemical signals that regulate the efficacy of synaptic connections. This phenomenon plays a role in modulating the strength and duration of synaptic connectivity and is crucial for maintaining the health of neurons. The depiction of the tripartite synapse in Fig. 1, which incorporates astrocytes, is a biologically plausible representation that embodies a multifaceted and evolving comprehension of synaptic signaling within the brain.

1) Astrocyte dynamics: The tripartite synapse's dynamics are based on Li and Rinzel Ca^{2+} dynamics [40], gatekeeper model [41], Nadkarni and Jung model [42], [43], and bidirectional study on astrocytic glutamate binding to postsynaptic neurons (AN model) [44]. According to the biophysical models, it is observed that an action potential (AP) is generated by the presynaptic axon through the release of neurotransmitters. Neurotransmitters bind to postsynaptic dendrites, causing depolarization. This leads to the release of 2-Arachidonoylglycerol (2-AG) from postsynaptic neurons to type 1 Cannabinoid Receptors (CB1Rs) on the presynaptic terminal. This process is called Depolarization-induced Suppression of Excitation (DSE) and depresses the continuous

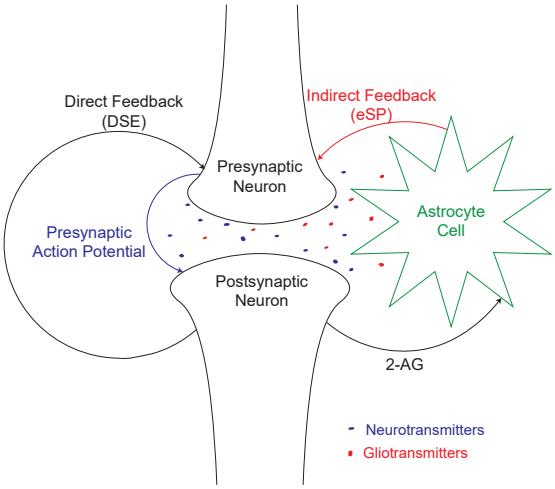

Fig. 1. A model of synaptic communication in the brain. The tripartite synapse consists of presynaptic neurons, postsynaptic neurons, and astrocytes. Astrocytes detect neuronal activity and respond bidirectionally by emitting gliotransmitters, thereby modulating the intensity and duration of synaptic communication.

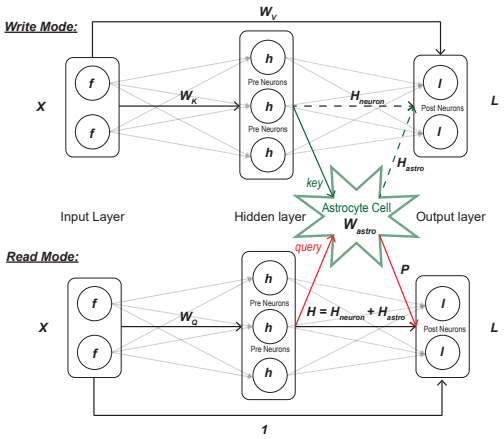

Fig. 2. The neural network architecture showing the three layers. As tokens are presented to the network as a d dimensional vector, there are d neurons in both the input and output layers. The hidden layer has m neurons. Solid lines indicate active operations; therefore, H_{neuron} and H_{astro} are learned during write mode but utilized during read mode.

release of neurotransmitters by inhibiting presynaptic neurons [45]. Neurotransmitter release also involves binding 2-AG to CB1Rs on an astrocyte, enveloping the synapse, and increasing Inositol 1,4,5-trisphosphate (IP_3) levels. IP_3 facilitates the release of calcium ions into the cytoplasm, thereby increasing calcium concentration. Three currents, $J_{channel}$, J_{leak}, and J_{pump}, are introduced to model the rate of change of calcium concentration inside the astrocyte [40]. When calcium concentration crosses a threshold, astrocytes discharge glutamate, which binds to presynaptic group I metabotropic Glutamate Receptors (mGluRs), creating a process called eSP (Indirect Feedback) from the astrocyte to the presynaptic neuron [46]. The comprehensive description of the computational model (Equation 5 introduces the basic equation of the model) can be found in the works of [44], [47], [48], which are outlined in Appendix D in the supplementary material.

$$\frac{d(Ca^{2+})}{dt} = J_{channel} + J_{leak} - J_{pump} \tag{5}$$

2) Neuron dynamics: In this work, we consider the passive Leaky Integrate and Fire (LIF) dynamics to describe the neuron firing process, which is described in Equation 6:

$$\tau_m \frac{dv}{dt} = -v(t) + R_m \sum_{i=1}^{M} I^i_{total-syn} \tag{6}$$

where τ_m represents the membrane time constant, v represents the membrane potential, and R_m represents the membrane resistance. $I^i_{total-syn}$ denotes the total synaptic current at the i^{th} synapse. M is the number of synapses connecting to the corresponding neuron. Total synaptic current at the i^{th} synapse can be defined as the summation of two currents, namely, Synaptic Current, I^i_{syn} and Slow Inward Current, $SIC(t)$ in Equation 7:

$$I^i_{total-syn} = I^i_{syn} + SIC(t) \tag{7}$$

I^i_{syn} is the intrasynaptic current, which varies linearly with the amount of neurotransmitters present in that synapse at any time, t ($I^i_{syn} = A_{se} y_i(t)$, where A_{se} and $y_i(t)$ represent the absolute synaptic efficacy and the amount of neurotransmitter discharged by the synapse at a given time t, respectively). Thus, I_{syn} *depends on the direct feedback (corresponds to* DSE*)* as direct feedback is modulated by the amount of neurotransmitters between the pre- and postsynaptic neuron. On the other hand, the extrasynaptic *Slow Inward Current* ($SIC(t)$) *is generated when astrocyte-released glutamate interacts with extrasynaptic NMDARs* (located in postsynaptic neurons) [49]. To elaborate, the activation of astrocytes leads to the release of glutamate from these cells. Upon its release, glutamate interacts with NMDARs located on the postsynaptic neuron, resulting in the emergence of a distinct form of current known as Slow Inward Current ($SIC(t)$) [44]. The details of the mathematical modeling of $SIC(t)$ can be found in [44]. Next, we develop a network architectural framework to mimic the self-attention mechanism in transformers using the astrocyte and neuron dynamics described herein and evaluate the impact of inherent non-linearities and feedback.

C. Neural Network Emulating Self-Attention

Self-attention is a fundamental component of all transformer models. Based on prior works of mimicking self-attention by astrocytes [18], we adopt a network topology as shown in Fig. 2. The neural network architecture comprises three distinct layers, namely input layer, hidden layer, and output layer. There exist two distinct operational modes for the aforementioned neural network: (i) write mode and (ii) read mode. Upon presentation of the tokens ($x_t \in R^{1\times d}$: which may include word embeddings and image embeddings) to the input layer, the write mode is initially employed, followed by the read mode to generate the outputs. In write mode, we encode information into the network by means of *keys* and *values*, whereas, in read mode, we retrieve the information by providing *queries*.

During the write mode, the input layer sequentially transmits the *keys* ($k_t = x_t W_K$; $W_K \in R^{d\times m}$) for each token to the

hidden layer and the *values* ($v_t = x_t W_V;\ W_V \in R^{d\times d}$) to the output layer. The hidden layer neurons introduce a non-linearity within the system by activating a feature map, denoted as $\phi(.)$, on the transmitted *keys*. The use of $\phi(.)$ reduces the computational complexity and memory usage inspired from the linearized attention mechanism. More information on the reduction of the order of complexity is discussed in Appendix B (with Fig. S2 illustrating the idea) in the supplementary material. A presynaptic neuron in the hidden layer ($h_t \in R^{1\times m}$) is intricately connected with an astrocyte and an output layer ($l_t \in R^{1\times d}$) neuron, forming a tripartite synapse. Similar to the mechanism by which the presynaptic neuron stimulates the astrocyte through the release of neurotransmitters, in this context, the astrocyte is being activated through the transmission of *keys* ($k_t \in R^{1\times m}$) from neurons in the hidden layer. The write mode can be mathematically expressed as in Equation 8:

$$\begin{aligned} f_t &= x_t \\ h_t &= \phi(f_t W_K) = \phi(k_t) \\ l_t &= f_t W_V = v_t \end{aligned} \tag{8}$$

In order to establish the read mode equations, it is necessary to first formulate the neural plasticities that occur among the components of the tripartite synapse. Moreover, our work aims at exploring strong bioplausible correlations in modeling such plasticities, which ultimately influence our neuroscience-algorithm-application co-design analysis. In a subsequent section, we will examine the read mode, which is responsible for retrieving the encoded information.

D. Mathematical Formulation of Plasticity

1) Hebbian Plasticity: Hebbian plasticity is characterized by the sequential adjustment of synaptic weights in accordance with the interconnection pattern between two neurons (pre- and postsynaptic neurons) at the synapse. Previous studies [18] have exclusively focused on the plasticity occurring between two neurons, neglecting non-linearities and the plasticity that may occur between astrocytes and postsynaptic neurons, which is also involved in the learning process.

Contribution I: In the context of Hebbian plasticity, we underscore the importance of the addition of astrocyte-postsynaptic neuron connected Hebbian-inspired learning along with inherent non-linearities. To elaborate, the synaptic weights (H_{neuron}) that establish connections between the presynaptic and postsynaptic neurons are acquired through Hebbian plasticity during the write mode. This learning occurs due to the connection between pre- and postsynaptic neurons, which can be mapped to the synaptic current (I_{syn}) from the neuroscience perspective. As discussed previously, I_{syn} is modulated by *DSE*, so H_{neuron} is inspired from the *direct feedback* in the tripartite synapse. This leads to Hebbian plasticity between the hidden layer (presynaptic neuron) and the output layer (postsynaptic neuron). Hidden neuron outputs are denoted as $h_t = \phi(k_t)$, whereas the output (last) layer outputs are denoted as $l_t = v_t$ in the write mode (refer to Equation 8). H_{neuron} weights are learned via Equation 9, where m acts as a scaling factor affecting the learning speed.

$$H_{neuron,t} = H_{neuron,t-1} + \frac{1}{m} h_t^T l_t \tag{9}$$

The astrocyte functions as the third constituent in the tripartite synapse, enabling bidirectional communication with the synaptic neurons. Hence, we propose that the connection between the astrocyte and the postsynaptic neuron also be represented in this formulation through a learnable weight matrix (H_{astro}), in accordance with the Hebbian rule. Equation 10 shows the learning mechanism of these weights in which the presynaptic neuron output (h_t) is replaced by the astrocytic activity, represented by w_{astro}, as this connection incorporates the astrocyte and the postsynaptic neuron. The formulation of w_{astro} will be considered in the succeeding text.

$$H_{astro,t} = H_{astro,t-1} + \frac{1}{m} w_{astro} l_t \tag{10}$$

The weight, denoted as H_{astro}, is inspired from the extrasynaptic current SIC (refer to the tripartite synapse dynamics section). According to Equation 7, SIC is added to I_{syn} to generate the total synaptic current. Consequently, the two distinct learnable weights, denoted as H_{neuron} and H_{astro}, are combined to produce the Hebbian weight $H = H_{neuron} + H_{astro}$.

Astrocytes have a significant impact on the Hebbian weight due to their inherent non-linearity. A recent study by [17] introduced an AstroNet model that optimizes neural network connections through temporal and global regulation mechanisms. The authors found that astrocytes introduce non-linear effects, which compound the pre-existing non-linearity of neurons. The concept of Hebbian plasticity is associated with temporal and global regulation, where the current weight is influenced by past neuron weights and astrocytic weights. To map the effect of non-linearity, we consider that the summation of neuron and astrocyte Hebbian weights undergoes non-linear activation, ultimately generating the Hebbian weight (H). As such, we propose the integration of a sigmoid (σ) non-linearity into the learning process. Thus, the weight H exhibits the collective Hebbian plasticity of a tripartite synapse, as elucidated by Equation 11, where K and V correspond to *keys* and *values* of all the N tokens in a single sequence (in matrix form: $K \in R^{N\times m}$, $V \in R^{N\times d}$). Assuming initial $H_{neuron,0}$ and $H_{astro,0}$ from Equations 9 and 10 are both zero, we can define the Hebbian weight in its matrix form ($H \in R^{m\times d}$) as:

$$\begin{aligned} H &= \sigma(H_{neuron} + H_{astro}) \\ H &= \sigma(\frac{1}{m}(\phi(K)^T V + W_{astro} V)) \end{aligned} \tag{11}$$

Next, we consider formulation of the astrocytic activity parameter W_{astro} (Note, w_{astro} in Equation 10 is the per-token representation and W_{astro} in Equation 11 is the matrix representation). We note that astrocytic Ca^{2+} dynamics is a relatively slower temporal process than the streaming rate of the input tokens. Therefore, it is expected that the astrocytic activity should encode relative information among successive tokens. In this context, we derive inspiration from algorithmic

formulations of relative positional information of tokens, popularly used in Vision transformers [50], [51], and map it to the activity parameter W_{astro}. The astrocytic activity parameter is therefore defined to use the edge of the tokens to capture relative position differences between input elements [50]. The edge of the tokens considers the pairwise relationships between input elements in the sense that the input is modeled as a labeled, directed, fully connected graph. To model astrocytic activity, we first define the edges between consecutive input tokens i and j as a_{ij} with the help of matrix $P \in R^{N\times N}$, containing the distance-based relative positional information between tokens, and $D \in R^{m\times m}$ which is a transformed version of P, achieved through a matrix basis transformation $D = MPM^T$. In this transformation, $M \in R^{m\times N}$ is a learnable matrix that projects the original N-dimensional space of positional relationships into an m-dimensional space. This transformation allows the model to learn an optimal representation of positional information, potentially reducing dimensionality if $m < N$, and capturing complex relationships between token positions. Mathematically, this transformation is a linear transformation, specifically a quadratic form, suitable for encoding pairwise interactions. Conceptually, this can be viewed as a learned change of basis, where M defines a new coordinate system that optimizes the representation of positional information for the task. The resulting matrix D encodes the spatial relative distances among tokens, represented as neuronal activity received by the astrocyte. Based on D, the edges are formulated as: $a_{ij} = W_{rel}D$, where $W_{rel} \in R^{N\times m}$ is a randomly initialized learnable weight matrix. To further enhance the efficiency of our model, we set $W_{rel} = M^T$. This design choice offers several advantages: First, it reduces the number of learnable parameters, promoting more efficient training and mitigating the risk of overfitting, especially in data-limited scenarios. Second, it encourages the learning of a more compact and generalizable representation, as the transformation M now serves the dual purpose of both transforming the positional information (P to D) and weighting the transformed information for edge computation ($a_{ij} = M^T D$). Finally, this constraint acts as a form of regularization, guiding the model to learn more robust and meaningful features by ensuring that the learned transformation is useful for both representing and processing positional relationships. The parameter, m, representing the number of neurons in the hidden layer, is crucial as it dictates the dimensionality of the learned representation. Its value is not fixed but rather can be adjusted based on the complexity of the dataset, allowing the model to adapt its capacity. Datasets with higher complexity may benefit from a larger m, providing the model with more representational power, while simpler datasets might learn effectively with a smaller m, promoting efficiency and potentially preventing overfitting. Building on the definition of edges, the astrocytic activity parameter $W_{astro} \in R^{m\times N}$ ($w_{astro} \in R^{m\times 1}$ for per token form) is defined as:

$$W_{astro} = \phi(a_{ij})^T \tag{12}$$

where ϕ introduces a non-linearity to capture the temporal dynamics of astrocytes. While W_{rel} is learned over time, the positional information provided by D ensures that the model maintains awareness of relative distances among tokens, allowing W_{astro} to adapt dynamically to the input. Consequently, $H_{astro} \in R^{m\times d}$ can be considered as a representation of acquired positional knowledge derived from the input tokens.

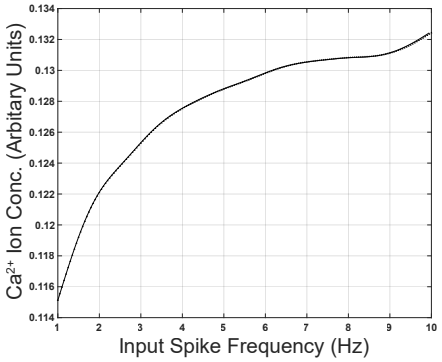

Fig. 3. As the input spike frequency in the presynaptic neurons increases, the rate of increase in calcium concentration slows down, thereby exhibiting a non-linear relationship between Ca^{2+} ion concentration and input spike rate.

2) Presynaptic Plasticity: The plasticity resulting from the induced calcium response inside the astrocyte due to bidirectional communication between the astrocyte and presynaptic neurons (hidden layer) can be defined as presynaptic plasticity. As tokens are sequentially presented to the hidden layer during the write mode, *keys*: $\phi(k_t)$ are generated, which can be mapped to an increasing number of released neurotransmitters (implying increment in $J_{channel}$ as per the computational neuroscience model), thereby resulting in elevation of the calcium concentration within the astrocyte (evident from Equation 5). Thus, the presynaptic plasticity parameter g can be mapped to calcium concentration in astrocytes modulated by indirect feedback (eSP), and can be expressed as the aggregate of the inputs ($h_t = \phi(k_t)$) received by the presynaptic neuron. Equation 13 defines the formulation of the presynaptic plasticity parameter ($g_t, g \in R^{1\times m}$). Assuming the initial calcium concentration to be zero ($g_0 = 0$), g can be expressed as:

$$\begin{aligned} g_t &= g_{t-1} + h_t = g_{t-1} + \phi(k_t) \quad \text{(pertoken form)} \\ g &= \sum_{t=1}^{N} \phi(k_t) \qquad\qquad\qquad \text{(matrix form)} \end{aligned} \tag{13}$$

Contribution II: Prior work [18] at modelling pre-synaptic plasticity has not considered the intrinsic non-linear Ca^{2+} dynamics of the astrocytes in response to increasing firing rate of presynaptic neurons. Due to the indirect feedback (*eSP*), the rate of increase in the release rate of neurotransmitters gradually reduces in response to an increase in presynaptic neuron firing over time. Hence, a non-linear relationship exists between the calcium concentration and the frequency of spikes received from the presynaptic neuron. We substantiate this observation by simulating the AN model [44], [47]. The associated non-linear variation is depicted in Fig. 3 which

leads us to modify the presynaptic plasticity parameter g as,

$$g = (\sum_{t=1}^{N} \phi(k_t))^{\alpha} \tag{14}$$

where the exponent α encodes the degree of non-linearity. From our computational modeling, we use a value of $\alpha = 0.25$. The subsequent section elaborates on the read mode operation, which involves the process of retrieving the information encoded during the write mode to implement the self-attention mechanism incorporating our key contributions towards more bioplausible formulations of Hebbian and presynaptic plasticity.

E. Realization of Self-Attention

The self-attention mechanism is realized when the encoded information is accessed during the read mode to obtain the activations of the output layer neurons. The hidden layer neurons receive *queries* ($q_t = x_t W_Q$) from the input layer and access the information that was previously encoded within the astrocyte. This information is stored via both the Hebbian weight (H) and the presynaptic plasticity parameter (g) during write mode. However, during read mode, H is already learned and not modulated by the transmitted *queries* from the presynaptic neurons, as it is directly connected to the postsynaptic neurons (output layer). As g is the presynaptic plasticity parameter, it has no direct connection to the postsynaptic neuron. Therefore, the information encoded in g has to be retrieved by the *queries* transmitted from the hidden layer during read mode by inducing a calcium response from the astrocytes. In order to read the calcium response ($c_t \in R^{1\times 1}$) of the astrocyte, the *queries* (h_t) are scaled by the corresponding presynaptic plasticity parameters (g_t) across all the m hidden neurons and then summed together. This is depicted in Equation 15, where the matrix form of the calcium response is defined by $C \in R^{N\times 1}$.

$$\begin{aligned} c_t &= \sum_m g_t h_t \quad \text{(pertoken form)} \\ C &= hg^T \qquad\quad \text{(matrix form)} \end{aligned} \tag{15}$$

Let us define the weight modulating the connection between the astrocyte and the output layer as P. According to studies regarding Tumor Necrosis Factor - α ($TNF-\alpha$), astrocytes possess the ability to boost synaptic weights in reaction to low neural activity and diminish weights in reaction to high neural activity [18] by means of calcium response (neural activity is proportional to released neurotransmitters). Consequently, it is possible to infer an inverse correlation between P and the calcium response (C). Simplifying, this relation ($P = \frac{1}{C}$) is given by Equation 16. To avoid the computational cost of a full matrix inversion, we compute the reciprocal calcium response element-wise by inverting only the non-zero elements of the diagonal matrix using a mask. This ensures numerical stability and computational efficiency for large-scale sequences.

$$P = \frac{1}{\phi(Q)[(\sum_{t=1}^{N} \phi(k_t))^{\alpha}]^T} \tag{16}$$

Given that the presynaptic plasticity modulated weight (P) is ascertained, we can define the network equations for the read mode. As seen from Fig. 2, the hidden layer and the output layer are connected by the Hebbian weight H, and the astrocyte is imposing another learned weight (from calcium response) denoted by P onto the existing weight H. Thus the weights are implemented by the Hadamard (element-wise: $\odot$) multiplication of H and P onto the output layer. Furthermore, the output layer receives the input (F) directly from the input layer modulated by unity weight to implement the residual connection present in the self-attention module of transformers. Considering the network architecture, the read mode network equations can be defined in Equation 17. It is important to highlight here that during the read mode, the decoding process occurs in a singular timestep, as opposed to the sequential nature of the write mode. Hence, all the parameters for input, hidden, and output layers are represented in their matrix format (e.g., $X \in R^{N\times d}$, $h \in R^{N\times m}$, $L \in R^{N\times d}$).

$$\begin{aligned} F &= X \\ h &= \phi(FW_Q) = \phi(Q) \\ L &= h(H \odot P) + F = \phi(Q)(H \odot P) + X \end{aligned} \tag{17}$$

In Equation 17, all the parameters are in matrix mode, where the dimensions are defined as follows: $W_Q \in R^{d\times m}$ and $Q \in R^{N\times m}$ (*queries*). L represents the generated output from the last layer. The self-attention algorithm is defined when the network completes a write and read mode, as discussed next.

By substituting the values of H and P from Equations 11 and 16 into Equation 17, the resulting Equation 18 can be obtained after the write and read mode:

$$\begin{aligned} L &= \frac{\phi(Q)(\sigma(\phi(K)^T V + W_{astro} V))}{m \times \phi(Q)[(\sum_{t=1}^{N} \phi(k_t))^{\alpha}]^T} + X \\ Y &= \text{Layernorm}(L) \\ \text{logits} &= \text{Softmax}(\text{Linear}(\text{FFN}(Y) + Y)) \end{aligned} \tag{18}$$

In Equation 18, L implements an identical operation as the transformer self-attention from Equation 1 (along with the inclusion of an extra residual connection from the input layer). Y is the layer-normalized version of L and is further feedforwarded through fully connected linear layers to finally obtain the output probabilities in the form of logits. Thus, with this implementation, the Astromorphic Transformer is formulated. The detailed implementation of this process, including neuron-astrocyte interactions and synaptic modulation, is provided in Appendix E in the supplementary material, where the pseudo-code outlines the sequence of operations governing synaptic plasticity and calcium-mediated attention adjustments in the Astromorphic Transformer.

IV. Results

Our Astromorphic Transformer is trained and tested on three different tasks: sentiment classification on the IMDB dataset [65], image classification on the CIFAR10 dataset [66], and language modeling on the Wikitext-2 dataset [67]. To date, there has not been any language or vision model that incorporates transformer architecture in a bioplausible manner derived from neuron-astrocyte interactions. We report the following performance for each task:

TABLE I
PERFORMANCE COMPARISON OF OUR PROPOSED ASTROMORPHIC ATTENTION-BASED TRANSFORMER ARCHITECTURE AGAINST OTHER BENCHMARKS WITH VARYING DEGREES OF BIOPLAUSIBILITY ON THE IMDB, CIFAR10 AND WIKITEXT-2 DATASETS.

Task	Model	Implementation	Bio-plausibility	Accuracy(%)/ Perplexity
Sentiment Classification (IMDB Dataset)	Memristor SNN [52]	Directly trained in SNN domain	Yes	84.86%
	Memristor SNN [52]	Converted from ANN	Yes	85.88%
	ANN [52]	No spiking activity	No	86.02%
	Spike-transformer [53]	Rate-coded approach using spiking neurons	Yes	86.36%
	Recurrent SNN [54]	Presynaptic current-based backpropagation	Yes	86.82%
	HRNaEMSA biGRU [55]	Implements multi-headed attention	No	87.3%
	HRNaEMSA biLSTM [55]	Implements multi-headed attention	No	87.4%
	CoRNN [56]	Coupled oscillators modeled as neurons	No	87.4%
	LSTM [57]	Memristor crossbars mimicking NN	No	87.48%
	SNN [58]	Compute-in-memory macro using SNN	Yes	88.15%
	UniCoRNN [59]	Solves vanishing gradient problem in BPTT	No	88.38%
	Vanilla Transformer	Softmax attention without bioplausibilities	No	$88.9 \pm 0.1\%$
	Linearized Transformer inspired from astrocytes (Iso-architecture) [18]	Linear attention with no astrocytic non-linearity or relative positional encoding	Yes	$88.4 \pm 0.1\%$
	Astromorphic Transformer (Our Model)	Implements astromorphic attention with non-linearity and relative positional information	Yes	$\mathbf{88.7 \pm 0.2}\%$
Image Classification (CIFAR10 Dataset)	BioLeaF [60]	SNNs trained by BP-based rules	Yes	86.88%
	STBP in CIFARNET [61]	Spatio-temporal backprop	No	89.83%
	VGG16-SNN [62]	ANN-SNN conversion	Yes	91.55%
	MS-ResNet [63]	SNN oriented ResNet-110 architecture	No	91.72%
	ANN [64]	ResNet-19 (12.63M parameters)	No	94.97%
	ANN [64]	Transformer-4-384 (9.32M parameters)	No	96.73%
	Spikformer [64]	Spikformer-4-384 (9.32M parameters)	Yes	95.19%
	Spikingformer [61]	Spikingformer-4-384 (9.32M parameters)	Yes	95.61%
	Vanilla Transformer	Softmax attention without bioplausibilities	No	$97.4 \pm 0.1\%$
	Linearized Transformer inspired from astrocytes (Iso-architecture) [18]	Linear attention with no astrocytic non-linearity or relative positional encoding	Yes	$96.8 \pm 0.2\%$
	Astromorphic Transformer (Our Model)	Implements astromorphic attention with non-linearity and relative positional information	Yes	$\mathbf{97.0 \pm 0.2}\%$
Language Modeling (Wikitext-2 Dataset)	LSTM [67]	Traditional RNN-based language model	No	65.9
	AWD-LSTM [70]	Standard LSTM with dropout, regularization	No	58.8
	AWD-LSTM-MoS [71]	Improved LSTM for language modeling	No	63.88
	AWD-FWM [72]	Fast weight memory-based language model	No	61.65
	Vanilla Transformer	Softmax attention without bioplausibilities	No	73.4
	Linearized Transformer inspired from astrocytes (Iso-architecture) [18]	Linear attention with no astrocytic non-linearity or relative positional encoding **(does not converge due to gradient explosion)**	Yes	123.3
	Astromorphic Transformer (Our Model)	Implements astromorphic attention with non-linearity and relative positional information	Yes	**33.8**

- $88.7 \pm 0.2\%$ for sentiment classification on the IMDB dataset, utilizing the non-linearity and relative positional information in the model.
- $97.0 \pm 0.2\%$ accuracy for image classification using an astromorphic vision transformer on the CIFAR10 dataset.
- A perplexity of 33.8 for language modeling on the Wikitext-2 dataset using one decoder layer with 6-headed self-attention.

A. Hardware Specifications & Dataset Details

The Astromorphic Transformer has been executed on a GPU-based platform running PyTorch. The hardware setup consists of a 12-core Intel(R) Xeon(R) Platinum $8167M$ CPU @ 2.00GHz, with an NVIDIA® Tesla® V100 GPU (16 GB memory).

The IMDB dataset contains $50,000$ movie reviews labeled as positive or negative, with each review spanning between 250 and 500 words. The CIFAR10 dataset comprises $60,000$ color images categorized into 10 distinct classes, with each image having dimensions of 32×32 pixels. The Wikitext-2 dataset is a widely used benchmark for language modeling and natural language generation tasks, containing over two million words with carefully preserved capitalization, punctuation, and rare terms.

B. Network Architecture

For the sentiment classification task on the IMDB dataset, we employ a transformer with an encoder-based architecture, using one encoder layer with 4-headed self-attention. A dropout layer is applied after the residual connection before layer normalization.

For the image classification task on the CIFAR10 dataset, we implement the CoAtNet transformer architecture, combining convolutional neural networks (CNNs) with self-attention. The architecture employs a 3×3 convolution before creating patches, followed by an 8-layered encoder stack with 8-headed self-attention. A patch size of 2 is used on the image inputs. In addition to the original CoAtNet architecture, we add non-linearity and average pooling after the encoder layers.

For the language modeling task on the Wikitext-2 dataset, we use a decoder-based transformer architecture with one decoder layer and 6-headed self-attention. The embeddings for this task are generated using the GPT-2 tokenizer, and they are passed through an astromorphic decoder block with layer normalization and dropout before the final output.

For all tasks, we use AdamW optimizer and Cross Entropy Loss for training. For tokenization, the GloVe pretrained word embeddings [69] are used with 840B tokens, 2.2M vocab (cased), and 300D vectors for the IMDB dataset, and for CIFAR10 dataset, pixels are converted to sequences by using a patch size of 2, while GPT-2 tokenizers are used for the Wikitext-2 dataset. All hyperparameters for these architectures are detailed in Appendix C.

C. Performance Evaluation and Comparative Analysis

Table I compares the performance of our Astromorphic Transformer against both bioplausible neuromorphic models and non-bioplausible models on the IMDB, CIFAR10, and Wikitext-2 datasets. We chose not to compare our results against pretrained transformer models like BERT and GPT, as these architectures leverage pretraining on large datasets and use non-bioplausible self-attention mechanisms with highly customized learnable embeddings. In contrast, our model is trained from scratch, making it simpler to implement the astromorphic attention in the transformer framework while still achieving competitive results. The results, summarized in Table I, demonstrate the model's effectiveness compared to both bioplausible and non-bioplausible architectures.

1) Sentiment Classification on IMDB: On the IMDB dataset, the Astromorphic Transformer achieves an accuracy of 88.7%, outperforming several traditional models, including RNNs, LSTMs, and SNNs. It also exceeds the accuracy of our implementation of Kozachkov *et al.*'s proposal [18]: Linearized Transformer inspired from astrocytes by 0.3%, which lacks the astrocytic non-linearity and relative positional encoding central to our model's design.

2) Image Classification on CIFAR10: In the CIFAR10 dataset, the Astromorphic Transformer attains an accuracy of 97.0%, outperforming the Linearized Transformer inspired from astrocytes [18] which achieves 96.8%. Both models deliver high performance, but the Astromorphic Transformer shows a slight edge, benefiting from enhanced feature learning enabled by astrocytic non-linearity and positional encoding, especially in complex image classification tasks.

3) Language Modeling on Wikitext-2: For language modeling, the Astromorphic Transformer demonstrates superior performance with a perplexity of 33.8 on the Wikitext-2 dataset. In contrast, *Linearized Transformer inspired from astrocytes [18] fails to converge due to gradient explosion*, yielding a perplexity of 123.3. This stark difference highlights the necessity of incorporating astrocytic non-linearity for stabilizing training and improving model generalization in sequential tasks.

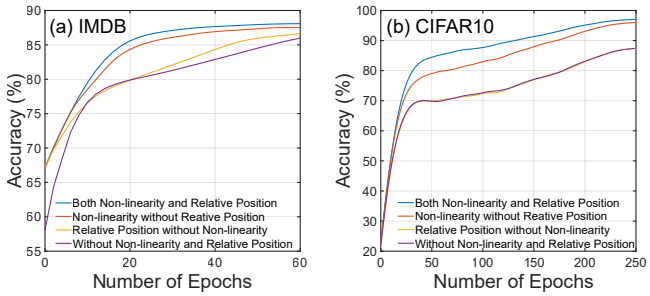

Fig. 4. Comparative analysis to examine the effects of incorporating non-linearity and relative positional encoding in the Astromorphic Transformer on the (a) IMDB and (b) CIFAR10 datasets. The yellow and purple lines in Fig. (b) are nearly overlapping. The purple curve represents the result of the model formulated by [18] which performs subpar in both datasets compared to our proposed Astromorphic Transformer (blue curve).

4) Ablation Studies in Learning Speed: The impact of *astrocytic non-linearity* and *relative positional encoding* on learning speed is evident in both datasets. On the IMDB dataset, the Astromorphic Transformer reaches 85% accuracy in **16±2 epochs**. Without *positional encoding*, this increases to **22±2 epochs**, and without both features, it takes **54±4 epochs**. These results reflect a **27.3% faster convergence** with positional encoding and a **70.4% improvement** when both features are included. In the CIFAR10 dataset, the model reaches 85% accuracy in **45±3 epochs**. Without *positional encoding*, it requires **111±4 epochs**, and without either feature, it stretches to **215±2 epochs**, demonstrating a **59.5% faster convergence** with positional encoding and a **79.1% speed improvement** with both features. For Wikitext-2 dataset, the Astromorphic Transformer converges to the lowest perplexity within 40 ± 5 epochs, whereas the Linearized Transformer inspired from astrocytes [18] fails to converge, and the model crashes due to gradient explosion after a few epochs. Fig. 4 makes it evident that our model significantly outperforms the Linearized Transformer inspired from astrocytes [18] in terms of both learning speed and final performance, demonstrating the crucial role of bioplausible features in improving the convergence and generalization of transformer models.

V. Conclusion

The Astromorphic Transformer advances the frontier of bioplausible computing by effectively leveraging neuron-astrocyte interactions to enhance transformer architectures. Through the integration of biologically inspired mechanisms, our model not only matches but also surpasses the efficiency and stability of existing artificial transformer models. By refining synaptic modulation with astrocytic feedback and Hebbian plasticity, we introduce a more adaptive and nuanced attention mechanism that proves advantageous in tasks such as sentiment analysis, image classification, and natural language generation.

While previous models, like the Linearized Transformer inspired from astrocytes [18], also utilize neuron-astrocyte interactions, they lack two critical features: astrocytic non-linearity and relative positional encoding. These omissions limit the model's ability to capture complex relationships in sequential data, resulting in slower convergence and reduced accuracy and, in some cases, failure to converge entirely—as observed in the Wikitext-2 task. Table II compares the vanilla transformer, linear transformer, linearized transformer inspired by astrocytes, and the proposed Astromorphic Transformer. It maps each model to relevant equations and figures, emphasizing how computational neuroscience equations underpin the bioplausible mechanisms and advancements of our approach.

TABLE II
COMPARISON OF TRANSFORMER ARCHITECTURES

Model	Description
Vanilla Transformer	Lacks all bioplausibility, employs softmax attention instead of feature maps, and has quadratic complexity for self-attention computation.
Linear Transformer [37]	Lacks bioplausibility, employs linear feature maps (replacing softmax), achieving linearized complexity for self-attention (see Fig. S2). Does not incorporate tripartite synapse-based formulation.
Linearized Transformer Inspired from Astrocytes [18]	Incorporates bioplausibility and tripartite synapse-based formulation (refer to Eqns. 8, 9, 13, 15, 17 and Fig. 1). Employs linear feature maps (no softmax) and has linearized complexity (Fig. S2). However, it lacks intricate bioplausible plasticities derived from computational neuroscience models.
Astromorphic Transformer (Our Model)	Fully bioplausible, employs linear feature maps (no softmax) with linear complexity (Fig. S2) and tripartite synapse formulation (Fig. 1). Shows a clear depiction of the astromorphic self-attention network (Fig. 2) and incorporates astrocytic non-linearity derived from calcium response (refer to Eqns. 5, 14, 16, and Fig. 3). Introduces astrocytic Hebbian plasticities (Eqns. 10, 11) encoding relative positional information (Eqn. 12) inspired by Slow Inward Currents (Eqns. 6, 7). Formulates an accurate model of the Astromorphic Transformer, as detailed in Eqn. 18.

Moreover, our Astromorphic Transformer is catered for simple implementation in current hardware without any additional overhead by leveraging the decomposition of astromorphic attention into distinct write and read modes, making it feasible for traditional computing architectures. Detailed molecular-level computational modeling is not required for practical implementation. Instead, macro-models developed for non-linearity, plasticity, and feedback processes are being utilized, which represent these dynamics in a simplified yet biologically plausible way. These macro-models (see Equation 18) should be easily integrable into conventional hardware, facilitating efficient deployment without needing highly specialized configurations. Energy-efficient in-memory processing with CMOS or emerging post-CMOS devices for transformer architectures [76] can also be potentially utilized for hardware implementation of Astromorphic Transformers. The Astromorphic Transformer can further be improved by incorporating more complex dynamics from neuroscience, such as astrocyte-astrocyte communication, which could enable more sophisticated processing and parallelization of plasticity mechanisms. Additionally, scaling the model to deeper architectures while leveraging characteristics like memory and learning from astrocytes may allow the model to adapt to more complex tasks with long-context sequences.

ACKNOWLEDGMENTS

Research was sponsored in part by the Army Research Office and was accomplished under Grant Number W911NF-24-1-0127. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Office or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein. This work was also supported partially by the National Science Foundation under award No. EFRI BRAID #2318101.

REFERENCES

[1] N. A. Perez-Catalan, C. Q. Doe, and S. D. Ackerman, "The role of astrocyte-mediated plasticity in neural circuit development and function," *Neural Development*, vol. 16, no. 1, pp. 1–14, 2021.

[2] T. Manninen, J. Aćimović, and M.-L. Linne, "Analysis of network models with neuron-astrocyte interactions," *Neuroinformatics*, vol. 21, no. 2, pp. 375–406, 2023.

[3] N. M. Wakida, V. Gomez-Godinez, H. Li, J. Nguyen *et al.*, "Calcium dynamics in astrocytes during cell injury," *Frontiers in Bioengineering and Biotechnology*, vol. 8, p. 912, 2020.

[4] C. Durkee, P. Kofuji, M. Navarrete, and A. Araque, "Astrocyte and neuron cooperation in long-term depression," *Trends in Neurosciences*, vol. 44, no. 10, pp. 837–848, 2021.

[5] K. Lenk, E. Satuvuori, J. Lallouette, A. Ladrón-de Guevara, H. Berry, and J. A. Hyttinen, "A computational model of interactions between neuronal and astrocytic networks: the role of astrocytes in the stability of the neuronal firing rate," *Frontiers in computational neuroscience*, vol. 13, p. 92, 2020.

[6] Y. Tsybina, I. Kastalskiy, M. Krivonosov, A. Zaikin *et al.*, "Astrocytes mediate analogous memory in a multi-layer neuron–astrocyte network," *Neural Computing and Applications*, vol. 34, no. 11, pp. 9147–9160, 2022.

[7] S. Gordleeva, Y. A. Tsybina, M. I. Krivonosov, I. Y. Tyukin *et al.*, "Situation-based memory in spiking neuron-astrocyte network," *arXiv preprint arXiv:2202.07218*, 2022.

[8] M. E. Gibbs, D. Hutchinson, and L. Hertz, "Astrocytic involvement in learning and memory consolidation," *Neuroscience & Biobehavioral Reviews*, vol. 32, no. 5, pp. 927–944, 2008.

[9] C. M. Alberini, E. Cruz, G. Descalzi, B. Bessières, and V. Gao, "Astrocyte glycogen and lactate: New insights into learning and memory mechanisms," *Glia*, vol. 66, no. 6, pp. 1244–1262, 2018.

[10] A. Sengupta, Y. Ye, R. Wang, C. Liu, and K. Roy, "Going deeper in spiking neural networks: VGG and residual architectures," *Frontiers in neuroscience*, vol. 13, 2019.

[11] M. Davies, A. Wild, G. Orchard, Y. Sandamirskaya *et al.*, "Advancing neuromorphic computing with loihi: A survey of results and outlook," *Proceedings of the IEEE*, vol. 109, no. 5, pp. 911–934, 2021.

[12] J. Liu, L. J. McDaid, J. Harkin, S. Karim, A. P. Johnson, A. G. Millard, J. Hilder, D. M. Halliday, A. M. Tyrrell, and J. Timmis, "Exploring self-repair in a coupled spiking astrocyte neural network," *IEEE transactions on neural networks and learning systems*, vol. 30, no. 3, pp. 865–875, 2018.

[13] M. Rastogi, S. Lu, N. Islam, and A. Sengupta, "On the Self-Repair Role of Astrocytes in STDP Enabled Unsupervised SNNs," *Frontiers in Neuroscience*, vol. 14, p. 603796, Jan. 2021. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fnins.2020.603796/full

[14] Z. Han, N. Islam, and A. Sengupta, "Astromorphic self-repair of neuromorphic hardware systems," *arXiv preprint arXiv:2209.07428*, 2022.

[15] J. Liu, J. Harkin, L. P. Maguire, L. J. McDaid, and J. J. Wade, "Spanner: A self-repairing spiking neural network hardware architecture," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 4, pp. 1287–1300, 2017.

[16] P. U. Diehl and M. Cook, "Unsupervised learning of digit recognition using spike-timing-dependent plasticity," *Frontiers in computational neuroscience*, vol. 9, p. 99, 2015.

[17] M. Han, L. Pan, and X. Liu, "Astronet: When astrocyte meets artificial neural network," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 20 258–20 268.

[18] L. Kozachkov, K. V. Kastanenka, and D. Krotov, "Building transformers from neurons and astrocytes," *Proceedings of the National Academy of Sciences*, vol. 120, no. 34, p. e2219150120, 2023.

[19] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[20] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[21] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," 2018.

[22] M. Pfeiffer and T. Pfeil, "Deep learning with spiking neurons: Opportunities and challenges," *Frontiers in Neuroscience*, vol. 12, p. 774, 2018.

[23] A. Tavanaei, M. Ghodrati, S. R. Kheradpisheh, T. Masquelier, and A. Maida, "Deep learning in spiking neural networks," *Neural Networks*, vol. 111, pp. 47-63, 2019.

[24] K. Roy, A. Jaiswal, and P. Panda, "Towards spike-based machine intelligence with neuromorphic computing," *Nature*, vol. 575, no. 7784, pp. 607-617, 2019.

[25] F. Zenke and T. P. Vogels, "Superspike: Supervised learning in multilayer spiking neural networks," *Neural Computation*, vol. 30, no. 6, pp. 1514–1541, 2018.

[26] H. Mostafa, "Supervised learning based on temporal coding in spiking neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 7, pp. 3227–3235, 2017.

[27] S. Yang, H. Wang, and B. Chen, "Sibols: robust and energy-efficient learning for spike-based machine intelligence in information bottleneck framework," *IEEE Transactions on Cognitive and Developmental Systems*, 2023.

[28] S. Yang and B. Chen, "SNIB: improving spike-based machine learning using nonlinear information bottleneck," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2023.

[29] M. Ranjbar and M. Amiri, "An analog astrocyte–neuron interaction circuit for neuromorphic applications," *Journal of Computational Electronics*, vol. 14, no. 3, pp. 694–706, 2015.

[30] G. Karimi, M. Ranjbar, M. Amirian, and A. Shahim-Aeen, "A neuromorphic real-time vlsi design of ca2+ dynamic in an astrocyte," *Neurocomputing*, vol. 272, pp. 197–203, 2018.

[31] U. Garg, K. Yang, and A. Sengupta, "Emulation of astrocyte induced neural phase synchrony in spin-orbit torque oscillator neurons," *Frontiers in neuroscience*, vol. 15, 2021.

[32] J. Wade, L. McDaid, J. Harkin, V. Crunelli, and S. Kelso, "Self-repair in a Bidirectionally Coupled Astrocyte-Neuron (AN) System based on Retrograde Signaling," *Frontiers in Computational Neuroscience*, vol. 6, 2012. [Online]. Available: https://www.frontiersin.org/article/10.3389/fncom.2012.00076

[33] M. Rastogi, S. Lu, N. Islam, and A. Sengupta, "On the Self-Repair Role of Astrocytes in STDP Enabled Unsupervised SNNs," *Frontiers in Neuroscience*, vol. 14, p. 1351, 2021.

[34] A. Fountain and C. Merkel, "Effect of biologically-motivated energy constraints on liquid state machine dynamics and classification performance," *Neuromorphic Computing and Engineering*, vol. 2, no. 2, p. 024005, 2022.

[35] I. Polykretis, G. Tang, and K. P. Michmizos, "An astrocyte-modulated neuromorphic central pattern generator for hexapod robot locomotion on intel's loihi," in *International Conference on Neuromorphic Systems 2020*, 2020, pp. 1–9.

[36] I. Schlag, K. Irie, and J. Schmidhuber, "Linear transformers are secretly fast weight programmers," in *International Conference on Machine Learning*. PMLR, 2021, pp. 9355–9366.

[37] A. Katharopoulos, A. Vyas, N. Pappas, and F. Fleuret, "Transformers are rnns: Fast autoregressive transformers with linear attention," in *International Conference on Machine Learning*. PMLR, 2020, pp. 5156–5165.

[38] S. Chaudhari, V. Mithal, G. Polatkan, and R. Ramanath, "An attentive survey of attention models," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 12, no. 5, pp. 1–32, 2021.

[39] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (elus)," *arXiv preprint arXiv:1511.07289*, 2015.

[40] Y.-X. Li and J. Rinzel, "Equations for InsP3 receptor-mediated Ca^{2+} i oscillations derived from a detailed kinetic model: a Hodgkin-Huxley like formalism," *Journal of theoretical Biology*, vol. 166, no. 4, pp. 461–473, 1994.

[41] V. Volman, E. Ben-Jacob, and H. Levine, "The astrocyte as a gatekeeper of synaptic information transfer," *Neural computation*, vol. 19, no. 2, pp. 303–326, 2007.

[42] S. Nadkarni and P. Jung, "Dressed neurons: modeling neural–glial interactions," *Physical biology*, vol. 1, no. 1, p. 35, 2004.

[43] ——, "Modeling synaptic transmission of the tripartite synapse," *Physical biology*, vol. 4, no. 1, p. 1, 2007.

[44] J. J. Wade, L. J. McDaid, J. Harkin, V. Crunelli, and J. S. Kelso, "Bidirectional coupling between astrocytes and neurons mediates learning and dynamic coordination in the brain: a multiple modeling approach," *PloS one*, vol. 6, no. 12, p. e29445, 2011.

[45] B. E. Alger, "Retrograde signaling in the regulation of synaptic transmission: focus on endocannabinoids," *Progress in neurobiology*, vol. 68, no. 4, pp. 247–286, 2002.

[46] M. Navarrete and A. Araque, "Endocannabinoids potentiate synaptic transmission through stimulation of astrocytes," *Neuron*, vol. 68, no. 1, pp. 113–126, 2010.

[47] J. Wade, L. J. McDaid, J. Harkin, V. Crunelli, and S. Kelso, "Self-repair in a bidirectionally coupled astrocyte-neuron (AN) system based on retrograde signaling," *Frontiers in computational neuroscience*, vol. 6, p. 76, 2012.

[48] M. De Pittà, M. Goldberg, V. Volman, H. Berry, and E. Ben-Jacob, "Glutamate regulation of calcium and ip 3 oscillating and pulsating dynamics in astrocytes," *Journal of biological physics*, vol. 35, pp. 383–411, 2009.

[49] P. G. Haydon and G. Carmignoto, "Astrocyte control of synaptic transmission and neurovascular coupling," *Physiological reviews*, vol. 86, no. 3, pp. 1009–1031, 2006.

[50] P. Shaw, J. Uszkoreit, and A. Vaswani, "Self-attention with relative position representations," *arXiv preprint arXiv:1803.02155*, 2018.

[51] P. Ramachandran, N. Parmar, A. Vaswani, I. Bello, A. Levskaya, and J. Shlens, "Stand-alone self-attention in vision models," *Advances in neural information processing systems*, vol. 32, 2019.

[52] J. Huang, A. Serb, S. Stathopoulos, and T. Prodromakis, "Text classification in memristor-based spiking neural networks," *Neuromorphic Computing and Engineering*, vol. 3, no. 1, p. 014003, 2023.

[53] E. Mueller, V. Studenyak, D. Auge, and A. Knoll, "Spiking transformer networks: A rate coded approach for processing sequential data," in *2021 7th International Conference on Systems and Informatics (ICSAI)*. IEEE, 2021, pp. 1–5.

[54] Z. Wang, Y. Zhang, H. Shi, L. Cao, C. Yan, and G. Xu, "Recurrent spiking neural network with dynamic presynaptic currents based on backpropagation," *International Journal of Intelligent Systems*, vol. 37, no. 3, pp. 2242–2265, 2022.

[55] X.-L. Leng, X.-A. Miao, T. Liu *et al.*, "Using recurrent neural network structure with enhanced multi-head self-attention for sentiment analysis," *Multimedia Tools and Applications*, vol. 80, pp. 12 581–12 600, 2021.

[56] T. K. Rusch and S. Mishra, "Coupled oscillatory recurrent neural network (cornn): An accurate and (gradient) stable architecture for learning long time dependencies," *arXiv preprint arXiv:2010.00951*, 2020.

[57] A. Aswani, R. Kumar, J. N. Tripathi, and A. James, "Performance of crossbar based long short term memory with aging memristors," in *2021 IEEE 3rd International Conference on Artificial Intelligence Circuits and Systems (AICAS)*. IEEE, 2021, pp. 1–4.

[58] A. Agrawal, M. Ali, M. Koo, N. Rathi, A. Jaiswal, and K. Roy, "Impulse: A 65-nm digital compute-in-memory macro with fused weights and membrane potential for spike-based sequential learning tasks," *IEEE Solid-State Circuits Letters*, vol. 4, pp. 137–140, 2021.

[59] T. K. Rusch and S. Mishra, "Unicornn: A recurrent model for learning very long time dependencies," in *International Conference on Machine Learning*. PMLR, 2021, pp. 9168–9178.

[60] Y. Yang and P. Li, "Bioleaf: A bioplausible learning framework for training of spiking neural networks," *arXiv preprint arXiv:2111.13188*, 2021.

[61] C. Zhou, L. Yu, Z. Zhou, H. Zhang, Z. Ma, H. Zhou, and Y. Tian, "Spikingformer: Spike-driven residual learning for transformer-based spiking neural network," *arXiv preprint arXiv:2304.11954*, 2023.

[62] G. Srinivasan, C. Lee, A. Sengupta, P. Panda, S. S. Sarwar, and K. Roy, "Training deep spiking neural networks for energy-efficient neuromorphic computing," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 8549–8553.

[63] Y. Hu, Y. Wu, L. Deng, and G. Li, "Advancing residual learning towards powerful deep spiking neural networks," *arXiv e-prints*, pp. arXiv–2112, 2021.

[64] Z. Zhou, Y. Zhu, C. He, Y. Wang, S. Yan, Y. Tian, and L. Yuan, "Spikformer: When spiking neural network meets transformer," *arXiv preprint arXiv:2209.15425*, 2022.
[65] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Portland, Oregon, USA: Association for Computational Linguistics, June 2011, pp. 142–150. [Online]. Available: http://www.aclweb.org/anthology/P11-1015
[66] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," 2009.
[67] S. Merity, C. Xiong, J. Bradbury, and R. Socher, "Pointer sentinel mixture models," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 1072–1081, 2017.
[68] Z. Dai, H. Liu, Q. V. Le, and M. Tan, "Coatnet: Marrying convolution and attention for all data sizes," *Advances in neural information processing systems*, vol. 34, pp. 3965–3977, 2021.
[69] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543. [Online]. Available: http://www.aclweb.org/anthology/D14-1162
[70] S. Takase, J. Suzuki, and M. Nagata, "Direct output connection for a high-rank language model," *arXiv preprint arXiv:1808.10143*, 2018.
[71] Z. Yang, Z. Dai, R. Salakhutdinov, and W. W. Cohen, "Breaking the softmax bottleneck: A high-rank RNN language model," *arXiv preprint arXiv:1711.03953*, 2017.
[72] I. Schlag, T. Munkhdalai, and J. Schmidhuber, "Learning associative inference using fast weight memory," *arXiv preprint arXiv:2011.07831*, 2020.
[73] N. Pappas and J. Henderson, "Deep residual output layers for neural language generation," in *International Conference on Machine Learning*, 2019, pp. 5000–5011.
[74] Wang, Chenguang, Mu Li, and Alexander J. Smola. "Language models with transformers." arXiv preprint arXiv:1904.09408 (2019).
[75] P.-Y. Chen, X. Peng, and S. Yu, "NeuroSim: A circuit-level macro model for benchmarking neuro-inspired architectures in online learning," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, no. 12, pp. 3067–3080, 2018.
[76] X. Yang, B. Yan, H. Li, and Y. Chen, "ReTransformer: ReRAM-based processing-in-memory architecture for transformer acceleration," in *Proceedings of the 39th International Conference on Computer-Aided Design*, 2020, pp. 1–9.

Md Zesun Ahmed Mia completed his bachelor's degree in Electrical and Electronic Engineering at the Bangladesh University of Engineering and Technology, Dhaka, Bangladesh in 2019, earning multiple Dean's List recognitions. From 2020 to 2021, he worked as an R&D Engineer for SEMWAVES Ltd., London, UK. Following this, he served as a Lecturer in the Department of Electrical and Electronic Engineering at the University of Liberal Arts Bangladesh, Dhaka, Bangladesh from 2021 to 2022. He earned his M.S. in Electrical Engineering from Penn State University in University Park, PA, USA in 2024, where he is currently pursuing his PhD and working as a Graduate Research Assistant within the Neuromorphic Computing Lab. His research focuses on the intersection of neuromorphic computing, spintronic devices, and bioplausible neuromorphic algorithms. He is a recipient of multiple prestigious graduate awards, including the Arthur Waynick Graduate Scholarship and the Milton and Albertha Langdon Memorial Graduate Fellowship.

Malyaban Bal earned his B.E. degree in Computer Science and Engineering from Jadavpur University, Kolkata, India, in May 2018. Following graduation, he worked as a Software Engineer at Microsoft R&D, Bangalore, India, for three years. Currently, he is pursuing the Ph.D. degree in Computer Science and Engineering at The Pennsylvania State University, University Park, PA, USA, under the supervision of Prof. Abhronil Sengupta. During his doctoral studies, he also served as a Research Intern at the Stanford Research Institute (SRI International), Menlo Park, CA, USA. His research focuses on designing brain-inspired, energy-efficient neural frameworks for sequence modeling tasks.

Abhronil Sengupta received the B.E. degree from Jadavpur University, Kolkata, India, in 2013, and the Ph.D. degree in Electrical and Computer Engineering from Purdue University, West Lafayette, IN, USA in 2018.

He worked as a DAAD (German Academic Exchange Service) Fellow at the University of Hamburg, Hamburg, Germany in 2012, and as a graduate research intern at Circuit Research Labs, Intel Labs, Hillsboro, OR, USA in 2016 and Facebook Reality Labs, Redmond, WA, USA in 2017. In 2018, he joined the School of Electrical Engineering and Computer Science at The Pennsylvania State University, University Park, PA, USA where he is an Associate Professor and holds the Joseph R. and Janice M. Monkowski Career Development Professorship. He is also affiliated with the Department of Materials Science and Engineering and the Materials Research Institute (MRI). The ultimate goal of Dr. Sengupta's research is to bridge the gap between Nanoelectronics, Neuroscience and Machine Learning. He is pursuing an inter-disciplinary research agenda at the intersection of hardware and software across the stack of sensors, devices, circuits, systems and algorithms for enabling low-power event-driven cognitive intelligence. Dr. Sengupta has published over 95 articles in referred journals and conferences and holds 3 US patents.

He has been awarded the National Science Foundation CAREER Award, Army Research Office Early Career Award, Purdue Engineering 38 by 38 Award, IEEE Electron Devices Society (EDS) Early Career Award, IEEE Circuits and Systems Society (CASS) Outstanding Young Author Award, Facebook Faculty Award, IEEE SiPS Best Paper Award, Schmidt Science Fellows Award nominee, Purdue Bilsland Dissertation Fellowship, CSPIN Student Presenter Award, Birck Fellowship from Purdue University, and the DAAD WISE Fellowship. He serves on the IEEE Circuits and Systems Society Technical Committee on Neural Systems and Applications, Editorial Board of IEEE Transactions on Cognitive and Developmental Systems, Scientific Reports, Neuromorphic Computing and Engineering, Frontiers in Neuroscience journals and the Technical Program Committee of several international conferences. Dr. Sengupta is a Senior Member of the IEEE, Senior Member of the Association for Computing Machinery (ACM), and member of the American Physical Society (APS) and the Association for the Advancement of Artificial Intelligence (AAAI). He currently serves as an ACM Distinguished Speaker (2024-2027) and IEEE CASS Distinguished Lecturer (2025-2026).

SUPPLEMENTARY MATERIAL

A. Appendix A: Transformer Illustration

Transformer employs only encoder architecture for performing sentiment classification and vision tasks. As illustrated in Fig. S1, the input embeddings, combined with the positional embeddings, enter the encoder layer as input tokens. Then *key*, *query*, and *value* are generated from the input tokens, and after performing self-attention on them, a skip connection is added from the inputs, and the overall result is layer normalized and fed into a feed-forward network. Another add & normalize operation is performed before feeding them into a linear layer and softmax function to obtain the output probabilities. Sometimes, there might be more than one encoder layer. For different tasks, like machine translation and text generation, encoder-decoder or only decoder-type architecture of transformers can be used. However, independent of the type of architecture needed, self-attention is a fundamental mechanism defining transformer operation. The "Multi-Head Attention" Block has self-attention modules parallelly running on different heads obtained from the input embeddings.

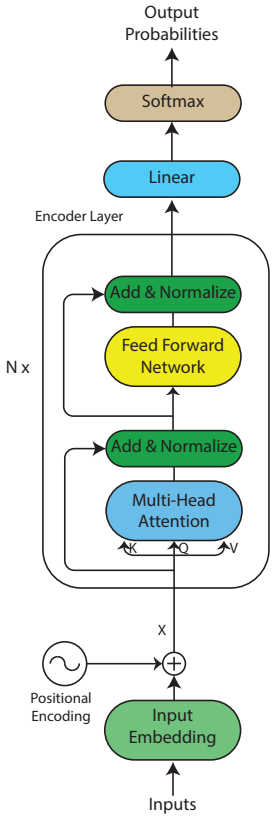

Fig. S1. Schematic representation of the Transformer Encoder architecture, illustrating the sequential flow of operations. N denotes the number of encoder layers in the network.

B. Appendix B: Memory Efficient Linearized Attention

Linearized attention involves utilizing feature maps with lower complexity levels, thereby minimizing memory and computational resources. As discussed before, the introduction of non-linearity in biological neurons is facilitated by an activation function ($\phi(.) = elu(.) + 1$) in the hidden layer neurons. The utilization of this positive similarity feature map as an activation function characterizes the implementation of the linearized attention operation as described by Equation 1. The utilization of linearized feature attention results in a reduction of computational complexity associated with the computation of the attention score. The advantage of utilizing feature map-based linearized attention over the conventional distribution function (*i.e.* softmax) based self-attention is demonstrated in Fig. S2. Usual self-attention employs softmax as the distribution function; however, the dot product of *Query* (Q) and *Key* (K) poses an $O(N^2D)$ complexity, which overuses the memory. Whereas, in linearized attention, the computation multiplies *Key* (K) and *Value* (V), yielding an $O(ND^2)$ complexity, properly utilizing the memory and computational efficiency when token length (N) is larger than internal embedding size (D)(We set token embedding dimension equal to the internal embedding size ($d = D$) in this work). Thus, the employment of the linearized feature map enables the realization of memory-efficient linearized attention.

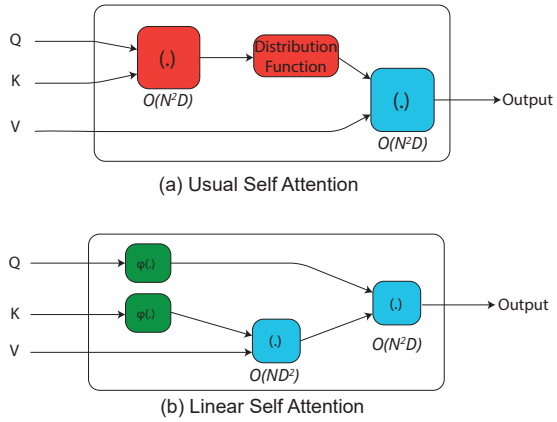

Fig. S2. Figure showing the comparison of order of complexity between usual and linearized self-attention. Distribution function can be mapped to softmax function used in usual transformers.

C. Appendix C: Hyper-parameter Details

Hyper-parameters for the encoder-type astromorphic transformer for IMDB dataset are in Table S1, for the encoder-type convolution and astromorphic self-attention-based transformer for CIFAR10 dataset are in Table S2, and for the astromorphic transformer for Wikitext-2 dataset are in Table S3.

D. Appendix D: Computational Model of Neuron-Astrocyte-Mediated Synaptic Dynamics

In the context of astromorphic computation, astrocytes play a pivotal role in modulating synaptic transmission through both direct and indirect pathways. These pathways, influenced by endocannabinoid signaling and astrocytic calcium dynamics,

TABLE S1
HYPER-PARAMETER LIST FOR THE IMDB DATASET.

Hyper-parameters	Value	Tested Range
Non-linearity parameter α	0.25	0.15-0.4
Model size D	128	32-256
Scaling parameter m	$500D$	(1-1000)D
Sequence length	400	100-600
Learning rate	0.00001	$1e^{-4}$ - $1e^{-6}$
Batch size	64	8-256
Number of epochs	120	40-150
Number of heads	4	1-12

TABLE S2
HYPER-PARAMETER LIST FOR THE CIFAR10 DATASET.

Hyper-parameters	Value	Tested Range
Non-linearity parameter α	0.25	0.15-0.4
Model size D	256	64-512
Scaling parameter m	$500D$	(1-1000)D
Patch size	2	2-4
Learning rate	0.001	$1e^{-2}$ - $1e^{-4}$
Batch size	32	8-128
Number of epochs	250	80-300
Number of heads	8	2-12

are essential for the dynamic control of synaptic weights in neuromorphic architectures such as the Astromorphic Transformer.

1) Endocannabinoid Dynamics: Endocannabinoids, such as 2-arachidonoylglycerol (2-AG), are released from the postsynaptic neuron following depolarization. They serve as retrograde messengers, modulating synaptic transmission probability through both depressive and potentiating effects. The release dynamics of 2-AG can be modeled as follows:

$$\frac{d(AG)}{dt} = -\frac{AG}{\tau_{AG}} + r_{AG} \cdot \delta(t - t_{\text{spike}})$$

where τ_{AG} is the decay constant of 2-AG, and r_{AG} is the rate of 2-AG production after a spike.

2) Direct Signaling Pathway—Depolarization-Induced Suppression of Excitation (DSE): In the direct pathway, 2-AG binds to CB1 receptors (CB1Rs) on the presynaptic neuron, reducing the synaptic transmission probability PR. This localized effect, termed Depolarization-Induced Suppression of Excitation (DSE), can be described as:

$$DSE(t) = -AG(t) \cdot K_{AG}$$

where K_{AG} is a constant representing the strength of suppression.

3) Indirect Signaling Pathway—Endocannabinoid-Mediated Synaptic Potentiation (e-SP): In the indirect pathway, 2-AG binds to CB1Rs on astrocytes, triggering intracellular calcium release from the endoplasmic reticulum (ER) via IP3 receptors (IP3Rs). This calcium release induces astrocytic glutamate release, which potentiates synaptic transmission at distant synapses. The potentiation is given by:

$$eSP(t) = m_{eSP} \cdot Glu(t)$$

where m_{eSP} is the scaling factor for synaptic potentiation, and $Glu(t)$ is the concentration of astrocytic glutamate.

TABLE S3
HYPER-PARAMETER LIST FOR THE WIKITEXT-2 DATASET.

Hyper-parameters	Value	Tested Range
Non-linearity parameter α	0.25	0.1-0.3
Model size D	768	256-1024
Scaling parameter m	$100D$	(50-500)D
Sequence length	512	128-512
Learning rate	0.00001	$1e^{-4}$ - $5e^{-6}$
Batch size	24	8-64
Number of epochs	50	20-100
Number of heads	6	4-12

4) Calcium Dynamics in Astrocytes: The astrocytic calcium response, which regulates synaptic potentiation, is governed by the following equation:

$$\frac{d[Ca^{2+}]}{dt} = J_{\text{chan}}(IP3, Ca^{2+}) + J_{\text{leak}} - J_{\text{pump}}$$

where $J_{\text{chan}}(IP3, Ca^{2+})$ represents calcium influx through IP3-regulated channels (dependent on IP3 and calcium concentration), J_{leak} accounts for passive calcium leakage from the ER, and J_{pump} describes calcium reuptake into the ER via ATP-dependent pumps. Once calcium levels exceed a threshold, the astrocyte releases glutamate, governed by:

$$\frac{d[Glu]}{dt} = -\frac{Glu}{\tau_{Glu}} + r_{Glu} \cdot H(Ca^{2+}(t) - \theta_{Ca})$$

where τ_{Glu} is the decay constant of glutamate, and θ_{Ca} is the calcium threshold that triggers release, and $H(x)$ is the Heaviside function defined as $H(x) = 1$ if $x \geq 0$ and $H(x) = 0$ if $x < 0$.

5) Modulation of Synaptic Transmission Probability (PR): The total synaptic transmission probability $PR(t)$ is influenced by both the direct depressive effect and indirect potentiation, as follows:

$$PR(t) = PR(0) + PR(0) \cdot \frac{DSE(t) + eSP(t)}{100}$$

where $PR(0)$ is the baseline synaptic transmission probability.

This appendix presents a detailed mathematical framework for neuron-astrocyte-mediated synaptic dynamics, which are key to the functionality of astromorphic computational systems.

E. Appendix E: Algorithm and Pseudo-code

The pseudo-code (Algorithm 1) represents the core steps of the astromorphic transformer, where neuron-astrocyte interactions modulate synaptic weights, enhancing the self-attention mechanism. The algorithm operates through two phases: Write Mode and Read Mode, followed by the generation of the final transformer output. In the initialization step, the weight matrices for neurons and astrocytes, the presynaptic plasticity parameter, the non-linearity parameter, and the edge matrix representing token positions are set up. During the Write Mode, keys and values are generated from the input tokens using learnable weight matrices. The activation function is applied to introduce non-linearity, resulting in an updated representation. Hebbian learning is then used to update both

neuron and astrocyte weight matrices based on the outer product of the activation and the values, while the presynaptic plasticity parameter is updated according to the sum of activated keys modulated by a non-linearity parameter. In the Read Mode, after all tokens have been processed, a query matrix is generated from the input tokens. The activation function is applied to these queries, and the astrocytic calcium response is computed. The modulation factor, which is inversely proportional to the calcium response, adjusts the synaptic weights dynamically. The final output is calculated by matrix multiplication of the synaptic weights and the activated queries along with an addition of a residual connection from the input. The final transformer output is processed through layer normalization to stabilize and accelerate training. A feed-forward network introduces additional non-linearity, followed by another layer normalization step with a residual connection to ensure gradient flow. Finally, a linear transformation and softmax function produce the final logits, representing the output probabilities of the model.

Algorithm 1 Astromorphic Transformer Algorithm

Initialization: Initialize parameters and hyper-parameters
Initialize input tokens X, token embedding dimension d, hidden layer neurons m
Initialize Hebbian weight matrix H_{neuron} and astrocyte weight matrix H_{astro} to 0
Initialize presynaptic plasticity parameter g to 0
Set non-linearity parameter, α
Set edge between input tokens x_i and x_j: a_{ij}
Write Mode: Encode keys and values from input tokens
for each token x_t in input tokens X **do**
    $k_t \leftarrow x_t W_k$
    $v_t \leftarrow x_t W_v$
    $h_t \leftarrow \phi(k_t)$
    {Hebbian learning for neuron and astrocyte plasticity}
    $H_{neuron,t} \leftarrow H_{neuron,0} + \frac{1}{m} h_t^T v_t$
    $W_{astro} \leftarrow \phi(a_{ij}^T)$
    $H_{astro,t} \leftarrow H_{astro,0} + \frac{1}{m} W_{astro} v_t$
    {Presynaptic plasticity based on calcium dynamics}
    $g \leftarrow g + (\sum(h_t))^{\alpha}$
end for
Read Mode: Retrieve information using query for all tokens
$Q \leftarrow X W_q$
$h \leftarrow \phi(Q)$
$C \leftarrow h g^T$
$P \leftarrow \frac{1}{C}$
$L \leftarrow h((H_{neuron} + H_{astro}) \odot P) + X$
$L_{norm} \leftarrow \mathrm{LayerNorm}(L)$
$L_{output} \leftarrow \mathrm{LayerNorm}(\mathrm{FFN}(L_{norm}) + L_{norm})$
$logits \leftarrow \mathrm{Softmax}(\mathrm{Linear}(L_{output}))$
return logits