This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

GliaNet: Adaptive Neural Network Structure Learning with Glia-Driven

Mengqiao Han[1] Liyuan Pan[2,*] Xiabi Liu [2]
[1] College of Information Engineering, Northwest Agricultural and Forestry (A&F) University
[2] School of Computer Science, Beijing Institute of Technology
hanmq@nwafu.edu.cn, {liyuan.pan, liuxiabi}@bit.edu.cn

Abstract

Neural networks derived from the M-P model have excelled in various visual tasks. However, as a simplified simulation version of the brain neural pathway, their structures are locked during training, causing over-fitting and over-parameterization. Although recent models have begun using the biomimetic concept and empirical pruning, they still result in irrational pruning, potentially affecting the accuracy of the model. In this paper, we introduce the Glia unit, composed of oligodendrocytes (Oli) and astrocytes (Ast), to emulate the exact workflow of the mammalian brain, thereby enhancing the biological plausibility of neural functions. Oli selects neurons involved in signal transmission during neural communication and, together with Ast, adaptively optimizes the neural structure. Specifically, we first construct the artificial Glia-Neuron (G-N) model, which is formulated at the instance, group, and interaction levels with adaptive and collaborative mechanisms. Then, we construct GliaNet based on our G-N model, whose structure and connections can be continuously optimized during training. Experiments show that our GliaNet advances state-of-the-art on multiple tasks while significantly reducing its parameters.

1. Introduction

The expansion of learnable parameters in artificial neural networks (ANNs) has contributed to their extraordinary successes [11]. However, during training, the structures of ANNs are fixed, which can induce over-fitting [5, 39] and over-parameterization [3, 15] of the network. Although network structure learning methods, including but not limited to Network Compression, Knowledge Distillation, Neural Architecture Search (NAS), and Dynamic Neural Networks (DyNN), aim to alleviate the issues mentioned above, they still yield problems such as unstable performance and expensive computational costs [6, 41, 51, 52].

*corresponding author

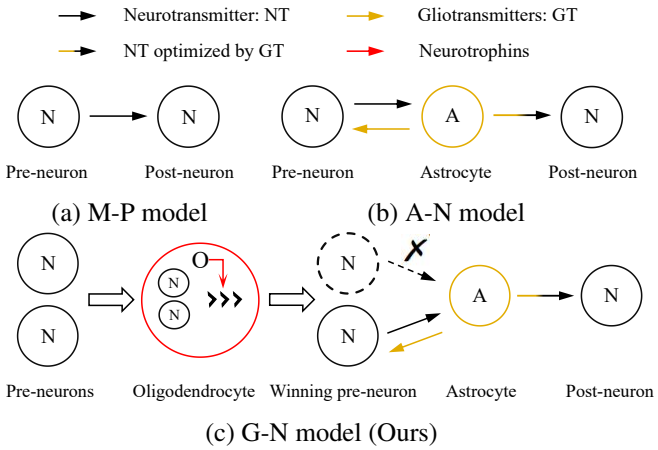


Figure 1. *Comparison of basic units of neuronal models. (a) The M-P model [33]. The connection between neurons is unidirectional propagation. (b) The Astrocyte-Neuron model [16, 17]. The connection of the pre-neuron is modulated by Ast through bidirectional communication and subsequently transmitted to the post-neuron. (c) Our G-N model. Glia, consisting of Oli and Ast, as key supportive elements for neuronal function, exhibits the optimization of the neural pathway. The winning pre-neuron, participating in subsequent propagation, is selected by Oli. Then, the winning connection is modulated by Ast. ✗ denotes the inhibitory neuron not connected to Ast and post-neuron.*

The aforementioned issues can be attributed to the M-P model [33], serving as the basis for building artificial perceptions. The M-P model is a simplified version of inter-neuronal communication that restricts neurotransmitters (formulated as connections or weights in a network) to unidirectional propagation, *i.e.*, from the pre-neuron to the post-neuron (see Fig. 1a). Several researchers have conducted preliminary explorations by incorporating an astrocyte into the nervous system to propose an artificial Astrocyte-Neuron (A-N) model [16, 17]. In this model, astrocytes facilitate connections between pre- and post-neurons through bidirectional communication with connections released by the pre-neurons (see Fig. 1b). During training, astrocytes can adaptively modulate network connections in different tasks.

However, the existing A-N model selects important connections by rigorously pruning those that are below a specified threshold following astrocyte modulation. This empirical pruning approach easily reduces accuracy when striving for a lightweight model, as evidenced in various pruning studies [29, 40].

Reflecting the mammalian brain's workflow, the brain's response to external stimuli initially manifests as the activation of neurons in a specific region, rather than engaging all neurons [1]. Thus, before the Astrocyte (Ast) modulates neurotransmitters, a set of neurons involved in neural communication—referred to as winning neurons—must first be selected [25], a task carried out by the Oligodendrocyte (Oli) [14]. Oli and Ast typically appear in pairs, forming a Glia unit. In this paper, we simulate functional interactions within the nervous system, comprising Glia and neuron, and propose a novel artificial Glia-Neuron (G-N) model. On the one hand, we redefine network structure learning by adaptively selecting winning neurons through Oli and modulating their neural connections via Ast. On the other hand, we expand the Glia unit into a group to achieve the optimal neural structure and connections by fostering the coordinated roles of Oli and Ast within the Glia units and their interaction between units. The formulation of our G-N model is followed in three aspects.

First, at the instance level, Glia influences various aspects of neuronal function, including excitability and synaptic efficacy [48]. Specifically, Oli is stimulated by neuron states and releases neurotrophins to select winning neurons involved in subsequent propagation and recalibrate their states [32]. Then, neurotransmitters released by winning neurons stimulate Ast, prompting Ast product gliotransmitters to modulate these neurotransmitters [13] (see Fig. 1c).

The fundamental unit in ANNs are neurons that reflect network activity through their states, while their connections signify the strength of associations between these states [50]. Consequently, we model a more exact workflow in brain mechanisms by simulating the interactions between Oli and neurons, as well as Ast and connections.

Second, at the group level, a Glia unit typically does not exist in isolation; instead, multiple Glia units form a cooperative group, *i.e.*, a group of Glia units, to jointly optimize neural communication [21]. This setup allows each member (*i.e.*, Oli and Ast) within a Glia unit to focus on processing in a specific area, thereby enhancing optimization efficiency. Moreover, multiple Glia units overcome a limitation of the A-N model, where a single Ast can maintain only one state per iteration, thus increasing its capacity for dynamic regulation.

Third, at the interaction level, we define the interactions of Glia units in two aspects: (i) interactions between a Glia unit across different iterations and (ii) interactions among multiple Glia units. i) The state of each Glia at time t depends on the result of its optimization of the corresponding subnetwork structure at time t-1. ii) Glia units collaborate with each other during iterations. A Glia unit is stimulated not only by states from its associated partial pre-neurons but also by implicit influences from other pre-neuron states through its interactions with other Glia units [35].

With our proposed G-N model, we construct the GliaNet, where its structure and connections can be adaptively optimized during training. Our primary contributions include:

- We formulate an artificial Glia-Neuron (G-N) model at instance, group and interaction levels with adaptive and collaborative mechanisms.
- We construct the GliaNet which adaptively optimizes the dynamically changing structure and neural connections during training based on our G-N model.
- We extensively evaluate GliaNet on multiple tasks with public datasets, *e.g.*, improving accuracy by $0.08\% \sim 2.93\%$ on classification. AP improved by $0.8\% \sim 1.0\%$ on segmentation and $0.7\% \sim 1.1\%$ on object detection.

2. Related Work

M-P model-based networks. These networks fix their structures during training, and several methods aim to address challenges arising from this constraint from different perspectives. We categorize representative methods below.

Parameter pruning contains non-structural pruning for disabling weak connections [15] and structural pruning for zeroing out groups of filters [46], which has a tedious fine-tuning process and easily leads to a decrease in accuracy.

Knowledge distillation shows that the student model learns little from some teacher models due to the model capacity gap between them [55].

Dynamic neural network (DyNN) includes sample-wise by adjusting architectures based on sample [34], spatial-wise by performing inference concerning different spatial locations of images [44], and temporal-wise by allocating less computation to the inputs at minor temporal locations [19]. The structure and parameters of DyNN varied by each sample or different spatial locations of images [18]. Instead, we adaptively optimize structure and parameters to fit data.

Neural architecture search (NAS) contains parameters and architecture optimization. NAS can be viewed as a method for searching connections from the M-P model-based search space [49]. The parameters optimization includes independent optimization [37] and sharing optimization [2]. The architecture optimization includes: 1) search space defines which architectures can be defined, including global and cell-based [56] search spaces; 2) search strategy includes: RL-based methods [57] use the performances of generated architectures as the rewards for training the controller. EA-based methods [37] search architectures with evolutionary algorithms. Gradient-based methods [9] regard network ar-

chitecture as a group of learnable parameters; 3) estimation strategy scores the searched architectures [49].

Biologically Plausible Neural Model. Based on the Tripartite Synapse [36], Astrocyte inserts itself into the nervous system to respond to neuron-released neurotransmitters, and then modulates them [4]. This shows that changes in neurotransmitters combine external disturbances and their state [26]. Hastings *et al*. focused on astrocytes communicating with distant neurons [21], and Han *et al*. explored modeling nervous systems with astrocytes [16, 17]. In contrast, we combine Olis with Asts to form Glia units to better mimic the optimization process of the neural pathway.

3. Methodology

This section first overviews the existing neuron model in Sec. 3.1, and introduces our Glia-Neuron model in Sec. 3.2, GliaNet in Sec. 3.3, and training scheme is in Sec. 3.4.

3.1. Preliminaries

M-P Model. The M-P model simulates simplified communication between neurons, restricting the connecting weights to unidirectional propagation. Its standard definition is

$$y = \phi_n\Big(\sum_i^N x_i w_i\Big) = \phi_n\Big(\mathbf{X}^\mathsf{T}\mathbf{W}\Big)\,, \tag{1}$$

where N is the number of pre-neurons, y denotes the output of the post-neuron, and $\phi_n()$ is the activation function of the neuron. Here, $\mathbf{X} = [x_1, \cdots, x_i \cdots, x_N]^\mathsf{T}$ be the data (or neuron state), $\mathbf{W} = [w_1, \cdots, w_i \cdots, w_N]^\mathsf{T}$ be the weight and w_i is a connecting weight from the i^{th} pre-neuron with state x_i to the post-neuron.

Artiffcial Astrocyte-Neuron Model. Ast establishes bidirectional communication with neurotransmitters (weights) released by pre-neurons to modulate the conveyed information. The simplified artificial Astrocyte-Neuron (A-N) model with only one Ast can be formulated as,

$$\mathbf{W}' = \mathcal{P}\Big(\phi_a\big(f(w_i^t)\big), \delta\Big) = \mathcal{P}\Big(\phi_a\big(f(\mathbf{W}^t)\big), \delta\Big)\,, \tag{2}$$

where $f()$ and $\phi_a()$ denote the modulation and activation function of Ast. Here, $\mathbf{W}'$ is the updated weight, $\mathbf{W}^t = [w_1^t, \cdots, w_i^t, \cdots, w_N^t]^\mathsf{T}$ and $w_i^t = f(w^{t-1})$ is a connecting weight concerning iterative modulation by Ast from $t-1$ to t, where $t \in [1, T]$ and T is the maximum iteration number.

However, the A-N model removes low-strength connections modulated by a single Ast through an empirical pruning function $\mathcal{P}()$ with a threshold δ. This constraint reduces model accuracy and limits the selection of optimal connection combinations.

3.2. Glia-Neuron Model

This section presents the main theoretical insights of this paper. We introduce Oli to select winning neurons adaptively, in contrast to the simplified A-N model in Eq. (2) that uses empirical pruning. We propose our Glia-Neuron (G-N) model, which considers Oli and Ast as integral components of the Glia unit. This model is constructed at the instance, group, and interaction levels.

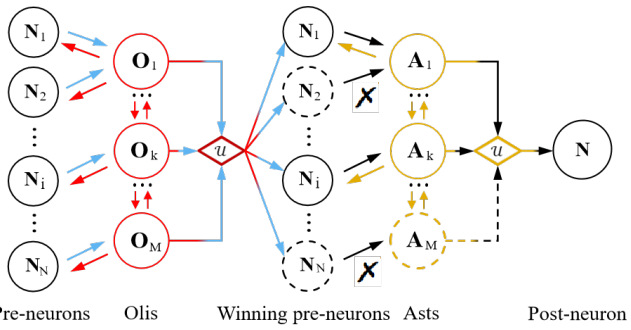


Figure 2. *The structure of our G-N model. M numbers of Olis and Asts are involved in jointly optimizing the communication between pre- and post-neurons.*

Instance-level: The Oli establish bidirectional communication with all pre-neurons. They are stimulated by the state of pre-neurons, and their output is used as feedback to select the winning neuron [28]. Inspired by [35], we build a selection function $g()$ of Oli. Specifically, given N pre-neurons $\{\mathbf{N}_i\}_{i=1}^N$ with state $\mathbf{X} \in \mathbb{R}^{N\times 1}$, where each neuron $\mathbf{N}_i$ with a state x_i. We search for winning neurons by,

$$\begin{aligned} \widetilde{\mathbf{X}} &= \mathcal{I}\big(g(\mathbf{X})\big) = \{g(x_i) \in g(\mathbf{X}) \mid g(x_i) \neq 0\}, \\ \mathbf{e} &= \phi_o\big(g(\mathbf{X})\big)\,, \end{aligned} \tag{3}$$

where $g()$ and $\phi_o()$ denote the selection and activation function of Oli. Here, $\mathcal{I}()$ denotes the operation of moving zero elements from the input vector, $\widetilde{\mathbf{X}} \in \mathbb{R}^{S\times 1}$ denote the state (that participates in subsequent propagation) of S winning neurons identified by Oli from $\mathbf{X}$, and $\mathbf{e} \in \mathbb{R}^{N\times 1}$ is a binary vector with value one on winning neurons with $\mathbf{1}_N^\mathsf{T}\mathbf{e} = S$, where $\mathbf{1}_N^\mathsf{T}$ is a 1-vector of size N.

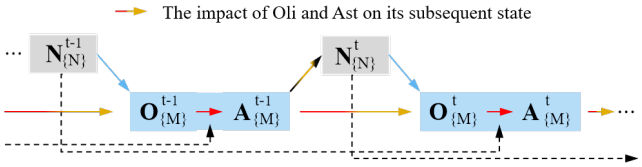
After selecting the winning neurons, the subsequent Ast modulate their connections as described by Eq. (2), excluding the pruning function $\mathcal{P}()$. We formulate the Glia unit as,

$$\mathbf{W}' = \phi_a\Big(f\big(\widetilde{\mathbf{W}}^t\big)\Big), \;\; \widetilde{\mathbf{W}}^t = \mathcal{I}\big(\mathbf{e} \odot \mathbf{W}^t\big)\,, \tag{4}$$

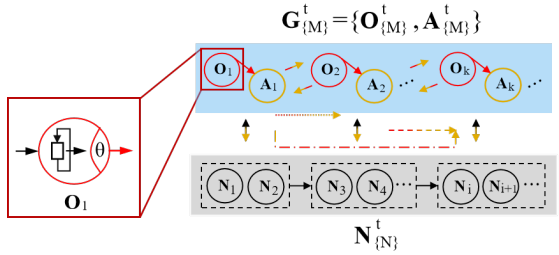
where $\odot$ is the element-wise product, $\mathbf{e}$ and $\mathcal{I}()$ select the weights $\widetilde{\mathbf{W}}$ associated with $\widetilde{\mathbf{X}}$ from $\mathbf{W}$, where $\widetilde{\mathbf{W}} \in \mathbb{R}^{S\times 1}$. In this way, Oli simplifies the network by empowering Ast to respond solely to specific weight components, thereby enhancing Ast's modulation capacity.

Group-level: Since Glia shares similar properties with Ast, we adopt the concepts from [17], and extend the model of the Glia unit into groups. Therefore, the model's complexity depends on the number of Glia, The Ablation study examines Glia numbers' impact on performance.

We conduct multiple, *e.g.*, M Glia units, where $\mathbf{G}_{\{M\}} = \{\mathbf{O}_{\{M\}}, \mathbf{A}_{\{M\}}\}$. Each Glia unit corresponding to a subset



(a) Interaction within a Glia unit



(b) Interaction between Glia units

Figure 3. *The interaction pipeline in Glia units. (a) Olis* $\mathbf{O}^t_{\{M\}}$ *and Asts* $\mathbf{A}^t_{\{M\}}$ *at iteration* t *is affected by history output of* $\mathbf{O}^{t-1}_{\{M\}}$ *and* $\mathbf{A}^{t-1}_{\{M\}}$ *from iteration* $t-1$. *(b) A Glia unit* $\{\mathbf{O}^t_k, \mathbf{A}^t_k\}$ *involves interactions with the rest* $M-1$ *Glia* $\{\mathbf{O}^t_{\{M-1\}}, \mathbf{A}^t_{\{M-1\}}\}$. *The red zoom in as an example of an Oli that varies temporally.*

from all N pre-neurons, *e.g.*, the k^{th} Oli, $\mathbf{O}_k$, establishes a bidirectional feedback connection with N_k pre-neuron states to select the winning neurons.

Then, the k^{th} Ast, $\mathbf{A}_k$, modulates the weights $\widetilde{\mathbf{W}}_k$ of these S_k winning neurons. Note, if the k^{th} Ast is not connected to any pre-neurons, it will be removed (Fig. 2). We rewrite Eq. (3) and Eq. (4) to define a group of Glia units,

$$\begin{aligned} \widetilde{\mathbf{X}} &= \mathcal{I}\Big(g_k(\mathbf{e}_k \odot \mathbf{X})\Big)_{k=1}^{M}, \\ \mathbf{W}' &= \phi_a\Big(f_k(\widetilde{\mathbf{e}}_k \odot \widetilde{\mathbf{W}}^t)\Big)_{k=1}^{M}, \end{aligned} \tag{5}$$

where $g_k()$ and $f_k()$ denote the selection and modulation functions of k^{th} Oli and Ast. We define $\mathbf{e}_k \in \mathbb{R}^{N\times 1}$ and $\widetilde{\mathbf{e}}_k \in \mathbb{R}^{N_k\times 1}$ to select the subset of all pre-neurons, where $\mathbf{e}_k$ and $\widetilde{\mathbf{e}}_k$ are binary vectors with value one for matching N_k pre-neuron states with the k^{th} Oli, and the weights $\widetilde{\mathbf{W}}_k$ of the associated S_k winning neurons with the k^{th} Ast. Note that $\mathbf{1}_N^\top \mathbf{e}_k = N_k$ and $\mathbf{1}_{N_k}^\top \widetilde{\mathbf{e}}_k = S_k$.

Interaction-level: The interaction of Glia units, $\mathbf{G}_{\{M\}}$, encompasses two aspects. i) The interaction within each Glia unit involves $\mathbf{O}_k$ collaborating with $\mathbf{A}_k$ to iteratively optimize the neural structure and connections. Moreover, $\mathbf{O}^t_k$ at the t^{th} iteration is influenced by the optimization results on the network of $\mathbf{O}^{t-1}_k$ and $\mathbf{A}^{t-1}_k$ from the $(t-1)^{th}$ iteration, which impacts the optimization of $\mathbf{O}^t_k$. This enables us to improve the previous functions $g_k()$ and $f_k()$ to $g^t_k()$ and $f^t_k()$ by incorporating temporal abilities (Fig. 3a). Note that the time step t in $\mathbf{W}^t$ denotes the iterative modulation of weights by Ast, while t in $g^t_k()$ and $f^t_k()$ denotes the t^{th} iteration of Oli and Ast.

ii) The interaction between Glia units, $\mathbf{G}_{\{M\}}$. Taking $\mathbf{G}_k = \{\mathbf{O}_k, \mathbf{A}_k\}$ as an example, $\mathbf{G}_k$ undergoes optimization by collaborating with the rest $M-1$ Glias, $\mathbf{G}_{\{M-1\}}$, at each iteration (Fig. 3b). We construct the interaction function $\mathcal{U}()$ to characterize the collaboration pattern in $\mathbf{G}_{\{M\}}$, which ensures $\mathbf{O}^t_k$ and $\mathbf{A}^t_k$ undergo continuous dynamic updates during optimization. We rewrite Eq. (5) as

$$\begin{aligned} \widetilde{\mathbf{X}} &= \mathcal{U}\Big(\mathcal{I}\big(g^t_k(\mathbf{e}_k \odot \mathbf{X})\big)\Big)_{k=1}^{M}, \\ \widetilde{\mathbf{X}}^\top \mathbf{W}' &= \mathcal{U}\Big(\widetilde{\mathbf{X}}^\top\Big(\phi_a\big(f^t_k(\widetilde{\mathbf{e}}_k \odot \widetilde{\mathbf{W}}^t)\big)\Big)\Big)_{k=1}^{M}, \end{aligned} \tag{6}$$

where $\mathcal{U}() = \{u_1(), u_2()\}$, $u_1()$ and $u_2()$ are applicable to the collaboration of $\mathbf{O}_{\{M\}}$ and $\mathbf{A}_{\{M\}}$.

Building on the concept of Glia [20], Oli receives stimulation from the pre-neuron states and provides neurotrophins as feedback to select the winning neurons. However, this becomes challenging in complex neural networks due to the potentially large number of neurons and connections. To address this, we introduce an integration function $\mathcal{H}()$ to enhance Oli's capacity to process large inputs. A similar approach is applied to Ast for integrating the connections from the winning neurons. Formally,

$$\begin{aligned} \widetilde{\mathbf{X}} &= \mathcal{U}\Big(\mathcal{I}\Big(g^t_k(\mathcal{H}(\mathbf{e}_k \odot \mathbf{X}))\Big)\Big)_{k=1}^{M}, \\ \widetilde{\mathbf{X}}^\top \mathbf{W}' &= \mathcal{U}\Big(\widetilde{\mathbf{X}}^\top\Big(\phi_a\Big(f^t_k(\mathcal{H}(\widetilde{\mathbf{e}}_k \odot \widetilde{\mathbf{W}}^t))\Big)\Big)\Big)_{k=1}^{M}, \end{aligned} \tag{7}$$

where $\mathcal{H}() = \{h_1(), h_2()\}$ denotes the integration function that enhances the processing ability of each $\mathbf{O}_k$ and $\mathbf{A}_k$ in $\mathbf{G}_{\{M\}}$. We name the Eq. (7) as our G-N model.

3.3. Framework of GliaNet

With our G-N model, we can construct the GliaNet to optimize network structure and connections adaptively. Given a set of input $\mathbf{X}$, we rearranging Eq. (1) to (7) and yields:

$$\mathbf{Y} = \mathcal{N}\Big(\mathcal{O}(\mathbf{X}, \Phi),\ \mathcal{A}(\widetilde{\mathbf{W}}, \Theta)\Big), \tag{8}$$

where $\mathcal{N}()$ aims to perform various downstream tasks, *i.e.*, $\mathcal{N}()$ is an M-P model-based neural module (NM) with connecting weights $\mathbf{W}$. We introduce the Glia module (GM) into the NM based on our G-N model. In GM, Oli identifies the winning neurons $\widetilde{\mathbf{X}}$ from $\mathbf{X}$ to participate in subsequent propagation via the function $\mathcal{O}()$, and their weights $\widetilde{\mathbf{W}}$ are modulated by Ast via function $\mathcal{A}()$, which defined in Eq. (2) (excluding the function $\mathcal{P}()$). The connecting weights of Oli $\Phi = \{\phi_\mathcal{H}, \phi_g, \phi_\mathcal{U}\}$ and Ast $\Theta = \{\theta_\mathcal{H}, \theta_f, \theta_\mathcal{U}\}$ are implied in $\{g^t_k()\}^{M,T}_{k,t=1}$, $\{f^t_k()\}^{M,T}_{k,t=1}$, $\mathcal{H}()$ and $\mathcal{U}()$.

Both $g^t_k()$ and $f^t_k()$ utilize a temporal-iterative approach to optimize the network structure, endowing them with the characteristics of processing time series data. Consequently, LSTM with a recursive structure is used to construct these functions, *i.e.*, $[\text{LSTM}^q]_{q=1}^{Q}$, where Q denotes

the number of layers within LSTM. The input of each LSTM layer integrates the output of all previous LSTM layers, *e.g.*, $g_k^t(x,q) = \sum_{q=1}^{q-1} f_k^t(x,q)$, to enhance each Oli response to its every optimization history.

The construction of the integration function $\mathcal{H}()$ must satisfy two requirements: 1) Compress the matrix composed of neuron states and weights in NM; 2) Ensure that the compressed matrix retains as much key information as possible. These requirements are implemented by $h_1()$ and $h_2()$ in $\mathcal{H}()$, respectively. $h_1()$ employs average pooling, compressing each Glia input into a lower-dimensional feature. For $h_2()$, the ViT module is utilized. Each Glia input features are treated as a patch, incorporating positional information to represent their associations across NM. Note that due to the compression operation, ViT can be designed as an extremely lightweight model.

The collaboration between $\mathbf{G}_{\{M\}}$ is established by interaction function $\mathcal{U}()$, and implemented via a chain optimization strategy. As depicted in Fig. 3b, in the k^{th} Glia unit $\mathbf{G}_k = \{\mathbf{O}_k, \mathbf{A}_k\}$, $\mathbf{O}_k$ is optimized to select winning neurons, followed by the optimization of $\mathbf{A}_k$ to modulate their connections. At each iteration, training proceeds from $\mathbf{G}_1$ to $\mathbf{G}_M$ until the optimization of $\mathbf{G}_{\{M\}}$ is completed. Therefore, the realization of $\mathcal{U}()$ involves the optimization of $\{g_k^t()\}_{k,t=1}^{M,T}$ and $\{f_k^t()\}_{k,t=1}^{M,T}$. The chain optimization strategy, by the interaction function $\mathcal{U}()$, establishes the temporal correlation between different neurons, enabling the optimized neuron to have a temporal impact on other neurons.

3.4. Training Scheme

Dynamic interaction exists between the neuron and Glia [20], offering an alternative optimization strategy for GliaNet. Since neurons tend to develop preferentially in the nervous system [12], we implement optimization on the NM in the training set first and then proceed to optimize the GM in the validation set. Note that we split the original training set ($\mathbf{X}$) into training ($\mathbf{X}^{tr}$) and validation ($\mathbf{X}^{val}$) sets, and $\mathbf{X}^{tr} \gg \mathbf{X}^{val}$. This brings two advantages: 1) $\mathbf{X}^{val}$ makes the optimized neural structure and connections have good predictive ability for unseen data via the optimization of Glia. 2) Small-scale $\mathbf{X}^{val}$ improves the training speed of GM. The optimal parameters $\mathbf{W}$ of NM are determined by minimizing the following function

$$\mathbf{W}^* = \arg\min_{\mathbf{W}} \mathcal{L}^{\mathrm{NM}}\Big(\mathcal{N}\big(\mathbf{X}^{tr}, \mathbf{W}\big), \mathbf{Y}_{\mathrm{gt}}\Big)\,, \tag{9}$$

where $\mathcal{L}^{\mathrm{NM}}()$ is the task specific loss function and $\mathbf{Y}_{\mathrm{gt}}$ is the ground-truth label.

With the initialized $\mathbf{W}^*$, we then optimize the GM with a data term $\mathcal{L}^{\mathrm{data}}()$ and regularization term $\mathcal{L}^{\mathrm{reg}}()$, where $\widetilde{\mathbf{W}}^*$ is from $\mathcal{O}_k()$. The data term forces k^{th} Glia, $\mathbf{G}_k$, to optimize NM for high performance with its supervision as,

$$\mathcal{L}_k^{\mathrm{data}}(\Phi,\Theta) = \mathcal{L}_k^{\mathrm{NM}}\Big(\mathcal{N}\big(\mathcal{O}_k(\mathbf{X}^{val},\Phi),\ \mathcal{A}_k(\widetilde{\mathbf{W}}^*,\Theta)\big), \mathbf{Y}_{\mathrm{gt}}\Big)\,, \tag{10}$$

Algorithm 1: The GliaNet Optimization Algorithm

Input: NM parameters: $\mathbf{W}$, GM parameters: Φ and Θ, training set $\mathbf{X}^{tr}$, validation set $\mathbf{X}^{val}$, ground-truth label: $\mathbf{Y}_{\mathrm{gt}}$, max iterations: R.
Output: GliaNet output: $\mathbf{Y}$

```
for r = 1 to R:
    if r%2 == 0:
        Optimize L^NM(X^tr, Y, Y_gt) → Update W* via Eq. (9)
    else:
        for k = 1 to M:
            Optimize L_k^GM(X^val, Y, Y_gt) → Update Φ_k* and Θ_k* via Eq. (12)
        end for
        Φ* and Θ*
end for
```

The regularization term, inspired by [54], encourages each $\mathbf{O}_k$ and $\mathbf{A}_k$ to learn similar representations for different features from GM, *e.g.*, z_i and z_{N+i}, from the same neuron state from the NM, $\mathbf{X}$ (see supplementary material details).

$$\mathcal{L}_k^{\mathrm{reg}}(\Phi,\Theta) = -\sum_{i=1}^{N} \log \frac{\exp(z_i \cdot z_{i+N})/\tau}{\sum_{p=1}^{2N} \mathbb{1}_{[p\neq i]} \exp(z_i \cdot z_p)/\tau}\,, \tag{11}$$

where $\mathbb{1} \in \{0,1\}$ is an indicator function evaluating to 1 if $p \neq i$ and τ is a temperature. Taking Eq. (10) and (11), the final learning objective of $\mathbf{G}_k$ is formulated as

$$\begin{aligned} \Phi_k^*, \Theta_k^* &= \arg\min_{\Phi_k,\Theta_k} \big(\mathcal{L}_k^{\mathrm{data}}(\Phi,\Theta) + \lambda \mathcal{L}_k^{\mathrm{reg}}(\Phi,\Theta)\big) \\ &= \arg\min_{\Phi_k,\Theta_k} \big(\mathcal{L}_k^{\mathrm{GM}}(\Phi,\Theta)\big)\,. \end{aligned} \tag{12}$$

where λ is to balance the two items. Compared with AstroNet [16] employing carefully tuned initialization, our alternative optimization strategy (see Algorithm 1) demonstrates superior effectiveness across multiple public datasets without additional manual pruning.

4. Experiments

Implementation Details. Our GliaNet is implemented in PyTorch and trained via SGD optimizer with a learning rate of $1e$-1. We set the number of Glia units as $M = 4$, *i.e.*, $\mathbf{G}_{\{4\}} = \{\mathbf{O}_{\{4\}}, \mathbf{A}_{\{4\}}\}$, each Oli and Ast contains $Q = 3$ LSTMs inside, $\lambda = 1e$-1 in Eq. (12). Please refer to the Supplementary Material for more details. All report figures are the average of six times repeated experiments. Our code and models will be made available to facilitate reproducible research.

Table 1. *Experiments on classification with different methods on CIFAR10. With the same settings, compare networks constructed by our G-N model and other models. Our (RNT) and Our (WRN) denote that the NM in GliaNet is set to RNT or WRN. We highlight the best and the second-best numbers in* **bold** *and* **bold**.

Architecture	Acc (%)	Params (M)	Inference speed (ms)
RNT18 [22]	94.96	11.17	3
CDS [54]	96.49	11.17	3
AstroNet [16]	96.52	7.31	**2**
MA-Net [17]	**96.81**	**6.94**	**1**
Ours (RNT18)	**96.90**	**6.95**	**1**
RNT50 [22]	95.07	25.60	5
CDS [54]	96.78	25.60	6
AstroNet [16]	96.76	7.64	**3**
MA-Net [17]	**97.10**	**7.36**	**3**
Ours (RNT50)	**97.26**	**7.34**	**2**
WRN50 [53]	95.01	68.90	10
CDS [54]	96.88	68.90	13
AstroNet [16]	96.90	42.55	8
MA-Net [17]	**97.25**	**38.89**	**7**
Ours (WRN50)	**97.42**	**38.26**	**6**

Image Classification. We validate our GliaNet performance on CIFAR10, CIFAR100 [27] and ImageNet-1k [8]), with commonly used convolutional neural networks as the NM including ResNet (RNT) [22], Wide ResNet (WRN) [53], DenseNet-BC (DNT-BC) [24], and vision transformers including ViT [10] and Swin [31]. We further compare our GliaNet with State-of-the-Art (SOTA) NAS methods [45, 47], DyNN method [52], A-N model-based AstroNet [16] and MA-Net [17], M-P model-based CDS [54] with contrastive learning to verify our adaptive ability.

Segmentation. Our method is evaluated on the COCO [30]. The baselines include box-supervised methods [43] and fully-supervised methods [42].

Object Detection. Our method is evaluated on COCO with Faster RCNN [38] and Dynamic DETR [7]. Note baselines in all our experiments are re-trained following [16].

4.1. Experimental Results

Image Classification. The experimental results of our GliaNet on CIFAR10 and ImageNet-1k are shown in Tab. 1 and Tab. 2. For CIFAR10, our GliaNet achieves SOTA accuracy on RNT18, RNT50, and WRN50 and outperforms CDS, AstroNet and MA-Net by 0.40%, 0.39% and 0.14% in accuracy on average. Notably, our method reduces the capacity of RNT18, RNT50 and WRN50 by 37.33%, 71.33% and 44.47%, respectively. For ImageNet-1k, on average, our GliaNet outperforms the CDS, AstroNet and MA-Net by 0.62%, 0.60% and 0.16% in accuracy, respectively. Our method reduces the capacity of networks by 35.87% and 68.79%, respectively. By optimizing the structure and connections of the same neural module, GliaNet can adaptively find a larger network structure to fit the distribution of complex datasets.

Table 2. *Experiments on classification with different methods on ImageNet-1k. With the same settings, compare networks constructed by our G-N model and other models.*

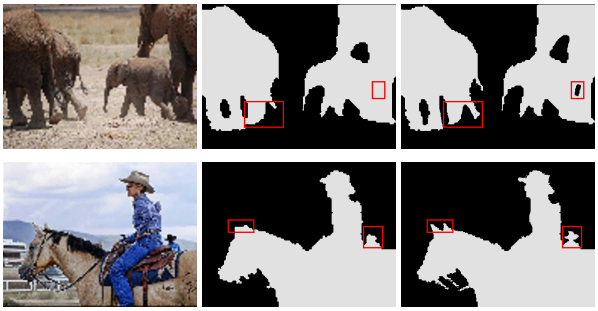
Architecture	Acc (%)	Params (M)	Inference speed (ms)
RNT18 [22]	69.21	11.68	6
CDS [54]	72.85	11.68	7
AstroNet [16]	72.82	8.05	**4**
MA-Net [17]	**73.27**	**7.52**	**3**
Ours (RNT18)	**73.39**	**7.49**	**3**
RNT50 [22]	75.30	26.11	15
CDS [54]	78.25	26.11	18
AstroNet [16]	78.31	8.70	**9**
MA-Net [17]	**78.96**	**8.16**	**9**
Ours (RNT50)	**79.15**	**8.15**	**7**

Table 3. *Experiments on segmentation with various backbones on COCO. RNT50, RNT50-CDS, RNT50-Ast, RNT50-MA and RNT50† pre-trained on ImageNet-1k by standard training, CDS, AstroNet, MA-Net and Our methods.*

Model	backbone	AP	AP_S	AP_M	AP_L
BoxInst [43]	RNT50	32.1	15.6	34.3	43.5
CDS [54]	RNT50-CDS	32.5	15.9	34.6	43.2
AstroNet [16]	RNT50-Ast	32.7	15.7	34.1	43.6
MA-Net [17]	RNT50-MA	**34.7**	**16.4**	**34.8**	**44.6**
Ours	RNT50†	**35.7**	**16.8**	**34.6**	**45.2**
CondIns [42]	RNT50	39.1	21.5	41.7	50.9
CDS [54]	RNT50-CDS	39.8	21.6	42.2	51.1
AstroNet [16]	RNT50-Ast	40.0	21.8	41.8	50.8
MA-Net [17]	RNT50-MA	**40.7**	**22.7**	**42.8**	**51.7**
Ours	RNT50†	**41.5**	**22.9**	**42.8**	**52.0**

Segmentation. We compare with BoxInst and CondInst methods on the COCO dataset. Note that the backbone, *i.e.*, RNT50, of the standard segmentation method is pre-trained on ImageNet-1k. Keeping the same as other settings in BoxInst and CondInst, we only replace RNT50 with RNT50-CDS, RNT50-Ast, RNT50-MA and RNT50†, which are pre-trained using different methods, including CDS, AstroNet, MA-Net and our GliaNet.

Tab. 3 reports the comparison of our method against baselines on the COCO dataset. The pre-trained G-N model based-backbone (RNT50†) with our method, compared to the second-best method (MA-Net), achieves AP improvements of 1.0% and 0.8%, respectively. Fig. 4 shows our qualitative segmentation comparison against previous backbones, affirming our advantages. We predict a sharper and



(a) Input (b) AstroNet (c) Ours

Figure 4. *We display the segmentation results on images (a) using different backbones, which are obtained by AstroNet (b) and Our (c). Our segmentation boundary is more precise compared to baselines,* e.g., *areas in the red rectangle.*

Table 4. *Experiments on object detection with various backbones on COCO. RNT*50*, RNT*50*-CDS, RNT*50*-Ast, RNT*50*-MA and RNT*50† *pre-trained on ImageNet-1k by standard training, CDS, AstroNet, MA-Net and Our methods.*

Model	backbone	AP	AP_S	AP_M	AP_L
Faster RCNN [38]	RNT50	37.4	21.2	41.0	48.1
CDS [54]	RNT50-CDS	38.3	21.6	42.0	50.1
AstroNet [16]	RNT50-Ast	38.0	21.3	42.2	50.3
MA-Net [17]	RNT50-MA	**39.8**	**21.9**	**42.5**	**51.6**
Ours	RNT50†	**40.5**	**22.1**	**42.9**	**52.0**
Dynamic DETR [7]	RNT50	42.9	24.6	44.9	54.4
CDS [54]	RNT50-CDS	43.2	25.1	45.4	54.8
AstroNet [16]	RNT50-Ast	43.5	25.0	45.5	50.0
MA-Net [17]	RNT50-MA	**44.0**	**25.8**	**45.8**	**55.2**
Ours	RNT50†	**45.1**	**26.0**	**46.0**	**54.8**

more complete contour compared to the AstroNet method, especially on elephant legs and horse ears.

Object Detection. Tab. 4 shows our object detection performance on the COCO dataset. We employ the same method for acquiring backbones as in the segmentation task and then fine-tune them with identical settings for the object detection task. Experimental results indicate that utilizing the pre-trained backbone (RNT50†) from our method results in AP improvements of 0.7% and 1.1% compared to the second-best method, *i.e.*, MA-Net (RNT50-MA) in Faster RCNN and Dynamic DETR, respectively.

4.2. Discusssions

Comparison with Vision Transformer. We employ the visual transformer as the NM in the GliaNet for evaluating the effectiveness of our method on ImageNet-1k. The results are shown in Tab. 5. Compared with ViT and Swin, by following their settings (*e.g.*, the input size is 384^2) for training, our method achieves a notable improvement in accuracy by 1.62% and 1.20%. It also reduces the capacity of

Table 5. *Results of our method on the visual transformer on ImageNet-1k. Our (ViT-B) and Our (Swin-B) denote that the NM in GliaNet is set to ViT-B and Swin-B, respectively. B denotes the standard model of ViT and Swin.*

Architecture	Acc (%)	Params (M)	Inference speed (ms)
ViT-B [10]	**77.90**	**86.00**	**25**
Ours (ViT-B)	**79.52**	**70.25**	**20**
Swin-B [31]	**84.50**	**88.00**	**37**
Ours (Swin-B)	**85.70**	**78.38**	**33**

Table 6. *Evaluation of GliaNet in transfer learning. GliaNet and ViT are pre-trained on ImageNet-1k and then fine-tuned on CIFAR100. L denotes the large model of ViT.*

Architecture	Acc (%)	Params (M)
ViT-B [10]	**87.13**	**86.00**
Ours (ViT-B)	**88.05**	**72.40**
ViT-L [10]	**86.35**	**307.00**
Ours (ViT-L)	**87.20**	**252.70**

Table 7. *Compared our method with the SOTA DyNN method on ImageNet-1k. Take RNT*50*, whose performance is close to that of the DyNN method as a reference.*

Architecture	Acc (%)	Params (M)	Inference speed (ms)
RNT50 [22]	75.30	25.60	15
Boost-DyNN [52]	**76.05**	**11.38**	**11**
Ours (RNT50)	**79.15**	**8.15**	**7**

ViT and Swin by 18.31% and 10.93%. For the popular visual transformer, our method shows the potential to alleviate the high parameter problem of transformers and provides a new perspective for obtaining efficient transformers.

Transfer Learning Ability. We adopt ViT as our experimental benchmark. Both GliaNet (with ViT as the NM) and ViT undergo pre-training on ImageNet-1K for 40 epochs, followed by fine-tuning on CIFAR100. Tab. 6 illustrates that transfer learning, especially utilizing the pre-trained GliaNet, yields superior performance, highlighting the remarkable transferability of our method.

Comparison with DyNN Method. We evaluate the performance of our connection-based adaptive modulation method with the sample (feature)-based adaptive inference DyNN method. Tab. 7 reveals that, in comparison with the SOTA Boost-DyNN (76.05%), our GliaNet (78.96%) maintains competitive performance with lower parameters.

Comparison with NAS Methods. We report the performance results of our GliaNet and SOTA NAS methods on CIFAR10 in Tab. 8. Compared with NAS-LID and HOTNAS, our method improves the accuracy by 0.30%

Table 8. *Comparison with SOTA NAS methods. GliaNet and NAS architectures are searched and evaluated on CIFAR10. Time cost only denotes the search time by our method and NAS methods. Our (DNT-BC) denotes that the TNet in our GliaNet is set to DNT-BC.*

Architecture	Acc (%)	Params (M)	Time Cost (GPU days)
DNT-BC [24]	96.54	25.60	-
NAS-LID [23]	97.52	**6.90**	**2.30**
HOTNAS [51]	**97.54**	**4.70**	3.40
Ours (DNT-BC)	**97.82**	7.49	**0.70**

Table 9. *Comparison with SOTA NAS methods for trasferability. GliaNet and NAS architectures are searched on CIFAR10 and then evaluated on ImageNet-1k.*

Architecture	Acc (%)	Params (M)	Time Cost (GPU days)
RNT50 [22]	75.30	26.11	-
NAS-LID [23]	77.10	6.90	**2.30**
HOTNAS [51]	**77.30**	**4.70**	3.40
Ours (RNT50)	**78.35**	**6.89**	**0.66**

and 0.28% with fewer time costs. The comparison results on ImageNet-1k are demonstrated in Tab. 9. We trained the best-found architectures (both our GliaNet and NAS) on CIFAR10 to evaluate their transferability on ImageNet-1k. Compared to the second-best HOTNAS method, our method leads to an improvement in accuracy by 1.05%, which verifies the transferability of our method.

4.3. Ablation Studies

This section discusses the rationality of our method, where the NM in GliaNet is set to RNT18 in all experiments and evaluated on CIFAR10.

The Number of Glia units. We investigate the GliaNet performance concerning the numbers of Glias in the GliaNet. Tab. 10 shows a positive correlation between the number and performance of Glias, *i.e.*, increasing Glias numbers can achieve higher accuracy. However, increasing the number of Glia units significantly increases the training time of GliaNet. Therefore, considering the time cost, we set the number of Glias to 4, *i.e.*, $\mathbf{G}_{\{4\}} = \{\mathbf{O}_{\{4\}}, \mathbf{A}_{\{4\}}\}$.

Ablation of GliaNet. Compared to the A-N model-based AstroNet and MA-Net, we select winning neurons by transforming from static connection removal to dynamic neuron search, adaptively searching for neurons rather than empirically pruning connections.

We set up progressive ablation experiments with four cases to verify the effectiveness of our method. Case 1: Study the existence of a single Glia unit in GliaNet to verify the rationale behind the collaboration between Oli and Ast in optimizing the neural structure and connections. Case 2: Study the existence of multiple Olis in GliaNet without introducing Ast to verify the necessity of Oli's adaptive selection of winning neurons. Case 3: Study the existence of multiple Ast in GliaNet without introducing Oli to verify the necessity of Ast's adaptive modulation of connections, this is consistent with the approach used in MA-Net. Case 4: Combining Case 2 and Case 3 to validate the effectiveness of our method. Tab. 11 shows the performance of four cases. Notably, the accuracy of all cases exceeds the RNT18 baseline, and our method achieves the best performance. This demonstrates that the G-N model is more generalizable than both the M-P and A-N models.

Table 10. *Comparing the accuracy of GliaNet with respect to different numbers of Glia units on CIFAR10. The "Params" in the table denotes the number of parameters in the Glia units.*

Glias Numbers	1	2	4	6	8
Acc (%)	96.68	96.77	96.86	**96.87**	**96.90**
Params (M)	**0.84**	**1.68**	3.36	5.04	6.72
Time Cost (GPU hours)	**1.16**	**2.44**	4.35	6.10	7.72

Table 11. *Experiments on image classification methods with different model construction manners on CIFAR10.*

Cases	Model Construction Manners	Acc (%)	Params (M)
RNT18	-	94.96	11.17
Case 1	A single Glia unit	96.68	7.04
Case 2	Multiple Olis	96.72	**6.95**
Case 3	Multiple Asts	**96.81**	**6.94**
Case 4 (Ours)	Multiple Glia units	**96.90**	**6.95**

More. Please refer to our supplementary materials for more details, *e.g.*, source code, training, hyper-parameter, and more experimental comparisons and visualization results.

4.4. Conclusions

In this paper, we propose the Glia-Neuron (G-N) model, which consists of Oli and Ast as integral components of Glia. We study the structural properties and interactions among Glias to develop the GliaNet. The structural properties of Glia enable it to adaptively select winning neurons suitable for network decision-making and modulate their connections. Interactions enhance the Glia's capacity to optimize neural communication. This allows our GliaNet to adaptively optimize its structure and connections during training. Experiments demonstrate that our GliaNet achieves SOTA accuracy and significantly reduces the network's parameters with adaptive optimization.

Acknowledgment. This work was supported by the National Natural Science Foundation of China (82171965, 62302045), and the Northwest A&F University Scientific Research Startup Foundation.

References

[1] Ben A Barres and Yves-Alain Barde. Neuronal and glial cell biology. *Current opinion in neurobiology*, 10(5):642–648, 2000. 2

[2] Gabriel Bender, Pieter-Jan Kindermans, Barret Zoph, Vijay Vasudevan, and Quoc Le. Understanding and simplifying one-shot architecture search. In *International conference on machine learning*, pages 550–559. PMLR, 2018. 2

[3] Simone Bombari, Mohammad Hossein Amani, and Marco Mondelli. Memorization and optimization in deep neural networks with minimum over-parameterization. *Advances in Neural Information Processing Systems*, 2022. 1

[4] Gilles Bonvento and Juan P Bolaños. Astrocyte-neuron metabolic cooperation shapes brain activity. *Cell metabolism*, 33(8):1546–1564, 2021. 3

[5] Yuan Cao, Zixiang Chen, Misha Belkin, and Quanquan Gu. Benign overfitting in two-layer convolutional neural networks. *Advances in neural information processing systems*, 2022. 1

[6] Yucheol Cho, Gyeongdo Ham, Jae-Hyeok Lee, and Daeshik Kim. Ambiguity-aware robust teacher (art): Enhanced self-knowledge distillation framework with pruned teacher network. *Pattern Recognition*, page 109541, 2023. 1

[7] Xiyang Dai, Yinpeng Chen, Jianwei Yang, Pengchuan Zhang, Lu Yuan, and Lei Zhang. Dynamic detr: End-to-end object detection with dynamic attention. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021. 6, 7

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 2009. 6

[9] Xuanyi Dong and Yi Yang. Searching for a robust neural architecture in four gpu hours. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019. 2

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. 6, 7

[11] Gongfan Fang, Xinyin Ma, Mingli Song, Mi, and Michael Bi. Depgraph: Towards any structural pruning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 1

[12] R Douglas Fields and Beth Stevens-Graham. New insights into neuron-glia communication. *Science*, 298(5593):556–562, 2002. 5

[13] Giuliana Fossati, Michela Matteoli, and Elisabetta Menna. Astrocytic factors controlling synaptogenesis: a team play. *Cells*, 9(10):2173, 2020. 2

[14] Antoinette Y Foster, Helena Bujalka, and Ben Emery. Axoglial interactions in myelin plasticity: evaluating the relationship between neuronal activity and oligodendrocyte dynamics. *Glia*, 67(11):2038–2049, 2019. 2

[15] Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations*, 2019. 1, 2

[16] Mengqiao Han, Liyuan Pan, and Xiabi Liu. Astronet: When astrocyte meets artificial neural network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 1, 3, 5, 6, 7

[17] Mengqiao Han, Liyuan Pan, and Xiabi Liu. Ma-net: Rethinking neural unit in the light of astrocytes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 2040–2048, 2024. 1, 3, 6, 7

[18] Yizeng Han, Gao Huang, Shiji Song, Le Yang, and Honghui Wang. Dynamic neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11): 7436–7456, 2021. 2

[19] Christian Hansen, Casper Hansen, Stephen Alstrup, Simonsen, and Jakob Grue. Neural speed reading with structural-jump-lstm. In *International Conference on Learning Representations*, 2019. 2

[20] Daniel K Hartline. The evolutionary origins of glia. *Glia*, 59 (9):1215–1236, 2011. 4, 5

[21] Nataly Hastings, Yi-Lin Yu, Botian Huang, Sagnik Middya, Misaki Inaoka, Nadia A Erkamp, Roger J Mason, Alejandro Carnicer-Lombarte, Saifur Rahman, Tuomas PJ Knowles, et al. Electrophysiological in vitro study of long-range signal transmission by astrocytic networks. *Advanced Science*, page 2301756, 2023. 2, 3

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016. 6, 7, 8

[23] Xin He, Jiangchao Yao, Yuxin Wang, Zhenheng Tang, Ka Chun Cheung, Simon See, Bo Han, and Xiaowen Chu. Nas-lid: Efficient neural architecture search with local intrinsic dimension. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023. 8

[24] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017. 6, 8

[25] AG Ivakhnenko, GA Ivakhnenko, and JA Muller. Self-organization of neural networks with active neurons. *Pattern Recognition and Image Analysis*, 4(2):185–196, 1994. 2

[26] Paulo Kofuji and Alfonso Araque. G-protein-coupled receptors in astrocyte-neuron communication. *Neuroscience*, 456: 71–84, 2021. 3

[27] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 6

[28] Sarah Kuhn, Laura Gritti, and Daniel Crooks. Oligodendrocytes in development, myelin generation and beyond. *Cells*, 8(11):1424, 2019. 3

[29] Yawei Li, Kamil Adamczewski, Wen Li, Shuhang Gu, Radu Timofte, and Luc Van Gool. Revisiting random channel pruning for neural network compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 2

[30] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence

Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, 2014. 6

[31] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2021. 6, 7

[32] Patrick Long and Gabriel Corfas. To learn is to myelinate. *Science*, 346(6207):298–299, 2014. 2

[33] Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943. 1

[34] Ravi Teja Mullapudi, William R Mark, Noam Shazeer, and Kayvon Fatahalian. Hydranets: Specialized dynamic architectures for efficient inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 2

[35] Erik Nutma, Démi van Gent, Sandra Amor, and Laura AN Peferoen. Astrocyte and oligodendrocyte cross-talk in the central nervous system. *Cells*, 9(3):600, 2020. 2, 3

[36] Gertrudis Perea, Marta Navarrete, and Alfonso Araque. Tripartite synapses: astrocytes process and control synaptic information. *Trends in neurosciences*, 32(8):421–431, 2009. 3

[37] Esteban Real, Alok Aggarwal, Yanping Huang, and Quoc V Le. Regularized evolution for image classifier architecture search. In *Proceedings of the aaai conference on artificial intelligence*, 2019. 2

[38] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 2015. 6, 7

[39] Niklas Risse. Detecting overfitting of machine learning techniques for automatic vulnerability detection. In *Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pages 2189–2191, 2023. 1

[40] Maying Shen, Pavlo Molchanov, and Hongxu Yin. When to prune? a policy towards early structural pruning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 2

[41] Zhimin Tang, Linkai Luo, Bike Xie, Yiyu Zhu, Rujie Zhao, Bi, and Lvqing. Automatic sparse connectivity learning for neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 2022. 1

[42] Zhi Tian, Chunhua Shen, and Hao Chen. Conditional convolutions for instance segmentation. In *European Conference on Computer Vision*, 2020. 6

[43] Zhi Tian, Chunhua Shen, Xinlong Wang, and Hao Chen. Boxinst: High-performance instance segmentation with box annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 6

[44] Huiyu Wang, Aniruddha Kembhavi, Ali Farhadi, Alan L Yuille, and Mohammad Rastegari. Elastic: Improving cnns with dynamic scaling policies. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019. 2

[45] Ruochen Wang, Minhao Cheng, Xiangning Chen, Xiaocheng Tang, and Cho-Jui Hsieh. Rethinking architecture selection in differentiable nas. In *International Conference on Learning Representation*, 2021. 6

[46] Wei Wen, Chunpeng Wu, Yandan Wang, Yiran Chen, and Hai Li. Learning structured sparsity in deep neural networks. *Advances in neural information processing systems*, 2016. 2

[47] Han Xiao, Ziwei Wang, Zheng Zhu, Jie Zhou, and Jiwen Lu. Shapley-nas: discovering operation contribution for neural architecture search. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022. 6

[48] Wendy Xin and Jonah R Chan. Myelin plasticity: sculpting circuits in learning and memory. *Nature Reviews Neuroscience*, 21(12):682–694, 2020. 2

[49] Yixing Xu, Yunhe Wang, Kai Han, Yehui Tang, Shangling Jui, Chunjing Xu, and Chang Xu. Renas: Relativistic evaluation of neural architecture search. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 2, 3

[50] Guangyu Robert Yang and Xiao-Jing Wang. Artificial neural networks for neuroscientists: a primer. *Neuron*, 107(6): 1048–1070, 2020. 2

[51] Jiechao Yang and Yong Liu. Hotnas: Hierarchical optimal transport for neural architecture search. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 1, 8

[52] Haichao Yu, Haoxiang Li, Gang Hua, Gao Huang, and Humphrey Shi. Boosted dynamic neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023. 1, 6, 7

[53] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *British Machine Vision Conference 2016*. British Machine Vision Association, 2016. 6

[54] Linfeng Zhang, Xin Chen, Junbo Zhang, and Runpei Dong. Contrastive deep supervision. In *European Conference on Computer Vision*, 2022. 5, 6, 7

[55] Borui Zhao, Quan Cui, Renjie Song, Yiyu Qiu, and Jiajun Liang. Decoupled knowledge distillation. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, 2022. 2

[56] Zhao Zhong, Zichen Yang, Boyang Deng, Junjie Yan, Wei Wu, Jing Shao, and Cheng-Lin Liu. Blockqnn: Efficient block-wise neural network architecture generation. *IEEE transactions on pattern analysis and machine intelligence*, 43(7):2314–2328, 2020. 2

[57] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018. 2