



City Research Online

City St George's, University of London

Citation: Nimmo, J. J. & Mondragón, E. (2025). Advancing the Biological Plausibility and Efficacy of Hebbian Convolutional Neural Networks. *Neural Networks*, 190, 107628. doi: 10.1016/j.neunet.2025.107628

This is the accepted version of the paper.

This version of the publication may differ from the final published version. To cite this item please consult the publisher's version.

Permanent repository link: <https://city-ref-test.eprints-hosting.org/id/eprint/35360/>

Link to published version: <https://doi.org/10.1016/j.neunet.2025.107628>

Copyright and Reuse: Copyright and Moral Rights remain with the author(s) and/or copyright holders. Copies of full items can be used for personal research or study, educational, or not-for-profit purposes without prior permission or charge, unless otherwise indicated, provided that the authors, title and full bibliographic details are credited, a hyperlink and/or URL is given for the original metadata page and the content is not changed in any way. For full details of reuse please refer to [City Research Online policy](#).

Revision: Advancing the Biological Plausibility and Efficacy of Hebbian Convolutional Neural Networks

Julian Jiménez Nimmo^{a,1}, Esther Mondragón^{a,*}

^a*Artificial Intelligence Research Centre (CitAI), Department of Computer Science, City St George's, University of London, Northampton Square, EC1V 0HBC, London, United Kingdom*

Abstract

The research presented in this paper advances the integration of Hebbian learning into Convolutional Neural Networks (CNNs) for image processing, systematically exploring different architectures to build an optimal configuration, adhering to biological tenability. Hebbian learning operates on local unsupervised neural information to form feature representations, providing an alternative to the popular but arguably biologically implausible and computationally intensive backpropagation learning algorithm. The suggested optimal architecture significantly enhances recent research aimed at integrating Hebbian learning with competition mechanisms and CNNs, expanding their representational capabilities by incorporating hard Winner-Takes-All (WTA) competition, Gaussian lateral inhibition mechanisms and Bienenstock–Cooper–Munro (BCM) learning rule in a single model. Mean accuracy classification measures during the last half of test epochs on CIFAR-10 revealed that the resulting optimal model matched its end-to-end backpropagation variant with 75.2% each, critically surpassing the state-of-the-art hard-WTA performance in CNNs of the same network depth (64.6%) by 10.6%. It also achieved competitive performance on MNIST (98%) and STL-10 (69.5%). Moreover, results showed clear indications of sparse hierarchical learning through increasingly complex and abstract receptive fields. In summary, our implementation enhances both the performance and the generalisability of the learnt representations and constitutes a crucial step towards more biologically realistic artificial neural networks

Keywords: Hebbian Learning, Unsupervised Feature Learning, Convolutional Neural Networks (CNNs), Sparse Neural Coding, Hard/Soft Competition

1. Introduction

Deep neural networks (DNNs) have become the dominant paradigm in artificial intelligence, achieving remarkable performance across computer vision, natural language processing, and reinforcement learning. However, their primary learning algorithm - error backpropagation (Rumelhart et al., 1986; Werbos, 1994) - presents significant challenges for practical deployment and biological

*Corresponding author

Email address: e.mondragon@citystgeorges.ac.uk (Esther Mondragón)

¹First author.

13 plausibility (Zarkeshian et al., 2022; Lillicrap et al., 2020). Backpropagation is a
14 supervised learning process that relies on propagating error gradients backward
15 through the network’s layers to adjust connection weights so that the network
16 outputs come close to some given target output. This global update process
17 requires storing all intermediate activations and computing gradients across the
18 entire network (Rumelhart et al., 1986).

19 These constraints impose several critical limitations: the algorithm’s require-
20 ment for precise error signals demands massive labelled datasets (Lagani, 2022),
21 while the global nature of weight updates leads to extreme computational costs
22 and energy consumption (Wang et al., 2024; Strubell et al., 2020). Additionally,
23 the sequential backward pass creates bottlenecks resulting in lengthy training
24 times (Krithivasan et al., 2022), and the rigid supervised learning approach
25 limits adaptability while leaving models vulnerable to adversarial attacks (Lee
26 et al., 2018).

27 Conversely, Hebbian learning, a biologically inspired learning algorithm, cir-
28 cumvents the need for error backpropagation, updating weights locally, using un-
29 supervised local neural activity correlations. Although Hebbian learning shows
30 promise in addressing these challenges, posing as an alternative that mirrors the
31 efficiency and adaptability of biological systems (Gerstner and Kistler, 2002),
32 levels of accuracy as those reported with backpropagation are still out of reach
33 (Demidovskij et al., 2023).

34 Recent research in Hebbian learning has focused on combining purely Heb-
35 bian learning for unsupervised feature extraction in convolutional layers, with
36 supervised classification training on these frozen features. Successful Hebbian
37 integration into DNNs (Amato et al., 2019; Krotov and Hopfield, 2019; Miconi,
38 2021; Journé et al., 2022) is primarily achieved through neural competition,
39 which allows distinct neurons to respond to input patterns, preventing redun-
40 dant feature learning and promoting the development of diverse hierarchical
41 representations. Two representation coding schemes have been used: soft com-
42 petition methods (Journé et al., 2022) that enable distributed updates across all
43 neurons, as well as several hard competition approaches (Grinberg et al., 2019;
44 Amato et al., 2019; Miconi, 2017) where neurons undergo sparse updates that
45 limit the number of participating neurons.

46 Whereas soft-Winner-Takes-All (WTA) competition through Journé et al.
47 (2022)’s SoftHebb model achieves an accuracy on CIFAR-10 of nearly 80%, cur-
48 rent hard-WTA Hebbian approaches face challenges to match backpropagation
49 SOTA performance while maintaining biological plausibility and efficiency. The
50 SOTA accuracy with a Hard competition WTA model (Hard-WTA) on CIFAR-
51 10 is 72.2% (Grinberg et al., 2019). Although this architecture utilised just one
52 convolutional layer, it does not facilitate the construction of hierarchical repre-
53 sentations of the input, which are crucial for developing intricate representations
54 and understanding the relationships within the data. Multilayer networks are
55 necessary to achieve this. A sparse 3-CNN layer Hebbian architecture led to only
56 64.6% accuracy (Miconi, 2021), and struggled to form meaningful hierarchical
57 representations.

58 Arguably, sparse neural coding derived from Hard-WTA has greater biolog-

ical plausibility (Barth and Poulet, 2012) paralleling the highly sparse brain connectivity with less than 1% neurons firing in the visual cortex when representing an image (Yoshida and Ohki, 2020). This sparseness facilitates the energy efficiency and high speeds of biological systems. However, local sparse representations in ANNs generally suffer from poor generalisation and scaling, resulting in decreased performance with each additional layer in the network (Lagani et al., 2022a).

Our research addresses these Hard-WTA challenges by advancing the integration of Hebbian learning into modern deeper CNN architectures for unsupervised feature extraction through three key contributions. First, we integrated and optimised the computational efficiency of Hard-WTA in Journé et al. (2022)’s architecture. Second, we furthered biological realism through Bienenstock, Cooper, and Munro (BCM) learning rule (Bienenstock et al., 1982), novel spatial lateral inhibition, pre-synaptic, temporal, and homeostatic competition mechanisms that mirror cortical processing (Blakemore et al., 1976). Third, we developed architectural improvements including depthwise separable convolutions and residual connections, reducing parameter count by 6.6x.

To assess performance and explore a suitable Hard-WTA model and architecture integrating these individual innovations, we implemented 14 distinct Hebbian feature extraction configurations, and a comparative end-to-end backpropagation model. These configurations used a two-phase training approach: unsupervised Hebbian learning for feature extraction in convolutional layers, followed by a supervised classifier layer trained via backpropagation. This approach was chosen to match earlier Hebbian CNN research models, which while maintaining biological plausibility in the feature learning phase, enable direct comparisons with conventional end-to-end backpropagation CNNs.

By integrating our best-performing learning mechanism (lateral inhibition and BCM Hebbian learning) with Hard-WTA, we achieved 75.2% CIFAR-10 test mean accuracy on the final 50% epochs in Journé et al. (2022)’s 3-CNN layer architecture, setting a new SOTA Hard-WTA performance. This represents a 10.6% improvement over the previous Hard-WTA SOTA Hebbian performance of 64.6% using the same network depth (Miconi, 2021).

Our experimental results evidenced that Hebbian learning for feature extraction leads to a comparable performance to end-to-end backpropagation methods, which achieved 75.2% mean accuracy under the same training conditions (20 epochs and same architecture). Our advancements were also sustained when additional datasets were used, with the Optimal-HardWTA setup reaching 98% test mean accuracy on MNIST and 69.5% on STL-10.

These results significantly advance the SOTA for Hard-WTA Hebbian models, addressing key challenges faced by the Hebbian-AI community: maintaining competitive performance across multiple layers while reducing computational requirements and enhancing biological realism.

To validate our implementation and results, we developed a comprehensive evaluation framework that incorporates a PyTorch-based, modular implementation enabling direct integration and comparison of different Hebbian approaches alongside visualisation tools such as Uniform Manifold Approximation and Pro-

105 jection (UMAP
 106 citepmcinnes2018umap) for feature embeddings, receptive fields and weight dis-
 107 tributions of neurons for the analysis of competition and feature learning.

108 2. Background and Related Work

109 2.1. Convolutional Neural Networks

110 Neurons, both biological and artificial, function as fundamental units of
 111 information processing. In Artificial Neural Networks (ANNs), neurons are
 112 mathematical models mimicking the behaviour of their biological counterparts.
 113 Learning in both systems involves modifying connection strengths or synapses
 114 between neurons.

115 In an ANN, a neuron pass operates only on local information during a for-
 116 ward pass, taking, taking an input vector x , applying synaptic weights w , and
 117 producing an output $y = f(\sum x^T w)$, where f is an activation function. This
 118 output is then compared against a target outcome (ground truth) and the error
 119 is used to update weights globally through the network via the backpropaga-
 120 tion algorithm and gradient descent optimisation rule. (Rumelhart et al., 1986;
 121 Rojas and Rojas, 1996; Amari, 1993).

122 Convolutional Neural Networks (CNNs) (LeCun et al., 1998) are deep ANN
 123 architectures built on key principles of local receptive fields and hierarchical
 124 feature extraction. These architectural elements, inspired by the organisation
 125 of the cat’s visual cortex (Fukushima, 1980), make CNNs particularly effective
 126 for visual processing tasks. In CNNs, neurons are structured into layers for hier-
 127 archical processing of visual data. At every level, neurons generate responses to
 128 patterns in their receptive field by aggregating inputs from units in a preceding
 129 layer, resulting in an expanded receptive field and learning progressively complex
 130 and abstract hierarchical relationships or representations from the data, with
 131 lower layers typically extracting simple patterns like edges or textures while
 132 higher layers combine these into sophisticated representations such as shapes or
 133 objects, thus enabling more effective understanding and classification.

134 While CNNs have traditionally been trained using backpropagation, their
 135 fundamental architectural principles are valuable regardless of the learning al-
 136 gorithm employed. Modern CNN models often incorporate additional features,
 137 such as integrated residual blocks and Depthwise Separable Convolutions (DSC),
 138 inspired by biological processes to enhance their effectiveness.

139 Residual blocks allow training for very deep backpropagation networks by
 140 providing skip connections between layers (He et al., 2016) mimicking the hier-
 141 archical feedback pathways found in biological visual systems. DSCs separate a
 142 typical convolution operation into two separate convolutional operations (Chol-
 143 let, 2017) for computational efficiency and preventing overfitting through pa-
 144 rameter reduction. These features can also be used alongside Hebbian learning.
 145 Hence, following the rationale to improve biological plausibility and efficiency,
 146 we integrated residual blocks and DSC into our Hebbian-CNN framework.

Our depthwise implementation follows Chollet (2017) Depthwise Separable Convolution (DSC) approach, which modifies standard CNN operations by dividing them into two steps: depthwise convolution (applies a single filter per input channel) and pointwise convolution (applies 1x1 convolution across all channels). This approach not only reduces parameters and computations while maintaining performance but also increases biological plausibility (Tomen et al., 2021). DSC achieves this through the separation of spatial and feature combination operations, sparser connections between layers, and independent processing of each channel. Recent research (Babaiee et al., 2024) has shown that kernels in backpropagation-trained depthwise separable networks exhibit centre-surround receptive fields similar to those found in biological visual systems.

2.2. Fundamental Challenges and Biological Implausibilities of Backpropagation

Backpropagation faces fundamental optimisation challenges that limit its effectiveness. The algorithm often struggles with local convergence in nonlinear optimisation problems, frequently becoming trapped in suboptimal local minima (Sexton and Gupta, 2000). Recent critiques have highlighted additional fundamental issues: backpropagation requires unrealistic precision in weight updates, cannot effectively handle temporal dependencies, and struggles with credit assignment across multiple timescales (Hinton, 2022). Despite its effectiveness in training ANNs, backpropagation is assumed to possess little biological plausibility. (Apparaju and Arandjelović, 2022; Lillicrap et al., 2020; Song et al., 2020; Bredenberg et al., 2023).

Backpropagation requires error signals from non-directly connected neurons (Lillicrap et al., 2020), whereas biological systems rely on local interactions for synaptic plasticity. It needs the transpose of the weight matrix during backward pass, while biological neurons use unidirectional synapses (Weight Transport problem) (Apparaju and Arandjelović, 2022). The error computation requires complete forward and backward passes, unlike biological neurons, which do not exhibit such precise coordination (Update Locking problem) (Song et al., 2020). It relies on top-down supervision signal through a global loss function (Lillicrap et al., 2020), whereas biological learning involves more localised error correction.

Dale’s Principle (Eccles, 1976), a fundamental concept in neuroscience, states that neurons release the same neurotransmitters at all their synapses, leading to either exclusively excitatory or inhibitory effects. This biological constraint is typically violated in artificial neural networks trained through backpropagation (Cornford et al., 2020), where single neurons can have both positive and negative weights.

These biological limitations directly translate to implementation challenges in neuromorphic computing (Schuman et al., 2022), creating bottlenecks in parallel processing, while its heavy energy and memory requirements, and synchronised update needs limit deployment on edge devices and real-time applications.

2.3. Hebbian Learning

Backpropagation’s biological limitations substantiate the significance and potential of Hebbian learning for theoretical research and practical applica-

191 tions. Hebbian learning (Hebb, 1949; Lagani et al., 2023), follows the principle
 192 *neurons that fire together, wire together*². Neural updates are reliant solely on
 193 the activity of locally connected neurons. These neurons undergo a completely
 194 unsupervised learning process, as they do not require input from an external
 195 teacher or an error signal, and can identify statistical patterns in input data.
 196 The local nature of Hebbian learning eliminates the need for backward passes
 197 and target signals through the network, reducing memory requirements and
 198 enabling parallel weight updates. This locality makes Hebbian learning partic-
 199 ularly attractive for neuromorphic hardware implementation (Schuman et al.,
 200 2022), edge computing applications, and scenarios with limited labelled data.

201 The biological soundness of Hebbian learning, specifically its continuous
 202 adaptation to input patterns and local learning rules, enables systems to nat-
 203 urally adjust to changing input distributions and maintain stability without
 204 global coordination. These properties contrast with backpropagation systems
 205 which typically require complete retraining to adapt to new patterns and can
 206 become unstable when input distributions shift from their training data.

207 The development of Hebbian learning rules has evolved to address key chal-
 208 lenges in neural network stability and functionality. The basic Hebbian rule,
 209 which forms the foundation of this evolution, is formally expressed as:

$$\Delta w(t) = \eta y(x, w)x \quad (1)$$

210 where $y(x, w)$ or y is the post-synaptic activation or output of the neuron,
 211 x is the pre-synaptic activation or input of a neuron, and η the learning rate.
 212 While this basic rule captures the essence of Hebbian learning by strengthen-
 213 ing connections between co-active neurons, it suffers from unbounded weight
 214 growth or Long Term Potentiation (LTP), leading to instability. This limita-
 215 tion prompted the development of three significant variants that we examine
 216 here for their relevance to our work in stable learning dynamics.

217 Due to its simplicity and effectiveness in deep network architectures, we
 218 build our implementation upon Grossberg’s Instar rule (Grossberg, 1976). This
 219 rule introduces weight w as a decay proportional to post-synaptic activity y ,
 220 enabling both weight growth and decay based on input-output correlations:

$$\Delta w(t) = \eta y(x - w) = \eta(yx - yw) \quad (2)$$

221 The equation is presented in two mathematically equivalent forms to em-
 222 phasise different interpretations of the learning process. The first expression,
 223 $\eta y(x - w)$, highlights the biological intuition: the weight change is proportional
 224 to the difference between the input signal x and the current weight w , gated
 225 by the post-synaptic activity y . The second expression, $\eta(yx - yw)$, reveals the

²Originally formulated as "When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency, as one of the cells firing B, is increased. (Hebb, 1949, p. 62)"

rule’s dual mechanism: a Hebbian term yx for weight growth and an normalisation term yw for weight decay or Long Term Depression (LTD), providing inherent stability without requiring complex normalisation schemes or additional parameters.

Our model also implements the BCM rule (Bienenstock et al., 1982) due to its strong biological plausibility and effectiveness in preventing runaway synaptic growth. The rule’s dynamic threshold mechanism closely mirrors biological synaptic modification processes, making it particularly suitable for our focus on biologically-inspired learning systems. Belonging to the family of pre-synaptic gating rules, it inverts the roles of x and y from Equation 2:

$$\Delta w(t) = \eta x \psi(y - \theta) = \eta x y (y - \theta) \quad (3)$$

The equation is presented in two equivalent forms to highlight different aspects of the rule. The first expression uses the nonlinear function ψ , which represents the general form of the modification threshold function. In the specific implementation we adopt, this function takes the form $\psi(y - \theta) = y(y - \theta)$, leading to the second expression. This quadratic form ensures that weight modifications exhibit both potentiation and depression depending on the post-synaptic activity level relative to the threshold θ , creating a natural stability mechanism in the learning process.

While Sanger’s rule (Sanger, 1989), an extension of Oja’s original work (Oja, 1982), achieves weight normalisation and enables online Principal Component Analysis, its requirement for non-local synaptic information $\sum y_j w_j$ makes it less suitable for biologically plausible learning mechanisms and thus is not implemented in our model. For completeness, the rule is expressed as:

$$\Delta w_i(t) = \eta y_i (x - \sum_{j=1}^i y_j w_j) \quad (4)$$

This theoretical Hebbian foundation underpins the weight update rule in Hopfield networks (Hopfield, 1982), where the Hebbian principle is applied to store patterns as attractors in the network. Variants like dense associative memory (Krotov, 2023) extend this connection by leveraging nonlinear interactions, further demonstrating the versatility of Hebbian principles.

2.4. Neural Competition and Stability Mechanisms

Neural competition ensures different neurons learn to respond to different input patterns, preventing redundant feature learning. Through competitive mechanisms, neurons in each layer specialise in detecting distinct features, from simple edges and textures in early layers to more complex patterns in deeper layers. This specialisation enables the emergence of hierarchical representations, where higher layers combine and build upon the features detected by lower layers. While Hebbian learning provides biologically-plausible learning, it tends toward instability and redundancy without competition, as neurons typically converge to respond to the same dominant features in the input. We

implemented and compared several competition and stability mechanisms to investigate their performance and adequacy.

Winner-Takes-All (WTA) competition serves as the primary mechanism for driving neural competition in Hebbian learning, with two main variants: A) Hard WTA (Rumelhart and Zipser, 1985) allows only the neuron with the maximum activation (winning neuron) to update weights, promoting sparse representations and distinct feature specialisation. This mechanism can be applied within layers, across channels, or the whole network, providing flexibility in how competition is structured. B) Soft WTA (Nowlan, 1989) uses a softmax function with a temperature parameter to allow distributed learning while maintaining competition, providing more nuanced feature representations at the cost of more complex parameter tuning.

We investigated the integration of other competition and stability strategies with WTA competition architectures. New architectural configurations were thus built by independently combining: C) Anti-Hebbian learning (Choe, 2022; Foldiak, 1990) which actively decorrelates neural activities by weakening synaptic strengths between co-activated neurons with lateral connections, supporting the WTA-driven specialisation through explicit decorrelation D) Lateral inhibition (Gabbott and Somogyi, 1986) enhances WTA competition by modulating neuron activity based on neighbouring neurons' activity levels, particularly valuable in visual processing tasks where local contrast enhancement directly influences feature detection quality. E) Pre-synaptic competition (Rasmussen and Willshaw, 1993) reflects biological systems' resource constraints and complements WTA by regulating input signal competition, affecting learning outcomes and representation efficiency.

Our network stability enhancements focused on two independent key mechanisms which neural systems employ to regulate activity: synaptic traces and homeostatic plasticity. F) We implemented synaptic traces (Morris, 2006), a form of short-term plasticity, which enable neurons to retain information about their recent activation history, influencing their response to subsequent inputs. This temporal integration of activity plays a crucial role in learning and memory formation. G) We also employed homeostatic plasticity, (Turrigiano and Nelson, 2004) which represents another fundamental biological mechanism where neurons adjust their properties to maintain stable activity levels while preserving their ability to respond to relevant stimuli. This process helps prevent neural overexcitation and ensures efficient coding of input patterns across varying conditions.

2.5. Previous Hebbian-CNN Integration

Recent research has incorporated Hebbian learning principles and neural competition into CNNs, with varying approaches to Winner-Take-All (WTA) competition yielding different trade-offs between performance and biological plausibility. A key characteristic in these approaches is the separation between unsupervised feature learning and supervised classification. The convolutional layers learn representations through purely local Hebbian updates without labels or error signals, while only the final classifier layer uses supervised learning. For

proper comparison with existing research, our models followed the same general scheme.

Journé et al. (2022) set the current Hebbian benchmark with 80% accuracy on CIFAR-10. Their implementation combined soft-WTA competition with subtractive weight competition learning, which is mathematically akin to a variant of the Grossberg Instar rule.

$$\Delta w_{ik}^{(SoftHebb)} = \eta \cdot y_k \cdot (x_i - u_k \cdot w_{ik}). \quad (5)$$

where w_{ik} is a synaptic weight from a pre-synaptic neuron i with activation x_i , u_k is the original post-synaptic output of neuron k , and y_k is the result of the post-synaptic softmax competition. While this leading approach represents significant progress, it required a fourfold increase in the neurons per layer, highlighting ongoing efficiency challenges.

Grinberg et al. (2019) achieved the leading performance in Hard-WTA, implementing a single CNN layer architecture with patch normalisation, reaching 72.2% accuracy on CIFAR-10 (a natural image dataset). Amato et al. (2019) further advanced the hard-WTA approach with cosine similarity activation, achieving 98.55% accuracy on MNIST (a simple dataset of handwritten digits) but only 64% on CIFAR-10 using a two-layer CNN. Data whitening emerged as an essential pre-processing step for enhanced performance with hard-WTA competition. Their work revealed fundamental limitations in scaling hard-WTA to deeper architectures, as its performance markedly decreased with additional layers. Building on this foundation, Lagani et al. (2022a) maintained 60% accuracy on CIFAR-10 with a deeper five-layer model by implementing Sanger’s rule.

Miconi (2021) reached the highest Hard-WTA accuracy for a shallow network with 64.6% in a three CNN-layer model using a hybrid approach combining hard-WTA, homeostasis, new Triangle Activation function, and extensive pruning. Despite this improvement in accuracy in a deeper architecture, it is trained through gradient-based backpropagation learning, using surrogate losses which are equivalent to Hebbian learning rules.

Still, these Hebbian approaches incorporating WTA competition demonstrate Hebbian learning’s advantages in convergence speed, performance with limited data, and resilience against adversarial attacks (Gupta et al., 2022). Recent research (Lagani, 2024) suggests several promising directions for improvement, including incorporating inter-layer feedback and top-down connections, implementing pre-synaptic competition, applying the BCM learning rule, and developing spatial decorrelation in WTA competition.

Our work addressed the challenge of scaling hard-WTA competition in deeper architectures, leveraging its key advantages over soft-WTA: it reflected the binary firing states of biological neurons, enables energy-efficient implementations, and crucially, enforces true sparseness by silencing all but the strongest neurons.

Integration of the theoretical foundations of Hebbian learning, neural competition, and stability mechanisms from the biological neural networks into modern convolutional neural networks promises more efficient and biologically plausible

352 deep learning systems, allowing networks to learn from limited data, exhibit
 353 greater noise robustness, and consume fewer computational resources. This
 354 approach bridges biological and artificial systems, potentially enabling more
 355 resilient and flexible learning. In this paper, we first searched for a suitable
 356 combination of these mechanisms, exploring their accuracy and potential to en-
 357 hance the formation of hierarchical representations, incorporating hierarchical
 358 layers of increasingly abstract, sparse representations shaped by competition.
 359 Then we selected the 4 most suited configurations based on their accuracy for
 360 further evaluation and analysis.

361 3. Methodology

362 This section outlines the PyTorch-based Hebbian-CNN framework adopted.
 363 We designed and implemented 15 distinct configurations to systematically evalu-
 364 ate their components and interactions and build a naturalistic optimal architec-
 365 ture based on their performance as measured by accuracy values. To this aim, we
 366 developed a configurable architecture using different learning algorithms, Heb-
 367 bian Grossberg Instar and BCM learning, two main neural competition mecha-
 368 nisms, hard and soft WTA, and additional temporal, homeostatic, pre-synaptic
 369 and spatial competition mechanisms. These mechanisms were implemented as
 370 optional components in a customisable Hebbian layer. Each element could be
 371 enabled, disabled, or combined into distinct configurations to allow us to ex-
 372 plore their performance. Architectural changes, including depthwise separable
 373 convolutions and its enhancement through skip connections in residual blocks
 374 were evaluated.

375 The design philosophy behind this work encompasses computational effi-
 376 ciency, code readability, modular architecture, parametric flexibility, and ex-
 377 tensibility. We prioritised a GPU-accelerated implementation leveraging the
 378 PyTorch framework, known for its efficient parallel processing capabilities in
 379 deep learning applications. The modular structure closely mirrors that of Py-
 380 Torch, providing an intuitive organisational system familiar to researchers ex-
 381 perienceed with this machine learning library. The code is available at: [https://github.com/Julian-JN/Advancing-the-Biological-Plausibility-and-](https://github.com/Julian-JN/Advancing-the-Biological-Plausibility-and-Efficacy-of-Hebbian-Convolutional-Neural-Networks)
 382 [-Efficacy-of-Hebbian-Convolutional-Neural-Networks](https://github.com/Julian-JN/Advancing-the-Biological-Plausibility-and-Efficacy-of-Hebbian-Convolutional-Neural-Networks).
 383

384 3.1. Learning Pipeline

385 A two-phase approach was used to ensure feature learning biological plausi-
 386 bility while having a standard classifier system. Unsupervised Hebbian learning
 387 was employed for feature extraction in convolutional layers, followed by super-
 388 vised backpropagation linear classifier training on the frozen features.

389 The framework implemented a consistent, unsupervised feature learning
 390 pipeline across all Hebbian configurations, enabling feature extraction without
 391 requiring labelled data or global error signals. Figure 1 illustrates the Hebbian
 392 learning process in each CNN layer. Each forward pass followed these steps:

1. Pre-synaptic competition is applied to modify weights w before any calculations if enabled
2. Post-synaptic activities are computed through convolution between modified weights and input
3. Lateral inhibition modulates post-synaptic activities through the DoG kernel when enabled
4. Post-synaptic competition mechanisms are applied when enabled:
 - Hard-WTA selects maximally activated neurons
 - Temporal Competition uses synaptic traces for temporal specialisation
 - Homeostatic Competition adjusts selectivity based on input statistics
5. Weights are updated using the chosen learning rule (Grossberg, BCM, or SoftWTA Grossberg)
6. Weight normalisation is applied to maintain stable learning dynamics

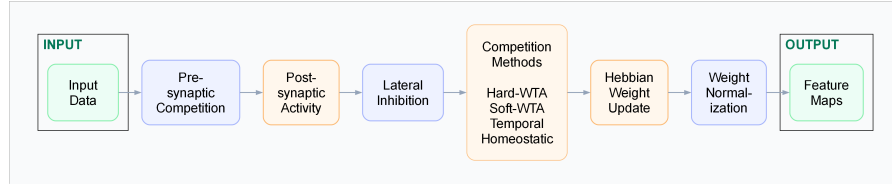


Figure 1: Diagram of the Hebbian learning process in our custom Hebbian-CNN layer. Green boxes correspond to input/output, blue boxes indicate optional processing, and orange boxes designate core learning and competition mechanisms.

This pipeline can be configured through layer parameters to activate different combinations of mechanisms based on research requirements.

3.2. Dataset Selection and Pre-Processing

To evaluate our framework’s performance and compare it with existing Hebbian CNN approaches, CIFAR-10 (Krizhevsky et al., 2009), MNIST (LECUN, 1998) and STL10 (Coates et al., 2011) were chosen as the benchmark dataset. The initial experimental configurations aimed at selecting the optional learning mechanisms and replicating the literature conducted on CIFAR-10. CIFAR-10 consists of 60,000 RGB images of 32x32 pixels, MNIST consists of 70,000 grayscale images of 28x28 pixels, and STL-10 consists of 13,000 RGB images of 96x96 pixels across 10 classes. Data processing included implementing random horizontal flip augmentations and zero-phase component analysis (ZCA) whitening. ZCA whitening was computed as improvements in performance were found for Hard-WTA competition (Amato et al., 2019).

421 3.3. Core Learning and Competition Mechanisms

422 Previous non-gradient-based implementations of synaptic plasticity using
 423 the Grossberg Instar rule with Hard-WTA (Amato et al., 2019) and Soft-WTA
 424 (Journé et al., 2022) were built into the system to be replicated as experimental
 425 controls and foundational elements for investigating enhanced biological learning
 426 approaches.

427 3.3.1. Grossberg Hebbian Learning Implementation

428 The Grossberg Instar stability modification of the basic Hebbian rule was
 429 used as the primary learning algorithm. The process began with computing
 430 post-synaptic activities through convolution:

$$y = w * x \quad (6)$$

431 where $*$ denotes the convolution operation between weights w and pre-
 432 synaptic activities x .

433 Following Lagani’s approach (Lagani et al., 2022a), we refined Hard-WTA
 434 neural selectivity by computing the post-synaptic activity using a cosine sim-
 435 ilarity function involving the postsynaptic convolution operation. With this
 436 approach, neurons fire strongly only when inputs closely match the learnt fea-
 437 ture, increasing feature specialisation. The post-synaptic activity y is calculated
 438 as:

$$y = \text{sim}(x, w) = \frac{x * \hat{w}}{\|x\|}, \quad (7)$$

439 where x is the input tensor, $\hat{w}$ is the L2-normalised weight kernel:

$$\hat{w} = \frac{w}{\|w\|_2}$$

440 The operation $x * \hat{w}$ denotes the 2D convolution, which naturally handles the
 441 different shapes of input and weight tensors by sliding the weight kernel across
 442 the input.

443 The norm $\|x\|$ is computed locally for each position where the convolution
 444 is applied:

$$\|x\| = \sqrt{x^2 * \mathbf{1} + \epsilon}$$

445 where $\mathbf{1}$ is an all-ones filter with the same shape as w , and $\epsilon = 10^{-8}$ prevents
 446 division by zero. This ensures that the computed similarity aligns with classical
 447 cosine similarity but is applied locally in a convolutional context.

448 The Grossberg weight update rule was then calculated for the convolutional
 449 neural network as:

$$\Delta w = (y * x) - \left(\sum y\right)w. \quad (8)$$

450 The derivation of Equation 8 from Equation 2 can be found in Appendix
 451 A.1 . These updates were normalised and stored in a buffer for efficiency and
 452 analysis.

3.3.2. Hard-WTA Competition

Hard Winner-Takes-All (WTA) competition was implemented to achieve sparse and specialised feature representations. For each spatial location (h, w) , competition took place across channels to create a binary mask $\text{mask}_{WTA}(y)$ that outputs 1 for the maximum activation across channels and 0 otherwise. The final post-synaptic activities are then computed as:

$$y_{final} = y \odot \text{mask}_{WTA}(y) \quad (9)$$

Our implementation enhanced Amato’s approach (Amato et al., 2019) by harnessing PyTorch’s GPU matrix and convolutional operations for efficient parallel computation, as suggested by Lagani et al. (2022b). This allowed weights to update during the forward pass without requiring gradients or backpropagation, significantly improving computational efficiency while maintaining biological plausibility.

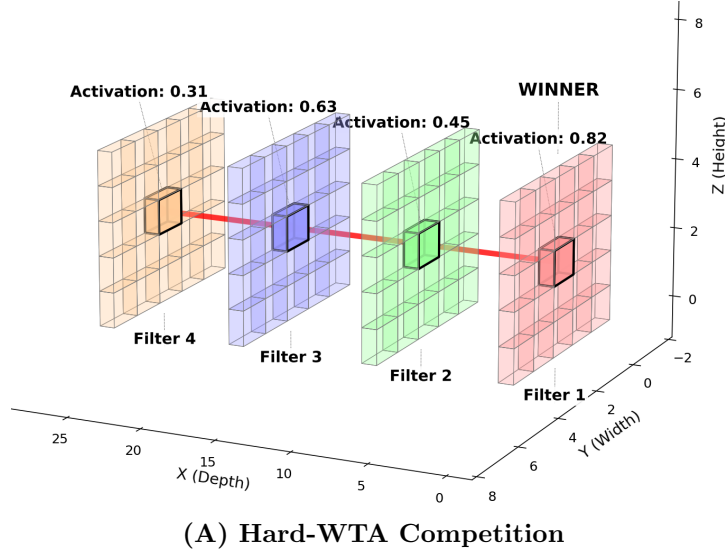


Figure 2: Implementation of Hard-WTA competition in CNNs influences neurons situated at the same spatial position across various filters.

3.3.3. SoftHebb Implementation

To establish a comprehensive comparison with current state-of-the-art Hebbian methods, we implemented SoftHebb as detailed in Journé et al. (2022). Unlike in Hard-WTA, SoftHebb applied soft competition to postsynaptic activity through a Softmax function with an inverse temperature parameter (Equation 5). A key distinction of this method is its use of a simple form of anti-Hebbian learning (negating SoftHebb’s weight update) for all neurons except the maximally activated one, creating a more nuanced form of competition than the binary selection of Hard-WTA (p.4 Journé et al. (2022)).

474 The combination of these three core procedures (learning algorithm and
 475 hard/soft-WTA) provided the foundation for our framework and the modularity
 476 of its components allowed for a systematic evaluation of their individual and
 477 combined effects on the network’s performance.

478 3.4. BCM Learning Rule

479 We implemented the BCM learning rule (Equation 3) within CNNs to eval-
 480 uate its unsupervised feature learning capacity compared to standard Hebbian
 481 approaches. Our convolutional implementation started by computing winner-
 482 take-all (WTA) activities for each neuron:

$$y_{WTA} = y \odot \text{mask}_{WTA}(y) \quad (10)$$

483 where $\odot$ represents element-wise multiplication. For each output channel, we
 484 maintained an adaptive threshold θ that tracked the average squared activity
 485 of WTA neurons through an exponential moving average:

$$\theta_t = (1 - \alpha)\theta_{t-1} + \alpha\mathbb{E}[y_{WTA}^2] \quad (11)$$

486 where α is the `theta_decay` parameter controlling adaptation speed (set at a
 487 default value of 0.5), and $\mathbb{E}[\cdot]$ denotes the spatial mean. The BCM non-linearity
 488 $\psi(y - \theta)$ was then computed as:

$$\psi(y - \theta) = y_{WTA} \odot (y_{WTA} - \theta) \quad (12)$$

489 The final weight updates $x\psi(y - \theta)$ were computed by convolving the presyn-
 490 aptic input x with the BCM non-linearity and normalising to maintain con-
 491 sistent weight magnitudes across layers. Our method preserved the Hard-WTA
 492 competition mechanism, substituting the Grossberg rule, commonly employed
 493 in traditional Hebbian learning, with BCM’s adaptable threshold dynamics.

494 3.5. Temporal and Homeostatic Competition

495 Two competition strategies, grounded in biological stability principles and
 496 relying on the competitive model proposed by Lagani (2024), were implemented
 497 among neurons situated in the same spatial area but across channels to enhance
 498 biological realism and robustness.

499 We developed two selection mechanisms following Lagani’s proposals. Our
 500 *Temporal Competition* implementation tracked neuronal activity through an
 501 activation history buffer H_t of size T for each neuron:

$$H_t = y_{t-T}, y_{t-T+1}, \dots, y_t \quad (13)$$

502 From this buffer (with default size of 500), we computed a temporal threshold
 503 θ_t as the median of historical activations:

$$\theta_t = \text{median}(H_t) \quad (14)$$

504 This threshold-based approach promoted neurons demonstrating consistent ac-
 505 tivity over time, as only neurons whose activation y exceeded θ_t are permitted
 506 to update their weights.

507 For *Homeostatic Competition*, we applied an input-driven competition mech-
 508 anism that calculates normalised similarity scores between weights and input:

$$S_{c,h,w} = \frac{\langle x, w \rangle}{|w|_2 + \epsilon} \quad (15)$$

509 where $S_{c,h,w}$ represents these scores across channels and spatial locations, with
 510 $\epsilon = 10^{-10}$. Based on these similarities, we computed an adaptive threshold:

$$\theta_{adaptive} = \mu_S + k\sigma_S \quad (16)$$

511 where μ_S and σ_S are the mean and standard deviation of similarities respec-
 512 tively, and k is the competition factor (typically set to 2). This adaptive thresh-
 513 old dynamically adjusted neural plasticity, increasing when input patterns pro-
 514 duced high similarity scores to prevent overactivation, and decreasing with low
 515 scores to facilitate more neural updates. Only neurons with post-synaptic ac-
 516 tivities y exceeding $\theta_{adaptive}$ updated their weights.

517 Both selection mechanisms were designed to be compatible with additional
 518 competition methods like Hard-WTA to further promote efficient specialisation
 519 and sparse connectivity. We note these competitive selection mechanisms may
 520 limit BCM's inherent anti-Hebbian learning for sub-threshold neurons, poten-
 521 tially affecting feature specialisation that would normally occur through negative
 522 weight updates in traditional BCM implementations.

523 3.6. Pre-synaptic Competition

524 Pre-synaptic competition was enforced using synaptic couplings, modifying
 525 the weights w before calculating the postsynaptic activity. Three competition
 526 modes were developed across the input channel, where different input features
 527 compete to influence the same output. Each mode first computed an inverse
 528 weight magnitude ($m = \frac{1}{|w|+\epsilon}$) to promote competition between input features
 529 with weaker connections, drawing inspiration from homeostatic synaptic plastic-
 530 ity mechanisms described by Turrigiano (2012). This inverse weighting allows
 531 less established pathways to maintain competitive influence, though it repre-
 532 sents a conflict with Hard-WTA approaches that exclusively reinforce dominant
 533 connections. Here, ϵ is a small constant (1e-6) to prevent division by zero:

534 1. Linear Competition, defined as:

$$w_{eff} = \frac{m}{\sum_i m_i + \epsilon} \quad (17)$$

535 It normalises the weights to values between 0-1.

536 2. Softmax Competition, defined as:

$$w_{eff} = \text{softmax}(m) = \frac{e^{m_i}}{\sum_j e^{m_j}} \quad (18)$$

537 This creates a more pronounced competition where stronger connections
 538 are emphasised, and the weights summed together add up to 1.

539 **3. L2 Norm Competition, defined as:**

$$w_{eff} = \frac{m}{\sqrt{\sum_i m_i^2}} \quad (19)$$

540 This ensures that the sum of squared effective weights equals 1.

541 3.7. Lateral Inhibition

542 A fixed kernel created through a difference of Gaussians (DoG) function was
 543 designed to simulate lateral inhibition and surround modulation, as proposed
 544 by (Hasani et al., 2019). This kernel models how neurons in the visual cortex
 545 respond to stimuli in their receptive field and surrounding areas. The centre
 546 (excitatory) region represents the classical receptive field, while the surround
 547 (inhibitory) region models lateral inhibition from neighbouring neurons. Math-
 548 ematically it is defined as :

$$K_{SM}(x, y) = \frac{1}{K_{center}} \left(\frac{G_e(x, y)}{2\pi\sigma_e^2} - \frac{G_i(x, y)}{2\pi\sigma_i^2} \right) \quad (20)$$

549 where G_e and G_i represent the excitatory and inhibitory Gaussian functions,
 550 respectively, and σ_e and σ_i denote the standard deviations of the excitatory and
 551 inhibitory Gaussians (with default value of 1.2 and 1.4). This kernel strengthens
 552 synapses in the immediate spatial neighbourhood of a neuron, and weakens
 553 synapses further from this neighbourhood. When applied, lateral inhibition
 554 was computed by applying a convolutional kernel of size 5, with filter in the
 555 form of Equation 20.

556 3.8. Architectures

557 Two main architectures were implemented, Journe’s and Lagani’s. The for-
 558 mer was the default architecture used in all configurations unless stated other-
 559 wise. To investigate biologically-plausible learning mechanisms while maintain-
 560 ing computational efficiency, we explored three distinct architectural variants,
 561 specifically Journe along with its counterparts, the depthwise separable convo-
 562 lution, and the residual block. Each variant built upon Journe baseline archi-
 563 tecture while introducing specific modifications aimed at reducing parameters,
 564 improving performance, and enhancing biological realism. These architectures
 565 were evaluated independently to assess their individual contributions to the
 566 model’s capabilities.

567 The baseline architecture, Journe (Figure 3), consisted of 3 convolutional
 568 layers featuring padding, Triangle/Rectified Polynomial Unit activation function
 569 (Miconi, 2021; Journé et al., 2022), and an increased number of filters in each
 570 subsequent layer (quadrupling the channels at each layer). This design choice
 571 aligns with current SOTA Hebbian architectures (Journé et al., 2022), moving
 572 beyond the single-CNN layer hard-WTA approach of Grinberg et al. (2019).

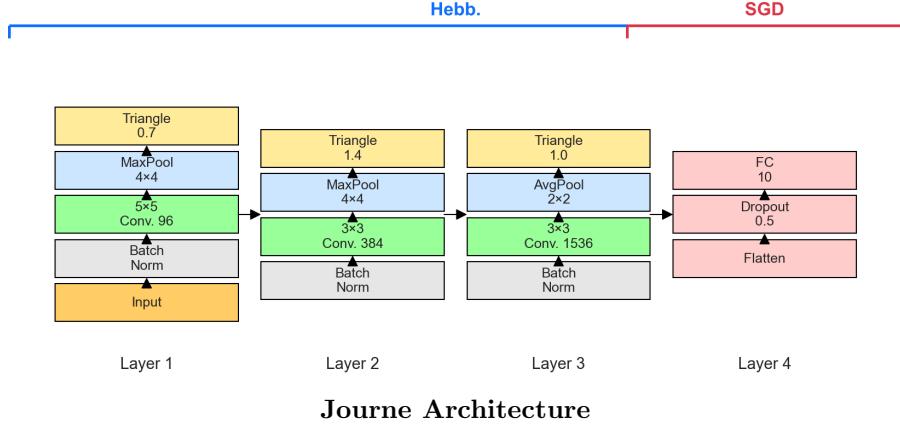


Figure 3: Visual representation of the main Journe Architecture, composed of 3 Hebbian CNN layers and a final backpropagation linear classifier layer.

For comparison, we also implemented Lagani’s architecture (Lagani et al., 2022a), a padding-free design with fewer channels per layer and 3-5 convolutional layers, enabling systematic evaluation of architectural effects on Hebbian learning and competition dynamics. Detailed specifications are provided in Appendix C.

3.8.1. Depthwise Separable Convolution

Our first architectural variant replaced standard convolutions in the Journe architecture with Depthwise Separable Convolutions to better approximate biological visual processing while reducing computational complexity. Furthermore, all our previously detailed Hebbian learning rules and competition mechanisms were incorporated into these convolutions to investigate their potential to produce centre-surround receptive fields.

In the depthwise component, we restricted competition and learning to operate independently within the spatial dimensions of each input channel. This was accomplished by computing activations and weight updates separately for each channel’s convolutional filter, preventing cross-channel competition. We then implemented the pointwise component as 1x1 standard convolutions that combined channel-wise features, with Hebbian learning rules and competition operating across the channel dimension. The pointwise layer’s weight updates were computed using the same learning rules and competition mechanisms as the depthwise layer, but applied to the channel-wise feature combinations rather than spatial patterns.

3.8.2. Residual Block

The second variant incorporated residual connections using Depthwise Separable Convolutions in an inverted bottleneck structure, motivated by the presence of feedback connections in biological neural circuits. Each block consisted

of three sequential operations: an initial pointwise convolution that expanded the channel dimension by a fixed factor, followed by a depthwise convolution for spatial feature learning, and a final pointwise convolution that projected features back to the original channel dimension. Skip connections bypassed these three operations, directly connecting the input to the output of each block.

This configuration was designed to match the Journe architecture’s feature map dimensions and channel counts at corresponding depths, enabling direct performance comparisons. The channel expansion factor of 4 and block placement were specifically chosen to maintain parameter counts comparable to Journe while preserving the network’s representational capacity. The complete architectural specifications, including layer dimensions and block placements, are detailed in Appendix C.

3.9. Dale’s Principle Weights

To pledge biological realism, all the models were designed to allow an optional adherence to Dale’s Principle. We ensured all synaptic weights were initialised and maintained excitatory values. This was achieved by using an absolute value function whenever an operation on weights took place. Weight changes still facilitated both Long-Term Potentiation (LTP) and Long-Term Depression (LTD), as both processes are crucial for plasticity and learning.

3.10. Analysis and Visualisation Methods

We implemented various visualisation techniques, including UMAP (McInnes et al., 2018), weight distributions and receptive fields, for qualitative and quantitative analysis of the network dynamics during training. Class separability was assessed through Uniform Manifold Approximation and Projection (UMAP) projections of layer feature embeddings, measuring the clustering quality of different class categories. Weight distributions were analysed using kernel density estimation to track the evolution of synaptic strengths.

To understand learned hierarchical feature representations, we implemented Projected Gradient Ascent (PGA) for receptive field visualisation. This technique optimises an input image I to maximise the activation A of specific neurons according to: $I_{t+1} = I_t + \eta \frac{\partial A}{\partial I} - \lambda I_t$ where η represents the learning rate and λ controls L2 regularisation to ensure visual coherence.

3.11. Configurations

To systematically evaluate our framework’s components and their interactions, we designed and implemented 15 total distinct experimental configurations: 3 of these replicated the results of published SOTA models, 11 which built upon previous Hard-WTA research approaches, and a comparative end-to-end backpropagation trained model. Each configuration represents a specific addition of either architectural design, competition mechanisms, or learning rule, allowing us to isolate and analyse their individual effects on network performance. Out of these configurations, the top four will be selected according to an accuracy criterion and thoroughly assessed.

641 3.12. Statistical Analyses

642 Post-initialisation behaviour performance of the selected configurations was
643 assessed over 5 different seed runs across the last 10 epochs of the test. De-
644 scriptives were computed, including arithmetic mean, median, standard devi-
645 ation and 99% confidence intervals, which were calculated using the Welch-
646 Satterthwaite equation to account for potential variance differences. We as-
647 sessed the normality of the distribution using the Shapiro-Wilk test with a p
648 >0.05 normality threshold for both paired models. We carried out pairwise
649 comparisons: for non-normally distributed data using the Friedman (Kruskal-
650 Wallis) test, and for normally distributed data using the ANOVA test. We chose
651 a conservative significance threshold of $p < 0.01$.

652 4. Experimental Settings

653 4.1. Experimental Configurations: Ablation Study

654 Table 1 lists all the experimental setups, outlining the architecture, types
655 of competition mechanisms, variants of the Hebbian learning rule for weight
656 updates, and any extra activations and features utilised. Table 2 summarises
657 the rationale behind each experimental setup, including the choice of specific
658 competition mechanisms, architectures, or learning rules.

659 The configurations were built upon two fundamental architectural approaches
660 established in previous research: the Journe CNN architecture from (Journe
661 et al., 2022), and the Lagani CNN architecture from (Lagani et al., 2022a). The
662 **SoftWTA** and **Lagani-HardWTA** and/or **Lagani_Deep-HardWTA** configurations
663 directly replicate these approaches respectively, serving as benchmarks for our
664 enhanced implementations.

665 Building upon these foundations, we developed a series of novel configura-
666 tions that integrate additional biologically-sound mechanisms or architectural
667 improvements. Unless specified otherwise, all HardWTA experiments were con-
668 ducted using the Journe architecture, incorporating the cosine similarity activa-
669 tion function and employing the Grossberg Instar update rule. Our configura-
670 tions explored combinations of Hard-WTA competition with presynaptic compe-
671 tition (**Presynaptic/HardWTA**), lateral inhibition (**Surr/HardWTA**), homeostatic
672 (**Homeostatic/HardWTA**) and temporal (**Temporal/HardWTA**) competition.
673 **No-WTA** evaluated the system performance with no competition mechanism.

674 We introduced architectural variants including depthwise separable convolu-
675 tions (**Depthwise-HardWTA**) and residual connections (**Residual-HardWTA**) to
676 investigate more efficient and biologically-inspired network structures. Either
677 the Grossberg Instar or BCM rule was employed to adjust the convolutional
678 weights.

679 The **Dale-Depthwise-Surr/HardWTA-BCM** configuration represents our
680 most biologically realistic implementation, uniquely combining three key ele-
681 ments: Dale’s Principle constraints on synaptic weights, depthwise separable
682 convolutions, and BCM Hebbian learning with Hard-WTA competition and
683 lateral inhibition.

Table 1: Hebbian Learning Initial Configurations and Features

Configuration	Architecture	Competition	Hebbian Rule
<i>Replications of Published SOTA models</i>			
Soft WTA	Journe	Soft WTA	GI
Lagani-HardWTA	Lagani	Hard WTA	GI
Lagani_Deep-HardWTA	Lagani	Hard WTA	GI
<i>Backpropagation Comparison</i>			
Backpropagation	Journe	N/A	BP
<i>Novel Implementations</i>			
HardWTA	Journe	Hard WTA	GI
HardWTA-BCM	Journe	Hard WTA	BCM
Presynaptic/HardWTA	Journe	Hard WTA+Pre	GI
Temporal/HardWTA	Journe	Hard WTA+Temp	GI
Homeostatic/HardWTA	Journe	Hard WTA+Hom	GI
Surr/HardWTA*	Journe	Hard WTA+Surr	GI
Depthwise-HardWTA [†]	Journe	Hard WTA	GI
Residual-HardWTA [‡]	Journe	Hard WTA	GI
Dale_Depth-Surr/HardWTA-BCM ^{§†}	Journe	Hard WTA+Surr	BCM
No-WTA	Journe	None	GI

Note: WTA: Winner-Take-All, GI: Grossberg Instar, BCM:

Bienenstock-Cooper-Munro, BP: Backpropagation, Surr: Lateral Surround Inhibition. Special features: *Surround lateral inhibition, [†]Depthwise separable convolutions, [‡]Residual connections, [§]Dale’s Principle.

To establish a direct comparison with traditional deep learning approaches, we included the **Journe-Backpropagation** configuration, which implements end-to-end backpropagation training using the same architecture as our Hebbian implementations. This configuration serves as a control, allowing us to evaluate the relative performance of our biologically-inspired learning mechanisms against conventional gradient-based methods.

Lastly, an **Optimal-HardWTA** configuration was built which integrates the mechanisms shown to improve performance to serve as the optimal configuration.

Table 2: Purposes of Initial Experimental Hebbian Learning Configurations

Configuration	Purpose
<i>Replications of Published SOTA models</i>	
SoftWTA	Evaluate SoftHebb research
Lagani-HardWTA	Evaluate Lagani Hard-WTA research
Lagani_Deep-HardWTA	Evaluate Lagani Hard-WTA 4-layer research
<i>Backpropagation Comparison</i>	
Backpropagation	Evaluate pure backpropagation baseline
<i>Novel Implementations</i>	
HardWTA	Evaluate Hard-WTA in network with padding
HardWTA-BCM	Compare BCM to Grossberg-Instar rule
Presynaptic/HardWTA	Assess impact of presynaptic learning
Temporal/HardWTA	Evaluate temporal competition with/without WTA
Homeostatic/HardWTA	Evaluate statistical competition with Hard-WTA
Surr/HardWTA	Assess effect of surround modulation kernel
Depthwise-HardWTA	Evaluate depthwise equivalent of hard-WTA research
Residual-HardWTA	Evaluate residual equivalent of hard-WTA research
Dale_Depth-Surr/HardWTA-BCM	Evaluate improved biological realism in a CNN
No-WTA	Evaluate basic learning without competition

4.2. Network Training

The training process for the configurations consisted of two distinct phases for all configurations except the end-to-end backpropagation comparison:

- 1. Unsupervised Feature Learning Phase:** Only convolutional layers were trained for one epoch with a batch size of 64 using Hebbian learning rules. Hard-WTA configurations used a learning rate of 0.1 across all layers, while Soft-WTA used a custom per-layer learning rate schedule as detailed in (Journé et al., 2022). A normal random distribution with a large radius range was used to initialise the SoftWTA Hebbian-CNN weights (detailed in (Journé et al., 2022) and A.2) as learning only occurred with this specific setup. All Hard-WTA and backpropagation experiments were initialised with the default PyTorch Kaiming Uniform distribution (He et al., 2015). ZCA-Whitening was applied only for configurations with Hard-WTA competition mechanisms. Custom visualisation tools were used throughout to analyse the network’s representational abilities. After feature learning, convolutional layer weights were frozen.

- 2. Supervised Classifier Training Phase:** The single-layer classifier head underwent training using backpropagation over 20 epochs employing the Adam optimizer with a learning rate of 0.001. A scheduler was used to cut the learning

rate by half every 2 epochs starting from the 10th epoch, and a dropout rate of 0.5 was applied to avoid overfitting, as performed by Journé et al. (2022). The classifier layer had as input size the flattened features of the CNN, and has as output size the number of classes (10 for CIFAR-10, MNIST and STL10). Both training and test sets were evaluated at each epoch. The pure backpropagation comparison model was trained end-to-end through the Adam optimiser with a learning rate of 0.001.

For consistent comparisons, all experimental configurations were trained using the same predetermined random seed, while distinct seeds were used across the 5 runs per configurations for the statistical analysis. To enable a quantitative comparison among configurations, we applied standard evaluation metrics on the test set during each epoch of the classifier training phase, namely accuracy, precision, recall, and F1 score.

Accuracy is calculated as:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

F1 score is calculated as:

$$\text{F1} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2TP}{2TP + FP + FN}$$

We report accuracy results in decimal form (e.g., 0.92) in all tables, figures, and statistical analyses, while we use percentage form (e.g., 92%) in the main text to facilitate comparisons with other papers in the literature.

The CIFAR-10 dataset was trained using both the 3-layer Journe and the 3-4 layer Lagani architectures. Training with different datasets entailed some specific modifications: the use of a greyscale variant of the Journe and Lagani architectures for MNIST, and the incorporation of an extra 4th layer in the Journe architecture for the larger images of the STL-10 dataset and usage of the 4-layer Lagani variant to match the identical architecture of Journé et al. (2022), as elaborated in Appendix C.

5. Results

5.1. Classification Accuracy on CIFAR-10

5.1.1. Replications of Published Research

Our initial configurations focused on reproducing previous research results. The **SoftWTA** configuration, with 5.9M parameters, achieved 79.2% accuracy and 0.79 F1 score, confirming the benchmark performance reported by Journé et al. (2022). To establish additional baselines, we implemented the **Lagani-HardWTA** configuration, and its deeper 4-CNN layer counterpart **Lagani_Deep-HardWTA** (with 0.39M and 0.8M parameters), which achieved 59.7% and 52.8% accuracy respectively, close to Amato’s research (Amato et al., 2019) which reached 59.69% and 49% in the 3 and 4-layer architectures.

Table 3: Top CIFAR-10 Test Performance Metrics for all Experimental Configurations across 20 Epochs

Configuration	Accuracy	F1 Score
<i>Replications of Published SOTA models</i>		
SoftWTA	0.792	0.79
Lagani-HardWTA	0.597	0.59
Lagani_Deep-HardWTA	0.528	0.53
<i>Backpropagation Comparison</i>		
Backpropagation	0.777	0.77
<i>Novel Implementations</i>		
HardWTA	0.748	0.75
HardWTA-BCM	0.753	0.75
Presynaptic/HardWTA	0.651	0.65
Temporal/HardWTA	0.746	0.74
Homeostatic/HardWTA	0.685	0.69
Surr/HardWTA	0.757	0.76
Depthwise-HardWTA	0.719	0.72
Residual-HardWTA	0.747	0.75
Dale_Depthwise-HardWTA-BCM	0.673	0.67
No-WTA	0.199	0.600
<i>Soft-WTA with BCM+Lateral Inhibition</i>		
SoftWTA-Surr-BCM	0.73	0.73
<i>Optimal Hard-WTA Configuration</i>		
Optimal-HardWTA	0.76	0.76

5.1.2. Ablation Study

A significant enhancement in Hard-WTA performance was achieved with the new configurations developed, directly building upon the previously described baselines. With the same number of parameters (5.9M), the **HardWTA** configuration achieved 74.8% accuracy and 0.75 F1 score, marking the first time Hard-WTA performance has surpassed Grinberg et al. (2019) 's SOTA performance (72.2%) in a deeper network. This integration of the Journe architecture with Hard-WTA mechanisms yielded a substantial improvement of 15.1% accuracy over the 3-CNN layer Lagani baseline, and an increase of 9.2% over the previous 3-CNN layer Hard-WTA SOTA (64.6%) by Miconi (2021).

Further biologically inspired enhancements rendered additional Hard-WTA improvements. The **Surr/HardWTA** configuration achieved 75.7% accuracy, further improving our new SOTA performance for Hard-WTA implementations. Alternative learning rules with BCM in **HardWTA-BCM** also improved performance to 75.3%, suggesting further evidence of biological efficiency of BCM.

While scoring slightly lower in accuracy (74.6%), the temporal competition variant, **Temporal/HardWTA**, also contributed to highlight the robustness of

our enhanced Hard-WTA framework. Other competition mechanisms showed varying degrees of effectiveness, with the **Homeostatic/HardWTA** and **Presynaptic/HardWTA** configurations achieving 68.5% and 65.1% accuracy respectively.

Our architectural variants focused on efficiency and biological plausibility. The **Depthwise-HardWTA**, which significantly reduced model parameters from 5.9M to 0.9M (a factor of 6.6), and the **Residual-HardWTA** with 4.03M parameters, achieved 71.9% and 74.7% accuracy respectively. They maintained robust performance while reducing network parameters. The putatively most biologically realistic network, **Dale_Depthwise-Surr/HardWTA-BCM**, only achieved 67.3% accuracy with exclusively centre-surround filters at Layer 2-3 (Appendix B). This result suggests that inhibitory weights may be essential for attaining high performance. The significant importance of competition mechanisms was emphasised by the **No-WTA** setup, which by lacking a competition mechanism reached only 19.9% accuracy. However, it is worth mentioning that it learnt features that enabled it to exceed the balanced 10-class baseline random accuracy of 10%.

5.1.3. Backpropagation Comparison

The Journe architecture trained through end-to-end backpropagation (**Backpropagation**) achieved 77.7% accuracy. This comparison demonstrates that our enhanced Hard-WTA Hebbian approaches (75% accuracy) can successfully approximate traditional gradient-based performance in shallow networks. It should be noted that the advantages of backpropagation become more evident with the addition of more deep layers and prolonged training durations, leading to significant performance improvements.

5.1.4. Combinatorial Setups: Hard and Soft-WTA Combined with Lateral Inhibition and BCM

The **Optimal-HardWTA** configuration was set up by integrating the most effective, biologically inspired mechanisms that enhanced performance. The configuration built included Hard-WTA, layer-specific kernel parameters for Lateral Inhibition, and the BCM learning rule’s **theta_decay**, similar to Journé et al. (2022) with per-layer Soft-WTA parameters

For Layer1, the settings were $\sigma_e = 1.2$, $\sigma_i = 1.3$, with a lateral kernel size of 5, **theta_decay**= 0.3, and a learning rate of 0.1. For Layer2, $\sigma_e = 1.0$, $\sigma_i = 1.2$, with a lateral kernel size of 3, **theta_decay**= 0.35, and a learning rate of 0.08. Lastly, for Layer3, the parameters were $\sigma_e = 0.8$, $\sigma_i = 1.1$, a lateral kernel size of 3, **theta_decay**= 0.35, and a learning rate of 0.05.

As hypothesised, the resulting **Optimal-HardWTA** configuration achieved the highest values: 76% accuracy with a F1-Score of 0.76.

Following the success of these combinations, the **SoftWTA-Surr-BCM** configuration was evaluated using BCM and Lateral Inhibition alongside soft-WTA and anti-Hebbian updates for non-maximum neurons, to measure if these additional competition mechanisms could be integrated successfully without affecting the

intrinsic cross-entropy minimisation of the original formulation by (Journé et al., 2022).

The **SoftWTA-Surr-BCM** configuration reached 73% accuracy and an F1-score of 0.73, suggesting a decrease of the implicit cross-entropy loss function optimisation of the SoftHebb model (Moraitis et al., 2022).

5.2. Comparison of Prior Work and New Hard-WTA State-of-the-Art

A detailed evaluation of our framework’s performance was conducted by comparing the results of our replications of the two published Hebbian SOTA models, **SoftWTA** and **Lagani-HardWTA**, and the **Backpropagation** against our highest performing Hard-WTA model, **Optimal-HardWTA**.

The following section presents detailed analyses of these selected configurations, examining their performance through accuracy plots and consistency across multiple random seeds and additional datasets, class clustering capabilities through UMAP embeddings, weight distribution characteristics and learnt receptive field patterns.

5.2.1. Statistical Analysis of CIFAR-10

We ran each model configuration (**SoftWTA**, **Backpropagation**, **Optimal-HardWTA**, and **Lagani-HardWTA**) 5 times with different random initialisation seeds. This approach allowed us to account for variability introduced by weight initialisation and stochastic training processes, providing a more reliable estimate of each model’s performance characteristics.

Table 4: Performance comparison on CIFAR-10 test dataset (last 10 epochs)

Model	Mean	Median	99% CI	Std Dev	Max
Backpropagation	0.752	0.754	[0.747, 0.757]	0.014	0.769
Optimal-HardWTA	0.752	0.753	[0.750, 0.755]	0.007	0.758
SoftWTA	0.775	0.781	[0.768, 0.783]	0.019	0.795
Lagani-HardWTA	0.597	0.600	[0.593, 0.601]	0.010	0.604

Table 4 presents the mean CIFAR-10 performance metrics and statistical comparisons for our learning algorithms across the final 50% test epochs.

Prior to conducting our statistical comparisons, we performed the Shapiro-Wilk tests to assess normality of our data distributions. The results were as follows: Backpropagation ($W=0.978$, $p=0.466$); Optimal-HardWTA ($W=0.921$, $p=0.003$); SoftWTA ($W=0.875$, $p=8.12 \times 10^{-5}$); Lagani-HardWTA ($W=0.877$, $p=9.10 \times 10^{-5}$). These results led us to reject the hypothesis of normality. Due to the non-normal distribution of the data, we employed the non-parametric Friedman (Kruskal-Wallis) tests for pairwise comparisons.

Our proposed **Optimal-HardWTA** model achieved comparable performance metrics to the traditional **Backpropagation**. The two model accuracy levels did not statistically differ ($H(1) = 0.11$, $p = 0.736$).

The existing **SoftWTA** approach significantly outperformed our **Optimal-HardWTA** model ($H(1) = 35.32$, $p = 2.80 \times 10^{-9}$), evidencing the effectiveness

of this established biologically-plausible alternative. Similarly, the accuracy performance of the **SoftWTA** method was significantly superior to **Backpropagation** ($H(1) = 33.06, p = 8.95 \times 10^{-9}$).

Our **Optimal-HardWTA** model substantially outperformed the **Lagani-HardWTA** baseline ($H(1) = 74.26, p = 6.84 \times 10^{-18}$;). **Backpropagation** ($H(1) = 74.26, p = 6.84 \times 10^{-18}$;) and **SoftWTA** ($H(1) = 74.26, p = 6.85 \times 10^{-18}$;) also significantly outperformed this baseline model.

We additionally conducted a separate analysis on the final epoch to evaluate the ultimate performance of each model after the learning process was complete. Before performing our statistical analyses on the last epoch, we performed the Shapiro-Wilk test to assess normality of our data distributions using a significance threshold of $\alpha = 0.05$. The results indicated normal distributions for all models: **Backpropagation** ($W=0.809, p=0.096$); **Optimal-HardWTA** ($W=0.965, p=0.844$); **SoftWTA** ($W=0.950, p=0.738$); **Lagani-HardWTA** ($W=0.905, p=0.441$). Shapiro-Wilk test analyses did not allow us to reject the default hypothesis of normal distribution of data. Hence, normality was assumed, and a series of ANOVA tests were conducted to compare pairs of configurations.

Our proposed **Optimal-HardWTA** model (75.6% mean accuracy performance value during the last epoch) achieved comparable performance metrics to the traditional **Backpropagation** approach (76.9%) ($F(1, 8) = 6.43, p = 0.035$), not differing significantly from this traditional gradient-based method.

The **SoftWTA** model (79.45%) significantly outperformed the **Backpropagation** method ($F(1, 8) = 23.61, p = 0.001$;) . Likewise, the **SoftWTA** model significantly performed better than our **Optimal-HardWTA** approach ($F(1, 8) = 451.25, p = 2.54 \times 10^{-8}$;) , highlighting the superior performance of this established method over our proposed biologically-plausible alternative.

Both our **Optimal-HardWTA** model and the **Backpropagation** method substantially outperform the **Lagani-HardWTA** baseline (60.3%). These differences were significantly reliable ($F(1, 8) = 4090.46, p = 3.97 \times 10^{-12}$; and $F(1, 8) = 936.93, p = 1.41 \times 10^{-9}$, respectively). The **SoftWTA** model also significantly outperformed **Lagani-HardWTA** ($F(1, 8) = 7260.67, p = 4.01 \times 10^{-13}$).

Figure 4 presents the boxplot distribution of mean test accuracy across multiple runs with different random seeds for each model. The **SoftWTA** model displayed the highest median accuracy (77.5%) with a wider interquartile range, indicating strong overall performance but with some variability across different initialisations. One notable outlier at the lower end suggests occasional suboptimal convergence.

Our **Optimal-HardWTA** model exhibits consistency, with the narrowest interquartile range among all models and lowest standard deviation (0.007). Its median performance (75.3%) closely aligns with the **Backpropagation** approach (75.4%), visually reinforcing our conclusion that these models perform comparably. However, the backpropagation variant shows greater variability, with a wider spread between the first and third quartiles.

The **Lagani-HardWTA** model is clearly separated from the other approaches, with substantially lower accuracy (median 60%) and no overlap with the per-

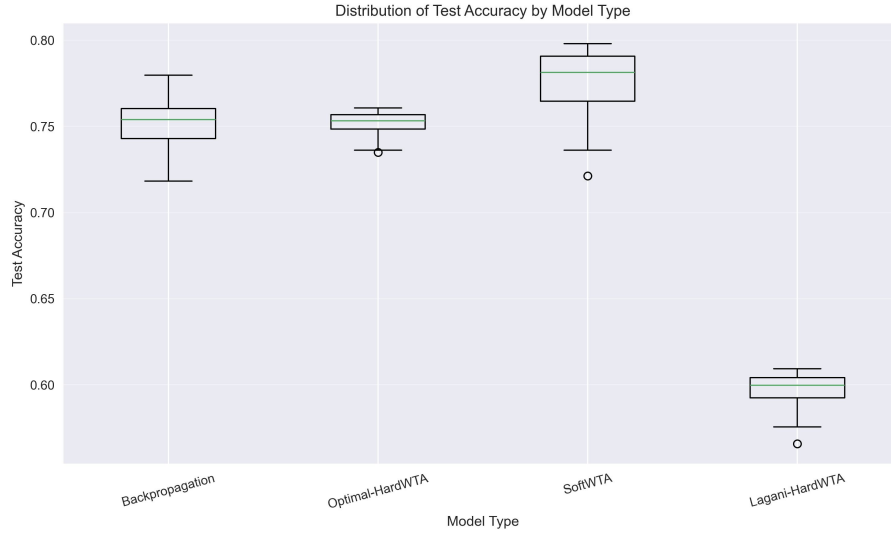


Figure 4: Boxplot of CIFAR-10 Test Accuracy by Configuration across different random seeds

886 formance distributions of the other models. This visual separation is consistent
 887 with the reported significance differences.

888 5.2.2. CIFAR-10 Accuracy Dynamics

889 Figure 5 displays accuracy per epoch during training and Figure 6 the test
 890 accuracy for the **Optimal-HardWTA** (solid blue line), **Backpropagation** (dashed
 891 orange line), **SoftWTA** (dotted green line), and **Lagani-HardWTA** (red dash-
 892 dotted line).

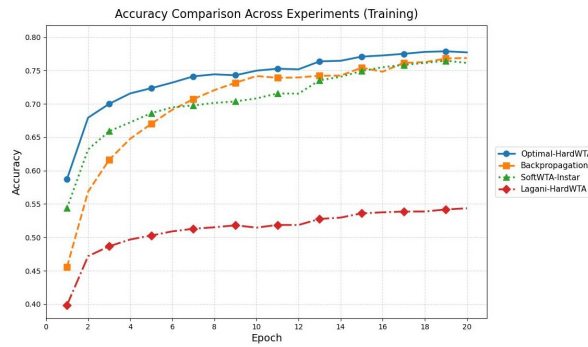


Figure 5: Accuracy values during CIFAR-10 training for Lagani-HardWTA (red-rhombus-dot-dash line), SoftWTA (green-triangle-dotted line), Backpropagation (orange-square-broken line), and Optimal-HardWTA (blue-circle-solid line)

893 Training and Test accuracy were remarkably higher and converged faster

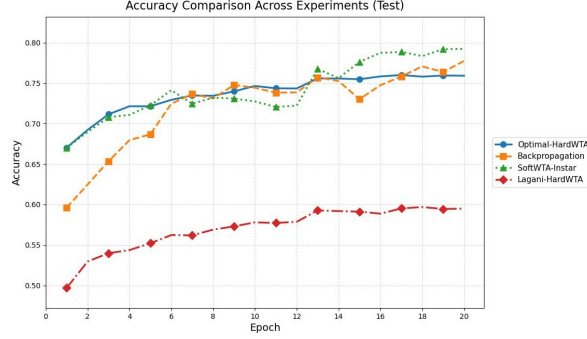


Figure 6: Accuracy values during CIFAR-10 test for Lagani-HardWTA (red-rhombus-dot-dash line), SoftWTA (green-triangle-dotted line), Backpropagation (orange-square-broken line), and Optimal-HardWTA (blue-circle-solid line)

in our **Optimal-HardWTA** than the previous Hard-WTA research **Lagani-HardWTA**. Moreover, the accuracy achieved by both **Optimal-HardWTA** and **SoftWTA** variants were comparable throughout training (Figure 5), with both Hebbian WTA approaches surpassing **Backpropagation** at each epoch in both higher accuracy and convergence, with a significant higher initial accuracy for **Optimal-HardWTA**. A similar pattern was observed during test (Figure 6), although the final accuracy level of the **SoftWTA** was slightly higher than other configurations.

5.2.3. Extension study to other datasets

A study was conducted on both MNIST and STL-10 datasets to assess generalisation across different data types and scales. All reported results are for the last 10 epochs during the testing phase of 5 random seed runs.

Table 5 shows the comparative performance of all model variants on the MNIST dataset.

Model	Mean	Median	99% CI	Std Dev	Max
Backpropagation	0.9853	0.9859	[0.9843, 0.9863]	0.0027	0.9884
Optimal-HardWTA	0.9829	0.9833	[0.9820, 0.9837]	0.0022	0.9850
SoftWTA	0.9787	0.9806	[0.9767, 0.9808]	0.0053	0.9830
Lagani-HardWTA	0.9468	0.9473	[0.9457, 0.9479]	0.0029	0.9493

We performed Shapiro-Wilk tests to assess normality of our data distributions. The results forced us to reject the normal distribution hypothesis: Backpropagation ($W=0.885$, $p = 1.62 \times 10^{-4}$); Optimal-HardWTA ($W=0.861$, $p = 3.13 \times 10^{-5}$); SoftWTA ($W=0.755$, $p = 9.22 \times 10^{-8}$); Lagani-HardWTA ($W=0.955$, $p=0.057$). Thus, we carried out non-parametric Friedman (Kruskal-Wallis) tests for pairwise comparisons.

Our proposed **Optimal-HardWTA** model achieved strong performance metrics (mean 98.29%). However, the small difference with the traditional **Backpropagation** approach (mean 98.53%) was statistically reliable, as confirmed by the Friedman test ($H(1) = 27.25$, $p = 1.79 \times 10^{-7}$). While traditional gradient-based methods maintain a slight advantage on the MNIST dataset, our biologically-inspired learning rule approaches comparable performance while maintaining biological plausibility. Notably, the **Optimal-HardWTA** approach demonstrates lower variability (std dev: 0.0022 vs. 0.0027), suggesting more stable learning dynamics across training runs.

It is worth noticing that unlike with the CIFAR-10 dataset, our **Optimal-HardWTA** model outperformed the **SoftWTA** approach (mean 97.87%) ($H(1) = 33.15$, $p = 8.53 \times 10^{-9}$). This result highlights the effectiveness of our approach over previous biologically-plausible alternatives. Both our model and the **Backpropagation** method significantly outperformed the **SoftWTA** model, with backpropagation showing the largest performance advantage ($H(1) = 54.63$, $p = 1.45 \times 10^{-13}$).

Optimal-HardWTA model outperformed the **Lagani-HardWTA** baseline (mean 94.68%). These differences were significantly reliable ($H(1) = 74.28$, $p = 6.78 \times 10^{-18}$). Similarly, both **Backpropagation** ($H(1) = 74.27$, $p = 6.80 \times 10^{-18}$) and **SoftWTA** ($H(1) = 74.27$, $p = 6.80 \times 10^{-18}$) also significantly outperform this baseline model.

We additionally conducted a separate analysis on the final epoch to evaluate the ultimate performance of each model after the learning process was complete. The results did not allow us to reject the null normal distribution hypothesis: **Backpropagation** ($W=0.833$, $p=0.148$); **Optimal-HardWTA** ($W=0.894$, $p=0.377$); **SoftWTA** ($W=0.912$, $p=0.482$); **Lagani-HardWTA** ($W=0.981$, $p=0.942$). Hence, normality was assumed, and a series of ANOVA tests were conducted to compare pairs of configurations.

Our proposed **Optimal-HardWTA** model (98.47% mean accuracy performance value during the last epoch) achieved comparable performance metrics to the traditional **Backpropagation** approach (98.59%), and no statistic differences were found between these two configurations ($F(1, 8) = 0.93$, $p = 0.363$). These results suggest that the **Optimal-HardWTA** model achieved a performance that could not be reliably distinguished from the traditional gradient-based methods at the conclusion of training on the MNIST dataset while maintaining biological plausibility.

The **Backpropagation** method performed similarly to the **SoftWTA** model (98.26%) in the final epoch ($F(1, 8) = 6.24$, $p = 0.037$). Our **Optimal-HardWTA** model significantly outperformed the **SoftWTA** approach at the final epoch ($F(1, 8) = 17.64$, $p = 0.003$). These results support a superior performance of our method over previous biologically-plausible alternatives.

Both our **Optimal-HardWTA** model and the **Backpropagation** method substantially outperform the **Lagani-HardWTA** baseline (94.8%) in the final epoch. The difference in performance was statistically reliable ($F(1, 8) = 3038.66$, $p = 1.30 \times 10^{-11}$; and $F(1, 8) = 711.21$, $p = 4.20 \times 10^{-9}$, respectively). The **SoftWTA** model also significantly outperformed this Hard-WTA baseline

960 $(F(1, 8) = 1819.02, p = 1.01 \times 10^{-10})$.

Table 6: Performance comparison on STL-10 test dataset (last 10 epochs)

Model	Mean	Median	99% CI	Std Dev	Max
Backpropagation	0.5821	0.5871	[0.5712, 0.5930]	0.0289	0.6185
Optimal-HardWTA	0.6949	0.7023	[0.6880, 0.7018]	0.0182	0.7091
SoftWTA	0.6603	0.6823	[0.6435, 0.6771]	0.0443	0.6931
Lagani-HardWTA	0.5296	0.5364	[0.5203, 0.5388]	0.0243	0.5521

961 To test the generalisation capabilities on more complex data, we also evalu-
 962 ated all models on the STL-10 dataset. Table 6 presents these results.

963 We performed Shapiro-Wilk tests to assess normality of our data distribu-
 964 tions. The results compelled us to reject the hypothesis of normality: Backprop-
 965 agation ($W=0.958, p = 0.070$); Optimal-HardWTA ($W=0.797, p = 7.62 \times 10^{-7}$);
 966 SoftWTA ($W=0.761, p = 1.24 \times 10^{-7}$); Lagani-HardWTA ($W=0.932, p =$
 967 0.007). Hence, we conducted the non-parametric Friedman (Kruskal-Wallis)
 968 tests for pairwise comparisons.

969 The performance of our **Optimal-HardWTA** model was the highest for this
 970 dataset (mean 69.49%), significantly outperforming all other approaches includ-
 971 ing the backpropagation method (mean 58.21%). These superior performance
 972 over **Backpropagation** was supported as confirmed by the Friedman test
 973 ($H(1) = 74.02, p = 7.72 \times 10^{-18}$). In addition, our **Optimal-HardWTA** ap-
 974 proach showed the lowest variability (std dev: 0.0182), indicating more stable
 975 learning dynamics across training runs.

976 **Optimal-HardWTA** model also significantly outperformed the **SoftWTA** vari-
 977 ant on STL-10 ($H(1) = 34.66, p = 3.92 \times 10^{-9}$), further suggesting better model
 978 accuracy or our approach on complex datasets. The **SoftWTA** model showed
 979 intermediate performance (mean: 66%), significantly outperforming both the
 980 backpropagation implementation ($H(1) = 53.00, p = 3.34 \times 10^{-13}$) and the
 981 **Lagani-HardWTA** method (mean 52.96%) ($H(1) = 67.19, p = 2.47 \times 10^{-16}$).
 982 Despite exhibiting the highest variability (std dev: 0.0443) among all tested
 983 models, **SoftWTA** still maintains a clear statistical advantage over backpropaga-
 984 tion on this more complex dataset. At this point it should be noted that the
 985 reported 76% accuracy on STL-10 by Journé et al. (2022) was achieved through
 986 longer training epochs (up to 100).

987 Our proposed **Optimal-HardWTA** model substantially outperformed the **La**
 988 **gani-HardWTA** baseline on the STL-10 dataset, with dramatic improvement
 989 observed ($H(1) = 74.26, p = 6.84 \times 10^{-18}$). The traditional **Backpropagation**
 990 approach also significantly outperformed the **Lagani-HardWTA** baseline ($H(1) =$
 991 $51.45, p = 7.33 \times 10^{-13}$).

992 To assess the final performance on STL-10, we conducted a further compar-
 993 ative analysis with the last epoch accuracy values. Shapiro-Wilk tests gave no
 994 support to reject the normality of the data on the last epoch: Backpropagation
 995 ($W=0.953, p = 0.755$); Optimal-HardWTA ($W=0.987, p = 0.969$); SoftWTA
 996 ($W=0.951, p = 0.746$); Lagani-HardWTA ($W=0.912, p = 0.483$). Therefore,
 997 ANOVA tests were conducted to compare performance.

Our proposed **Optimal-HardWTA** model (70.8% mean accuracy performance value during the last epoch) significantly outperformed the traditional **Backpropagation** approach (57.63%) ($F(1, 8) = 360.33$, $p = 6.14 \times 10^{-8}$). The **SoftWTA** model (69%) also significantly outperformed the **Backpropagation** method in the final epoch ($F(1, 8) = 229.50$, $p = 3.57 \times 10^{-7}$), further supporting the effectiveness of biologically-plausible learning approaches on challenging visual tasks.

Our **Optimal-HardWTA** model maintained its significant advantage over the **SoftWTA** variant in the final epoch ($F(1, 8) = 27.22$, $p = 8.05 \times 10^{-4}$). Interestingly, the traditional **Backpropagation** approach did not significantly outperform the **Lagani-HardWTA** baseline (57.6%) in the final epoch ($F(1, 8) = 10.02$, $p = 0.013$).

Our proposed **Optimal-HardWTA** model substantially outperformed the **Lagani-HardWTA** baseline in the final epoch, with dramatic improvement observed ($F(1, 8) = 1098.65$, $p = 7.49 \times 10^{-10}$). Similarly, the accuracy performance of **SoftWTA** model was also significantly better than the **Lagani-HardWTA** baseline in the final epoch ($F(1, 8) = 625.50$, $p = 6.99 \times 10^{-9}$). The results during this final epoch on STL-10, strongly support the superior performance of our biologically-plausible approach, which not only outperformed existing biologically-plausible alternatives but also exceeds the capabilities of traditional gradient-based methods on complex visual recognition tasks.

5.2.4. UMAP Class Clustering Embeddings

The UMAP, weight distributions, and PGA receptive fields were all executed using the identical fixed seed during the CIFAR-10 trial.

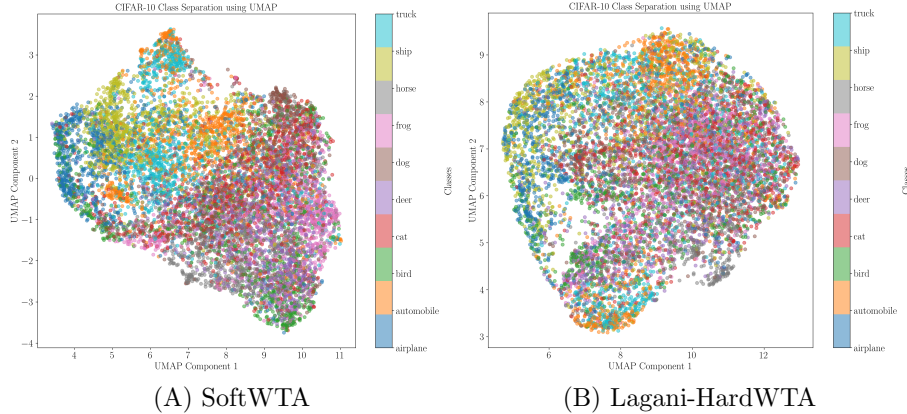


Figure 7: UMAP projection of CIFAR-10 class clusters from unsupervised feature extraction of the **SoftWTA** and **Lagani-HardWTA** configurations.

The UMAP embeddings at the final layer (Figures 7 and 8) revealed distinct patterns in feature organization. In these visualisations, dots of the same colour represent class labels, with Hebbian configurations showing clustering tenden-

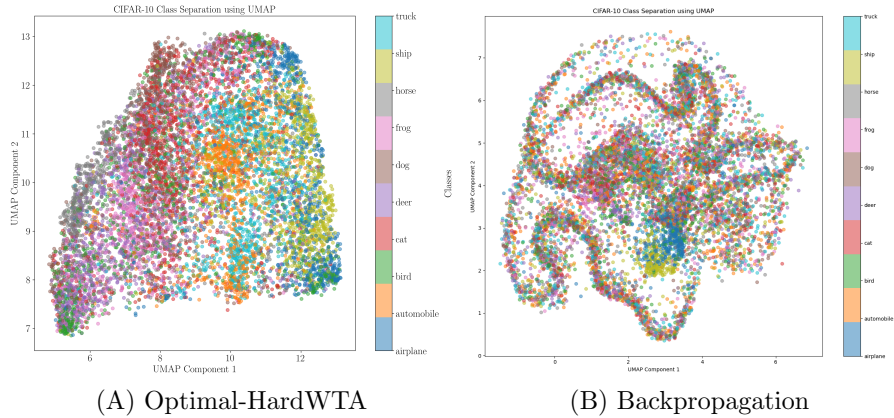


Figure 8: UMAP projection of CIFAR-10 class clusters from unsupervised feature extraction of the **Optimal-HardWTA** and **Backpropagation** configurations.

cies, where instances of the same class gather together and are separated from other classes.

The current Hebbian SOTA **SoftWTA** (Figure 7 A) demonstrated highly effective class separation, with clearly delineated clusters showing minimal overlap between classes. Some cross-clustering occurred at cluster boundaries, but the overall organisation suggested effective hierarchical learning with semantically related classes (e.g., automobile in orange and truck in cyan) positioned in proximity to each other while semantically different classes (e.g., ship in yellow and cat in red) located at distant positions.

The previous Hard-WTA benchmark **Lagani-HardWTA** (Figure 7 B) showed weaker clustering characteristics, with less defined boundaries and significant overlap between class clusters. This more diffuse distribution pattern indicates limited ability to form discriminative feature representations, explaining its lower classification performance.

Our enhanced Hard-WTA framework, **Optimal-HardWTA** (Figure 8 A), achieved clustering quality approaching that of **SoftWTA**. The visualisation reveals well-defined semantic clusters while maintaining natural class boundaries, validating that our enhanced competition mechanisms enable more effective feature learning than previous Hard-WTA approaches.

Intriguingly, while **Backpropagation** (Figure 8 B) achieved high classification accuracy, its UMAP embedding shows a radically different organisation of the learnt feature space. Unlike the Hebbian models which form clear cluster boundaries, the backpropagation model creates a more continuous manifold structure where classes flow into each other with no distinct separation.

This striking difference in representation structure, despite similar performance metrics, suggests that backpropagation and Hebbian learning discover fundamentally different solutions to the classification task. The backpropagation model appeared to learn non-linear discriminative features for optimisation, which create high-dimensional decision boundaries that do not necessarily

1054 cluster visually, rather than the discrete clustering preferred by competitive
 1055 Hebbian learning. The findings suggest that Hebbian learning offers greater
 1056 interpretability and explainability compared to backpropagation.

1057 5.2.5. Weight Distributions

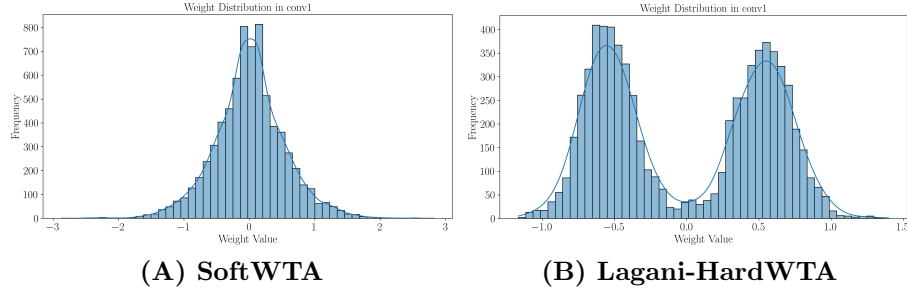


Figure 9: Weight distribution of **SoftWTA** and **Lagani-short-Hard/Cos-Instar** configurations for CIFAR-10 at Layer 1.

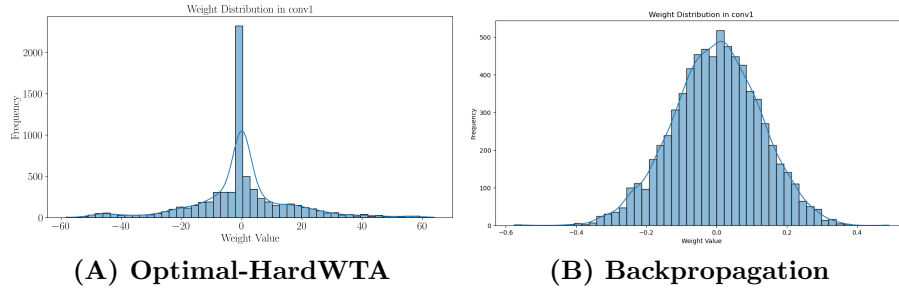


Figure 10: Weight distribution of **Optimal-HardWTA** and **Backpropagation** configurations for CIFAR-10 at Layer 1.

1058 Analysis of weight distributions (Figures 9 and 10) revealed distinct patterns
 1059 across learning approaches. **SoftWTA** (Figure 9A) maintained a broad normal
 1060 distribution across all layers, reflecting its soft competition approach. In con-
 1061 trast, **Lagani-HardWTA** (Figure 9B) showed a bimodal distribution with two
 1062 distinct peaks, suggesting more binary weight patterns emerging from the hard
 1063 competition mechanism.

1064 Our **Optimal-HardWTA**'s BCM learning rule (Figure 10A) produced a highly
 1065 sparse distribution with a pronounced peak near zero and long tails, closely mir-
 1066 roring the sparse connectivity observed in biological neural networks (Hawkins
 1067 et al., 2016). **Backpropagation** (Figure 10B) displayed a classic normal distri-
 1068 bution, characteristic of the global nature of gradient descent optimization.

1069 5.2.6. PGA Receptive Fields

1070 **SoftWTA** (Figure 11) demonstrated features that deviate from traditional
 1071 Gabor-like patterns. Layer 1 exhibited diverse patterns including solid colour

1072 detectors (visible in bright orange/green squares), chequerboard patterns cap-
 1073 turing local contrast and various colour-mixing effects. Layer 2 showed a tran-
 1074 sition to more complex combinations with fragmented patterns, smaller pixel
 1075 clusters, and intricate colour interactions. Layer 3 developed highly abstract,
 1076 sparse representations with noise-like features and scattered colour points, sug-
 1077 gesting specialised detectors for complex distributed patterns.

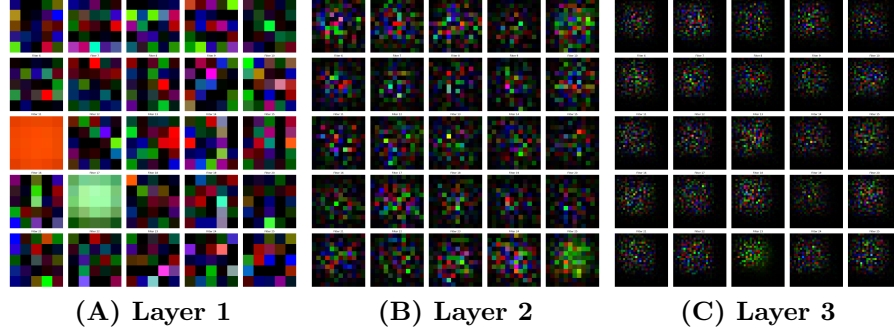


Figure 11: PGA Receptive fields for first 25 neurons at each layer of the **SoftWTA** configuration for CIFAR-10.

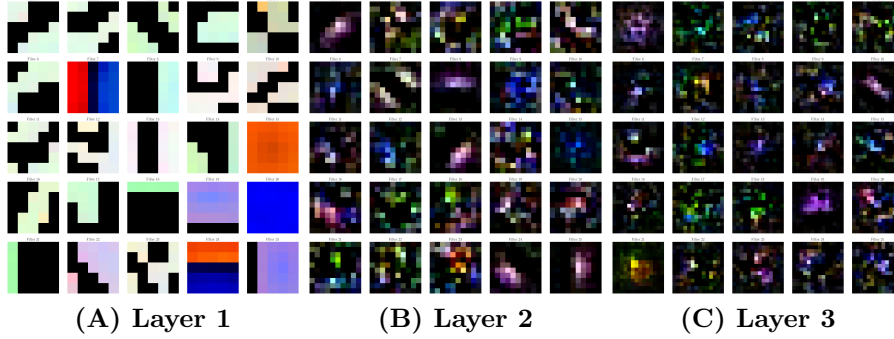


Figure 12: PGA Receptive fields for first 25 neurons at each layer of the **Lagani-HardWTA** configuration for CIFAR-10.

1078 **Lagani-HardWTA** (Figure 12) exhibited classical biological neural net-
 1079 work characteristics with Gabor-like edge detection. Layer 1 showed strong
 1080 binary patterns with sharp edges and clear geometric shapes, where black-and-
 1081 white regions and isolated pure colours (purple, blue, orange) demonstrate deci-
 1082 sive edge detection from hard-WTA competition. Layer 2 maintained structured
 1083 organisation with discernible orientation shapes and subtle colour blending, in-
 1084 dicated effective combination of edge detectors with colour information. Layer
 1085 3 preserved defined shapes compared to **SoftWTA**'s diffuse activations, reflecting
 1086 maintained feature selectivity in deeper layers.

1087 **Optimal-HardWTA** (Figure 13) showcased successful integration of hard-
 1088 WTA, lateral inhibition and BCM learning. Layer 1 combined sharp geomet-

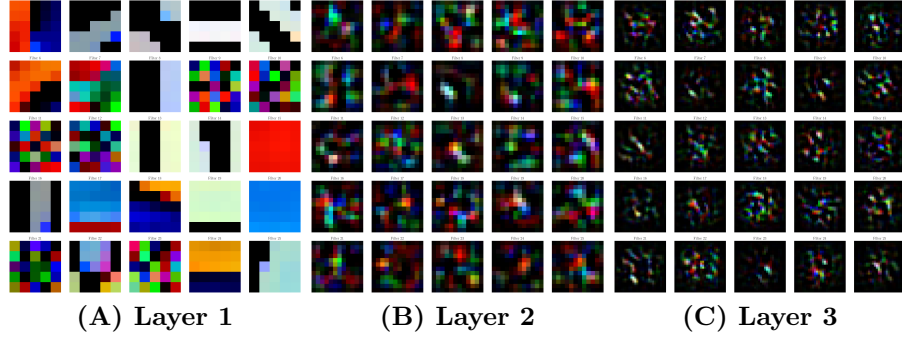


Figure 13: PGA Receptive fields for first 25 neurons at each layer of the **Optimal-HardWTA** configuration for CIFAR-10.

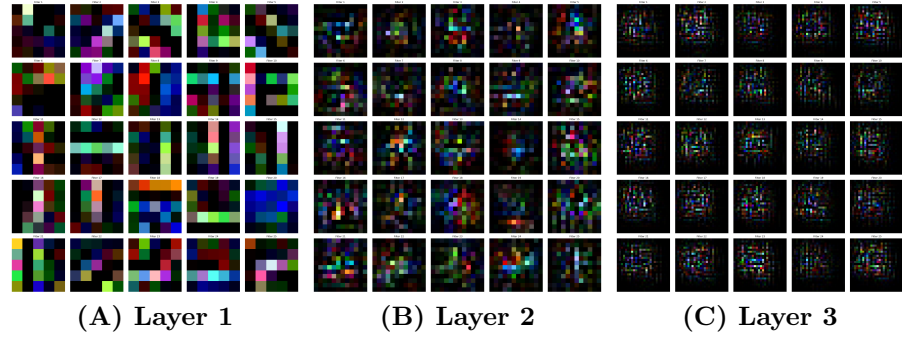


Figure 14: PGA Receptive fields for first 25 neurons at each layer of the **Backpropagation** configuration for CIFAR-10.

1089 ric boundaries comparable to Lagani’s Gabor-like edge detection, features that
 1090 characterises biological neural networks, with more sophisticated colour process-
 1091 ing, demonstrated through solid or multiple-colour colour detectors and complex
 1092 edge patterns with smoother gradients and colour information. Layer 2 exhib-
 1093 ited refined feature organisation with distinct colour combinations and organised
 1094 spatial arrangements, where lateral inhibition promoted coherent transitions be-
 1095 tween spatial regions. Layer 3 developed specialised detectors with sparse but
 1096 organised patterns maintaining structural coherence, suggesting effective preser-
 1097 vation of feature selectivity while enabling complex pattern detection.

1098 **Backpropagation** (Figure 14) revealed characteristics of gradient-based
 1099 optimisation. Layer 1 demonstrated strong directional selectivity with vertical
 1100 and horizontal structures alongside systematic colour organisation, appearing
 1101 more regularly than in Hebbian models. Layer 2 developed complex features
 1102 while maintaining organisational principles, with uniform pattern and colour
 1103 distribution suggesting global optimisation benefits. Layer 3 exhibited distinc-
 1104 tive grid-like patterns with architectural regularity, reflecting systematic feature
 1105 decomposition optimised for classification rather than biological plausibility.

5.3. Limitations

Several limitations were also encountered. While competitive with backpropagation in shallow networks, Hebbian learning may face challenges in deeper architectures, as the width-factor significantly increases the parameters in the architecture by a factor of 16x with each additional layer. Enhanced feature representation by eliminating redundant information among neurons might help resolve this problem. The small pixel images and 3-4 layer models limited the complexity of learnt representations, particularly in deeper layers. Performance varied significantly with architectural changes, highlighting the need for careful design considerations. Furthermore, the study focused on complete clean datasets, and limited training data scenarios and adversarial attacks remain to be explored.

6. Discussion

This work advances the state-of-the-art in biologically-plausible Convolutional Neural Networks by demonstrating that Hard Winner-Takes-All (WTA) competition can achieve performance comparable to backpropagation while maintaining biological realism. By reaching 75.2% mean accuracy on CIFAR-10, our implementation significantly improves upon previous Hard-WTA Hebbian approaches of the same network depth (64.6% by Miconi (2017)) and approximates backpropagation performance (mean 75.2%), albeit only in shallow networks, through careful integration of lateral inhibition and BCM learning rule. These results extend across datasets, achieving backpropagation-level performance on MNIST (98%) and superior on STL-10 (69.5% mean in under 20 epochs), all while demonstrating exceptional consistency across experimental runs.

Our results revealed that the performance of the Optimal-Hard WTA configuration was significantly superior to that of the Lagani-Hard WTA across all three datasets during the latter half of the testing period and at the end of it. Furthermore, it closely matched the accuracy achieved by Backpropagation and Soft WTA, outperforming both methodologies with the STL-10 under restricted train-time scenarios. These findings highlight the robustness and generality of the superior performance within the scope of this research.

The superiority of Hard-WTA over Soft-WTA (Journé et al., 2022) stems from its modelling of biological neural dynamics, particularly sparse coding where only a small fraction of neurons are active for any input. This sparsity translates into practical benefits: improved feature interpretability and reduced computational overhead during both training and inference.

Our approach addresses fundamental limitations of backpropagation by eliminating the need for labelled data and global error signals. The local nature of Hebbian learning, combined with Hard-WTA competition, enables efficient parallel processing and reduced memory requirements, critical advantages for neuromorphic computing and edge applications. Moreover, the emergence of orientation-selective neurons and centre-surround receptive fields in our network provides computational neuroscientists with a more faithful model of biological visual processing.

Two key challenges remain: scaling to larger datasets and enhancing biological realism. Future work should focus on developing efficient learning schemes for high-resolution images while controlling parameter growth in deeper architectures. Additionally, incorporating distinct excitatory and inhibitory populations could further bridge the gap between artificial and biological networks while potentially improving representation learning.

This research demonstrates that embracing biological constraints, particularly Hard-WTA competition and local Hebbian learning rules, can enhance both the efficiency and robustness of artificial neural networks while maintaining competitive accuracy. Our framework validates that principles underlying biological neural computation can be successfully translated into practical artificial systems, opening new possibilities for energy-efficient, interpretable, and robust AI architectures that better reflect biological intelligence.

Data and Code Availability

All code used in this study are publicly available. The code is available at: <https://github.com/Julian-JN/Advancing-the-Biological-Plausibility-and-Efficacy-of-Hebbian-Convolutional-Neural-Networks>. Any requests for additional data or materials should be directed to the corresponding author.

Declaration of competing interest

No competing interests

References

- Amari, S.-i. (1993). Backpropagation and stochastic gradient descent method. *Neurocomputing*, 5(4-5):185–196.
- Amato, G., Carrara, F., Falchi, F., Gennaro, C., and Lagani, G. (2019). Hebbian learning meets deep convolutional neural networks. In *Image Analysis and Processing-ICIAP 2019: 20th International Conference, Trento, Italy, September 9–13, 2019, Proceedings, Part I 20*, pages 324–334. Springer.
- Apparaju, A. and Arandjelović, O. (2022). Towards new generation, biologically plausible deep neural network learning. *Sci*, 4(4):46.
- Babaiee, Z., Kiasari, P. M., Rus, D., and Grosu, R. (2024). Neural echos: Depth-wise convolutional filters replicate biological receptive fields. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 8216–8225.
- Barth, A. L. and Poulet, J. F. (2012). Experimental evidence for sparse firing in the neocortex. *Trends in neurosciences*, 35(6):345–355.

1186 Bienenstock, E. L., Cooper, L. N., and Munro, P. W. (1982). Theory for the
1187 development of neuron selectivity: orientation specificity and binocular inter-
1188 action in visual cortex. *Journal of Neuroscience*, 2(1):32–48.

1189 Blakemore, C., Van Sluyters, R. C., and Movshon, J. A. (1976). Synaptic com-
1190 petition in the kitten’s visual cortex. In *Cold Spring Harbor Symposia on*
1191 *Quantitative Biology*, volume 40, pages 601–609. Cold Spring Harbor Labo-
1192 ratory Press.

1193 Bredenberg, C., Williams, E., Savin, C., Richards, B., and Lajoie, G. (2023).
1194 Formalizing locality for normative synaptic plasticity models. In Oh, A.,
1195 Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S., editors,
1196 *Advances in Neural Information Processing Systems*, volume 36, pages 5653–
1197 5684. Curran Associates, Inc.

1198 Choe, Y. (2022). Anti-hebbian learning. *Encyclopedia of Computational Neu-*
1199 *roscience*, pages 213–216.

1200 Chollet, F. (2017). Xception: Deep learning with depthwise separable convolu-
1201 tions. In *Proceedings of the IEEE conference on computer vision and pattern*
1202 *recognition*, pages 1251–1258.

1203 Coates, A., Ng, A., and Lee, H. (2011). An analysis of single-layer networks
1204 in unsupervised feature learning. In *Proceedings of the fourteenth interna-*
1205 *tional conference on artificial intelligence and statistics*, pages 215–223. JMLR
1206 Workshop and Conference Proceedings.

1207 Cornford, J., Kalajdzievski, D., Leite, M., Lamarquette, A., Kullmann, D. M.,
1208 and Richards, B. (2020). Learning to live with dales principle: Anns with
1209 separate excitatory and inhibitory units. *bioRxiv*, pages 2020–11.

1210 Demidovskij, A., Kazyulina, M., Salnikov, I., Tugaryov, A., Trutnev, A., and
1211 Pavlov, S. (2023). Implementation challenges and strategies for hebbian learn-
1212 ing in convolutional neural networks. *Optical Memory and Neural Networks*,
1213 32(Suppl 2):S252–S264.

1214 Eccles, J. C. (1976). From electrical to chemical transmission in the central
1215 nervous system: the closing address of the sir henry dale centennial sympo-
1216 sium cambridge, 19 september 1975. *Notes and records of the Royal Society*
1217 *of London*, 30(2):219–230.

1218 Foldiak, P. (1990). Forming sparse representations by local anti-hebbian learn-
1219 ing. *Biological cybernetics*, 64(2):165–170.

1220 Fukushima, K. (1980). Neocognitron: A self-organizing neural network model
1221 for a mechanism of pattern recognition unaffected by shift in position. *Bio-*
1222 *logical cybernetics*, 36(4):193–202.

1223 Gabbott, P. and Somogyi, P. (1986). Quantitative distribution of gaba-
1224 immunoreactive neurons in the visual cortex (area 17) of the cat. *Experimental*
1225 *brain research*, 61:323–331.

1226 Gerstner, W. and Kistler, W. M. (2002). Mathematical formulations of hebbian
1227 learning. *Biological cybernetics*, 87(5):404–415.

1228 Grinberg, L., Hopfield, J., and Krotov, D. (2019). Local unsupervised learning
1229 for image analysis. In *Real Neurons \{\mathcal{E}\} Hidden Units: Future directions at*
1230 *the intersection of neuroscience and artificial intelligence@ NeurIPS 2019*.

1231 Grossberg, S. (1976). Adaptive pattern classification and universal recoding:
1232 I. parallel development and coding of neural feature detectors. *Biological*
1233 *cybernetics*, 23(3):121–134.

1234 Gupta, M., Modi, S. K., Zhang, H., Lee, J. H., and Lim, J. H. (2022). Is
1235 bio-inspired learning better than backprop? benchmarking bio learning vs.
1236 backprop. *arXiv preprint arXiv:2212.04614*.

1237 Hasani, H., Soleymani, M., and Aghajan, H. (2019). Surround modulation:
1238 A bio-inspired connectivity structure for convolutional neural networks. *Ad-*
1239 *vances in neural information processing systems*, 32.

1240 Hawkins, J., Ahmad, S., Purdy, S., and Lavin, A. (2016). Biological and machine
1241 intelligence (bami). *Initial online release 0.4*.

1242 He, K., Zhang, X., Ren, S., and Sun, J. (2015). Delving deep into rectifiers: Sur-
1243 passing human-level performance on imagenet classification. In *Proceedings*
1244 *of the IEEE international conference on computer vision*, pages 1026–1034.

1245 He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image
1246 recognition. In *Proceedings of the IEEE conference on computer vision and*
1247 *pattern recognition*, pages 770–778.

1248 Hebb, D. O. (1949). *The organization of behavior: A neuropsychological theory*.
1249 Psychology press.

1250 Hinton, G. (2022). The forward-forward algorithm: Some preliminary investi-
1251 gations. *arXiv preprint arXiv:2212.13345*.

1252 Hopfield, J. J. (1982). Neural networks and physical systems with emergent
1253 collective computational abilities. *Proceedings of the national academy of*
1254 *sciences*, 79(8):2554–2558.

1255 Journé, A., Rodriguez, H. G., Guo, Q., and Moraitis, T. (2022). Hebbian
1256 deep learning without feedback. In *The Eleventh International Conference*
1257 *on Learning Representations*.

1258 Krithivasan, S., Sen, S., Venkataramani, S., and Raghunathan, A. (2022). Ac-
1259 celerating dnn training through selective localized learning. *Frontiers in Neu-*
1260 *roscience*, 15:759807.

1261 Krizhevsky, A., Hinton, G., et al. (2009). Learning multiple layers of features
1262 from tiny images.

1263 Krotov, D. (2023). A new frontier for hopfield networks. *Nature Reviews Physics*,
1264 5(7):366–367.

1265 Krotov, D. and Hopfield, J. J. (2019). Unsupervised learning by competing
1266 hidden units. *Proceedings of the National Academy of Sciences*, 116(16):7723–
1267 7731.

1268 Lagani, G. (2022). Recent advancements on bio-inspired hebbian learning for
1269 deep neural networks. In *SEBD*, pages 610–615.

1270 Lagani, G. (2024). Bio-inspired deep learning research directions. *Google Docs*.
1271 Document found in: [https://docs.google.com/document/d/1mRNC0AEkrW](https://docs.google.com/document/d/1mRNC0AEkrW7tcztbNuXdoz428ZCwbBW8kA2FZKZBh2o/edit)
1272 [7tcztbNuXdoz428ZCwbBW8kA2FZKZBh2o/edit](https://docs.google.com/document/d/1mRNC0AEkrW7tcztbNuXdoz428ZCwbBW8kA2FZKZBh2o/edit).

1273 Lagani, G., Falchi, F., Gennaro, C., and Amato, G. (2022a). Comparing the
1274 performance of hebbian against backpropagation learning using convolutional
1275 neural networks. *Neural Computing and Applications*, 34(8):6503–6519.

1276 Lagani, G., Falchi, F., Gennaro, C., and Amato, G. (2023). Synaptic plasticity
1277 models and bio-inspired unsupervised deep learning: A survey. *arXiv preprint*
1278 *arXiv:2307.16236*.

1279 Lagani, G., Gennaro, C., Fassold, H., and Amato, G. (2022b). Fasthebb: Scaling
1280 hebbian training of deep neural networks to imagenet level. In *International*
1281 *Conference on Similarity Search and Applications*, pages 251–264. Springer.

1282 LECUN, Y. (1998). The mnist database of handwritten digits.
1283 <http://yann.lecun.com/exdb/mnist/>.

1284 LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learn-
1285 ing applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–
1286 2324.

1287 Lee, K., Lee, K., Lee, H., and Shin, J. (2018). A simple unified framework for
1288 detecting out-of-distribution samples and adversarial attacks. *Advances in*
1289 *neural information processing systems*, 31.

1290 Lillicrap, T. P., Santoro, A., Marris, L., Akerman, C. J., and Hinton, G. (2020).
1291 Backpropagation and the brain. *Nature Reviews Neuroscience*, 21(6):335–346.

1292 McInnes, L., Healy, J., and Melville, J. (2018). Umap: Uniform mani-
1293 fold approximation and projection for dimension reduction. *arXiv preprint*
1294 *arXiv:1802.03426*.

1295 Miconi, T. (2017). Biologically plausible learning in recurrent neural networks
1296 reproduces neural dynamics observed during cognitive tasks. *Elife*, 6:e20899.

1297 Miconi, T. (2021). Hebbian learning with gradients: Hebbian convolutional
1298 neural networks with modern deep learning frameworks. *arXiv preprint*
1299 *arXiv:2107.01729*.

1300 Moraitis, T., Toichkin, D., Journé, A., Chua, Y., and Guo, Q. (2022). Soft-
1301 hebb: Bayesian inference in unsupervised hebbian soft winner-take-all net-
1302 works. *Neuromorphic Computing and Engineering*, 2(4):044017.

1303 Morris, R. (2006). Elements of a neurobiological theory of hippocampal func-
1304 tion: the role of synaptic plasticity, synaptic tagging and schemas. *European*
1305 *Journal of Neuroscience*, 23(11):2829–2846.

1306 Nowlan, S. (1989). Maximum likelihood competitive learning. *Advances in*
1307 *neural information processing systems*, 2.

1308 Oja, E. (1982). Simplified neuron model as a principal component analyzer.
1309 *Journal of mathematical biology*, 15:267–273.

1310 Rasmussen, C. E. and Willshaw, D. J. (1993). Presynaptic and postsynaptic
1311 competition in models for the development of neuromuscular connections.
1312 *Biological cybernetics*, 68(5):409–419.

1313 Rojas, R. and Rojas, R. (1996). The backpropagation algorithm. *Neural net-*
1314 *works: a systematic introduction*, pages 149–182.

1315 Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning repre-
1316 sentations by back-propagating errors. *nature*, 323(6088):533–536.

1317 Rumelhart, D. E. and Zipser, D. (1985). Feature discovery by competitive
1318 learning. *Cognitive science*, 9(1):75–112.

1319 Sanger, T. D. (1989). Optimal unsupervised learning in a single-layer linear
1320 feedforward neural network. *Neural networks*, 2(6):459–473.

1321 Schuman, C. D., Kulkarni, S. R., Parsa, M., Mitchell, J. P., Kay, B., et al.
1322 (2022). Opportunities for neuromorphic computing algorithms and applica-
1323 tions. *Nature Computational Science*, 2(1):10–19.

1324 Sexton, R. S. and Gupta, J. N. (2000). Comparative evaluation of genetic
1325 algorithm and backpropagation for training neural networks. *Information*
1326 *sciences*, 129(1-4):45–59.

1327 Song, Y., Lukasiewicz, T., Xu, Z., and Bogacz, R. (2020). Can the brain
1328 do backpropagation?—exact implementation of backpropagation in predic-
1329 tive coding networks. *Advances in neural information processing systems*,
1330 33:22566–22579.

1331 Strubell, E., Ganesh, A., and McCallum, A. (2020). Energy and policy con-
1332 siderations for modern deep learning research. In *Proceedings of the AAAI*
1333 *conference on artificial intelligence*, volume 34, pages 13693–13696.

- 1334 Tomen, N., Pintea, S.-L., and Van Gemert, J. (2021). Deep continuous net-
1335 works. In *International Conference on Machine Learning*, pages 10324–10335.
1336 PMLR.
- 1337 Turrigiano, G. (2012). Homeostatic synaptic plasticity: local and global mech-
1338 anisms for stabilizing neuronal function. *Cold Spring Harbor perspectives in*
1339 *biology*, 4(1):a005736.
- 1340 Turrigiano, G. G. and Nelson, S. B. (2004). Homeostatic plasticity in the de-
1341 veloping nervous system. *Nature reviews neuroscience*, 5(2):97–107.
- 1342 Wang, Q., Li, Y., and Li, R. (2024). Ecological footprints, carbon emissions,
1343 and energy transitions: the impact of artificial intelligence (ai). *Humanities*
1344 *and Social Sciences Communications*, 11(1):1–18.
- 1345 Werbos, P. J. (1994). *The Roots of Backpropagation: From Ordered Derivatives*
1346 *to Neural Networks and Political Forecasting*. John Wiley & Sons.
- 1347 Yoshida, T. and Ohki, K. (2020). Natural images are reliably represented by
1348 sparse and variable populations of neurons in visual cortex. *Nature commu-*
1349 *nications*, 11(1):872.
- 1350 Zarkeshian, P., Kergan, T., Ghobadi, R., Nicola, W., and Simon, C. (2022).
1351 Photons guided by axons may enable backpropagation-based learning in the
1352 brain. *Scientific Reports*, 12(1):20720.

1353 Appendix A. Details of Methods

1354 A.1. Derivation of Equation 8 from Equation 2

1355 Equation 2 defines the Grossberg Instar rule:

$$\Delta w = \eta y(x - w) = \eta(yx - yw). \quad (\text{A.1})$$

1356 where: y is the post-synaptic activation, x is the pre-synaptic input, w is
1357 the synaptic weight, yx is the Hebbian learning term and yw is a normalisation
1358 term preventing unbounded growth.

1359 In a convolutional neural network (CNN), weight updates occur over lo-
1360 calised receptive fields rather than single synapses. To extend this to CNNs, we
1361 replace the simple product yx with a convolution operation $*$:

$$yx \rightarrow x * y, \quad (\text{A.2})$$

1362 where $*$ denotes a 2D convolution:

$$(y * x)_{c,h,w} = \sum_{i,j} x_{c,h+i,w+j} y_{c,i,j}. \quad (\text{A.3})$$

1363 Similarly, the second term yw , which in the original rule represents a decay
1364 proportional to y , is implemented as:

$$yw = \left(\sum_{b,h,w} y_{b,h,w} \right) w. \quad (\text{A.4})$$

1365 Thus, the final convolutional update rule is:

$$\Delta w = (y * x) - yw. \quad (\text{A.5})$$

1366 This ensures weight updates occur based on local convolutional correlations
1367 while maintaining stability across spatial locations. This functions as a GPU op-
1368 timisation of Amato et al. (2019)’s research and also incorporates the Grossberg
1369 derivation applied by Journé et al. (2022).

1370 A.2. SoftWTA Weight Initialisation

1371 The weight initialisation scheme used for the SoftWTA Hebbian (**SoftWT**
1372 **A**) configuration is detailed below, mirroring the Normal Distribution used in
1373 (Journé et al., 2022). To allow all neurons to learn and reach convergence in a
1374 single epoch, a positive distribution with a radius larger than 2.5 is necessary:

$$\text{weight_range} = \frac{25}{\sqrt{C_{in} \cdot K_h \cdot K_w}}$$

$$W = \text{weight_range} \cdot X$$

$$X \sim \mathcal{N}(0, 1)$$

1375 Appendix B. Centre-Surround Filters through Dale's Principle

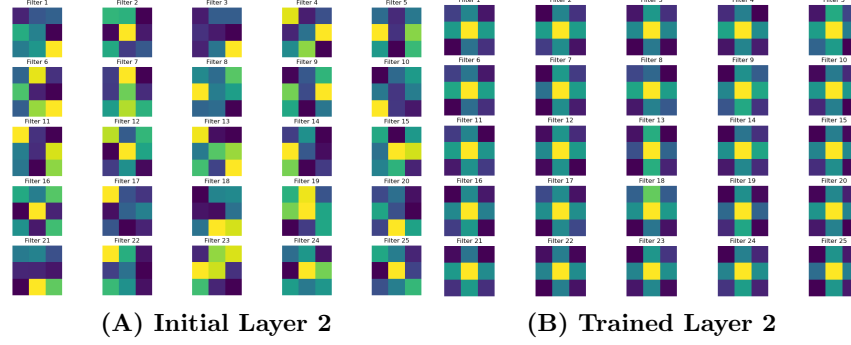


Figure B.15: Direct weight visualisation of filters in Layer 2 of **Dale-Depthwise-Surr/HardWTA-BCM** before and after training. Note the exclusive formation of centre-surround filters.

1376 The depthwise network (**Dale_Depthwise-Surr/HardWTA/Cos-BCM**) which
 1377 fully respected Dale's principle exhibited exclusively centre-surround patterns
 1378 in its depthwise filters at Layers 2-3 (seen through the direct filter visualisa-
 1379 tion in Figure B.15), further accentuating the biological realism of combining
 1380 the depthwise architecture with Hebbian Learning and its biologically-inspired
 1381 competition mechanisms.

1382 Appendix C. Network Architectures

Table C.7: Journe CNN Architecture (Non-Depthwise)

Layer	Type	Output Shape	Kernel	Stride	Padding	Activation
Input	—	(3, 32, 32)	—	—	—	—
1	BatchNorm2d	(3, 32, 32)	—	—	—	—
	HebbianConv2d	(96, 32, 32)	5x5	1	2	—
	Triangle	(96, 32, 32)	—	—	—	power=0.7
	MaxPool2d	(96, 16, 16)	4x4	2	1	—
2	BatchNorm2d	(96, 16, 16)	—	—	—	—
	HebbianConv2d	(384, 16, 16)	3x3	1	1	—
	Triangle	(384, 16, 16)	—	—	—	power=1.4
	MaxPool2d	(384, 8, 8)	4x4	2	1	—
3	BatchNorm2d	(384, 8, 8)	—	—	—	—
	HebbianConv2d	(1536, 8, 8)	3x3	1	1	—
	Triangle	(1536, 8, 8)	—	—	—	power=1.0
	AvgPool2d	(1536, 4, 4)	2x2	2	0	—
Output	Linear	(10)	—	—	—	—

Table C.8: Lagani 3-layer CNN Architecture (Non-Depthwise)

Layer	Type	Output Shape	Kernel	Stride	Padding	Activation
Input	–	(3, 32, 32)	–	–	–	–
1	BatchNorm2d	(3, 32, 32)	–	–	–	–
	HebbianConv2d	(96, 28, 28)	5x5	1	0	Cosine
	Triangle	(96, 28, 28)	–	–	–	power=1.0
	MaxPool2d	(96, 14, 14)	2x2	2	0	–
2	BatchNorm2d	(96, 14, 14)	–	–	–	–
	HebbianConv2d	(128, 12, 12)	3x3	1	0	Cosine
	Triangle	(128, 12, 12)	–	–	–	power=1.0
3	BatchNorm2d	(128, 12, 12)	–	–	–	–
	HebbianConv2d	(192, 10, 10)	3x3	1	0	Cosine
	Triangle	(192, 10, 10)	–	–	–	power=1.0
	AvgPool2d	(192, 5, 5)	2x2	2	0	–
Output	Linear	(10)	–	–	–	–

Table C.9: Lagani 4-layer CNN Architecture (Non-Depthwise)

Layer	Type	Output Shape	Kernel	Stride	Padding	Activation
Input	–	(3, 32, 32)	–	–	–	–
1	BatchNorm2d	(3, 32, 32)	–	–	–	–
	HebbianConv2d	(96, 28, 28)	5x5	1	0	Cosine
	Triangle	(96, 28, 28)	–	–	–	power=1.0
	MaxPool2d	(96, 14, 14)	2x2	2	0	–
2	BatchNorm2d	(96, 14, 14)	–	–	–	–
	HebbianConv2d	(128, 12, 12)	3x3	1	0	Cosine
	Triangle	(128, 12, 12)	–	–	–	power=1.0
3	BatchNorm2d	(128, 12, 12)	–	–	–	–
	HebbianConv2d	(192, 10, 10)	3x3	1	0	Cosine
	Triangle	(192, 10, 10)	–	–	–	power=1.0
	AvgPool2d	(192, 5, 5)	2x2	2	0	–
4	BatchNorm2d	(192, 5, 5)	–	–	–	–
	HebbianConv2d	(256, 3, 3)	3x3	1	0	Cosine
	Triangle	(256, 3, 3)	–	–	–	power=1.0
Output	Linear	(10)	–	–	–	–

Table C.10: Journe 4-layer CNN Architecture (Non-Depthwise)

Layer	Type	Output Shape	Kernel	Stride	Padding	Activation
Input	–	(3, 32, 32)	–	–	–	–
1	BatchNorm2d	(3, 32, 32)	–	–	–	–
	HebbianConv2d	(96, 32, 32)	5x5	1	2	–
	Triangle	(96, 32, 32)	–	–	–	power=0.7
	MaxPool2d	(96, 16, 16)	4x4	2	1	–
2	BatchNorm2d	(96, 16, 16)	–	–	–	–
	HebbianConv2d	(384, 16, 16)	3x3	1	1	–
	Triangle	(384, 16, 16)	–	–	–	power=1.4
	MaxPool2d	(384, 8, 8)	4x4	2	1	–
3	BatchNorm2d	(384, 8, 8)	–	–	–	–
	HebbianConv2d	(1536, 8, 8)	3x3	1	1	–
	Triangle	(1536, 8, 8)	–	–	–	power=1.0
	AvgPool2d	(1536, 4, 4)	2x2	2	0	–
4	BatchNorm2d	(1536, 4, 4)	–	–	–	–
	HebbianConv2d	(6144, 4, 4)	3x3	1	1	–
	Triangle	(6144, 4, 4)	–	–	–	power=1.0
	AvgPool2d	(6144, 2, 2)	2x2	2	0	–
Output	Linear	(10)	–	–	–	–

Table C.11: Journe CNN Architecture (Depthwise)

Layer	Type	Output Shape	Kernel	Stride	Padding	Activation
Input	–	(3, 32, 32)	–	–	–	–
1	BatchNorm2d	(3, 32, 32)	–	–	–	–
	HebbianConv2d	(96, 32, 32)	5x5	1	2	–
	Triangle	(96, 32, 32)	–	–	–	power=0.7
	MaxPool2d	(96, 16, 16)	4x4	2	1	–
2	BatchNorm2d	(96, 16, 16)	–	–	–	–
	HebbianDepthConv2d	(96, 16, 16)	3x3	1	1	–
	BatchNorm2d	(96, 16, 16)	–	–	–	–
	HebbianConv2d	(384, 16, 16)	1x1	1	0	–
	Triangle	(384, 16, 16)	–	–	–	power=1.4
	MaxPool2d	(384, 8, 8)	4x4	2	1	–
3	BatchNorm2d	(384, 8, 8)	–	–	–	–
	HebbianDepthConv2d	(384, 8, 8)	3x3	1	1	–
	BatchNorm2d	(384, 8, 8)	–	–	–	–
	HebbianConv2d	(1536, 8, 8)	1x1	1	0	–
	Triangle	(1536, 8, 8)	–	–	–	power=1.0
	AvgPool2d	(1536, 4, 4)	2x2	2	0	–
Output	Linear	(10)	–	–	–	–

Table C.12: Residual CNN Architecture

Layer	Type	Output Shape	Kernel	Stride	Padding	Activation
Input	–	(3, 32, 32)	–	–	–	–
1	BatchNorm2d	(3, 32, 32)	–	–	–	–
	HebbianConv2d	(96, 32, 32)	5x5	1	2	Cosine
	Triangle	(96, 32, 32)	–	–	–	power=0.7
	MaxPool2d	(96, 16, 16)	4x4	2	1	–
2	HebbianResidualBlock	(384, 16, 16)	–	–	–	power=1.4
	MaxPool2d	(384, 8, 8)	4x4	2	1	–
3	HebbianResidualBlock	(1536, 8, 8)	–	–	–	power=1.0
	AvgPool2d	(1536, 4, 4)	2x2	2	0	–
Output	Linear	(10)	–	–	–	–
HebbianResidualBlock Internal Structure						
Main Path	BatchNorm2d	(in_ch, H, W)	–	–	–	–
	HebbianConv2d	(hidden_dim, H, W)	1x1	1	0	Cosine
	Triangle	(hidden_dim, H, W)	–	–	–	power=act
	BatchNorm2d	(hidden_dim, H, W)	–	–	–	–
	HebbianDepthConv2d	(hidden_dim, H, W)	3x3	1	1	Cosine
	Triangle	(hidden_dim, H, W)	–	–	–	power=act
	BatchNorm2d	(hidden_dim, H, W)	–	–	–	–
	HebbianConv2d	(out_ch, H, W)	1x1	1	0	Cosine
Shortcut	BatchNorm2d*	(in_ch, H, W)	–	–	–	–
	HebbianConv2d*	(out_ch, H, W)	1x1	1	0	Cosine
	Add	(out_ch, H, W)	–	–	–	–
	Triangle	(out_ch, H, W)	–	–	–	power=act

* Only if in_channels $\neq$ out_channels