

# Advancing the Biological Plausibility and Efficacy of Hebbian Convolutional Neural Networks

Julian Jiménez Nimmo<sup>a,1</sup>, Esther Mondragón<sup>a,\*</sup>

<sup>a</sup>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. The resulting model achieved 76% classification accuracy on CIFAR-10, rivalling its end-to-end backpropagation variant (77%) and critically surpassing the state-of-the-art hard-WTA performance in CNNs of the same network depth (64.6%) by 11.4%. 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

---

\*Corresponding author

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

<sup>1</sup>First author.

learning algorithm - error backpropagation (Rumelhart et al., 1986; Werbos, 1994) - presents significant challenges for practical deployment and biological plausibility (Zarkeshian et al., 2022; Lillicrap et al., 2020). Backpropagation is a supervised learning process that relies on propagating error gradients backward through the network's layers to adjust connection weights so that the network outputs come close to some given target output. This global update process requires storing all intermediate activations and computing gradients across the entire network (Rumelhart et al., 1986).

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

Conversely, Hebbian learning, a biologically inspired learning algorithm, circumvents the need for error backpropagation, updating weights locally, using unsupervised local neural activity correlations. Although Hebbian learning shows promise in addressing these challenges, posing as an alternative that mirrors the efficiency and adaptability of biological systems (Gerstner and Kistler, 2002), it is still needed to achieve similar high levels of accuracy as those reported with backpropagation training (Demidovskij et al., 2023).

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

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

representations.

Arguably, sparse neural coding derived from Hard-WTA has greater biological 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 the SoftHebb architecture. Second, we furthered biological realism through Bienenstock, Cooper, and Munro (BCM) learning rule (Bienenstock et al., 1982) (75%), 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 76% accuracy on CIFAR-10 in the SoftHebb 3-CNN layer architecture, setting a new SOTA Hard-WTA performance. This represents a 11.4% 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 77% accuracy under the same training conditions (20 epochs and same architecture).

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 Projection (UMAP) for feature embeddings, receptive fields and weight distribu-

tions of neurons for the analysis of competition and feature learning.

## 2. Background and Related Work

### 2.1. Convolutional Neural Networks

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

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

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

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

Residual blocks allow training for very deep backpropagation networks by providing skip connections between layers (He et al., 2016) mimicking the hierarchical feedback pathways found in biological visual systems. DSCs separate a typical convolution operation into two separate convolutional operations (Chollet, 2017) for computational efficiency and preventing overfitting through parameter reduction. These features can also be used alongside Hebbian learning. Hence, following the rationale to improve biological plausibility and efficiency, 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; Bredenborg 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 applications. Hebbian learning (Hebb, 1949; Lagani et al., 2023), follows the principle

*neurons that fire together, wire together*<sup>2</sup>. Neural updates are reliant solely on the activity of locally connected neurons. These neurons undergo a completely unsupervised learning process, as they do not require input from an external teacher or an error signal, and can identify statistical patterns in input data. The local nature of Hebbian learning eliminates the need for backward passes and target signals through the network, reducing memory requirements and enabling parallel weight updates. This locality makes Hebbian learning particularly attractive for neuromorphic hardware implementation (Schuman et al., 2022), edge computing applications, and scenarios with limited labelled data.

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

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

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

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

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

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

The equation is presented in two mathematically equivalent forms to emphasise different interpretations of the learning process. The first expression,  $\eta y(x - w)$ , highlights the biological intuition: the weight change is proportional to the difference between the input signal  $x$  and the current weight  $w$ , gated by the post-synaptic activity  $y$ . The second expression,  $\eta(yx - yw)$ , reveals the rule's dual mechanism: a Hebbian term  $yx$  for weight growth and an

---

<sup>2</sup>Originally 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)"

anti-Hebbian 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:

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

The equation is presented in two equivalent forms to highlight different aspects of the rule. The first expression uses the nonlinear function  $\psi$ , 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  $\theta$ , creating a natural stability mechanism in the learning process.

While Oja’s rule (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:

$$\Delta w(t) = \eta y_i(x - \sum y_j w_j) \quad (4)$$

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) actively decorrelates neural activities by weakening synaptic strengths between co-activated neurons, supporting the WTA-driven specialization 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, 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) 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

anti-Hebbian 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  $x_i$ ,  $y_k$  is the postsynaptic output of a neuron  $k$ , and  $u_k$  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 the Oja 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 deep learning systems, allowing networks to learn from limited data, exhibit greater noise robustness, and consume fewer computational resources. This approach bridges biological and artificial systems, potentially enabling more resilient and flexible learning. In this paper, we first searched for a suitable

combination of these mechanisms, exploring their accuracy and potential to enhance the formation of hierarchical representations, incorporating hierarchical layers of increasingly abstract, sparse representations shaped by competition. The we selected the 4 most suited configurations based on their accuracy for further evaluation and analysis.

### 3. Methodology

This section outlines the PyTorch-based Hebbian-CNN framework adopted. We designed and implemented 15 distinct configurations to systematically evaluate their components and interactions and build a naturalistic optimal architecture based on their performance as measured by accuracy values. To this aim, we developed a configurable architecture using different learning algorithms, Hebbian Grossberg Instar and BCM learning, two main neural competition mechanisms, hard and soft WTA, and additional temporal, homeostatic, pre-synaptic and spatial competition mechanisms. These mechanisms were implemented as optional components in a customisable Hebbian layer. Each element could be enabled, disabled, or combined into distinct configurations to allow us to explore their performance. Architectural changes, including depthwise separable convolutions and its enhancement through skip connections in residual blocks were evaluated.

The design philosophy behind this work encompasses computational efficiency, code readability, modular architecture, parametric flexibility, and extensibility. We prioritised a GPU-accelerated implementation leveraging the PyTorch framework, known for its efficient parallel processing capabilities in deep learning applications. The modular structure closely mirrors that of the PyTorch, providing an intuitive organisational system familiar to researchers experienced with this machine learning library. The code is available on GitHub: [Link](#)

#### 3.1. Learning Pipeline

A two-phase approach was used to ensure feature learning biological plausibility while having a standard classifier system. Unsupervised Hebbian learning was employed for feature extraction in convolutional layers, followed by supervised backpropagation linear classifier training on the frozen features.

The framework implemented a consistent, unsupervised feature learning pipeline across all Hebbian configurations, enabling feature extraction without requiring labelled data or global error signals. Figure 1 illustrates the Hebbian 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 selection uses synaptic traces for consistent activation
  - Homeostatic selection adjusts based on input statistics
5. Weights are updated using the chosen learning rule (Grossberg, BCM, or SoftHebb)
6. Weight normalisation is applied to maintain stable learning dynamics

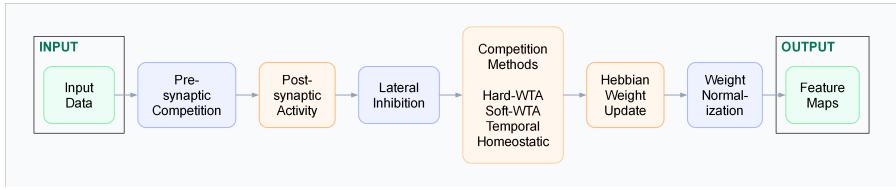


Figure 1: Diagram of the Hebbian learning process in our custom Hebbian-CNN layer. Green boxes correspond to input/output, blue boxes indicate optional processing, and orange boxes designate core learning and competition mechanisms.

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) was chosen as the benchmark dataset. CIFAR-10 consists of 60,000 RGB images of 32x32 pixels across 10 classes. Data processing included implementing random horizontal flip augmentations and zero-phase components analysis (ZCA) whitening. ZCA whitening was computed as improvements in performance were found for Hard-WTA competition.

### 3.3. Core Learning and Competition Mechanisms

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

#### 3.3.1. Grossberg Hebbian Learning Implementation

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

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

where  $*$  denotes the convolution operation between weights  $w$  and pre-synaptic activities  $x$ . The Grossberg weight update rule was then calculated as:

$$\Delta w = y * x - y \odot w \quad (7)$$

where the  $y * x$  term was computed via convolution, and the  $y * w$  term is calculated via scalar multiplication as a single value per output channel representing total activation across the batch and spatial locations. These updates were normalised and stored in a buffer for efficiency and 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:

$$y_{final} = y \odot \text{mask}_{WTA}(y) \quad (8)$$

Following Lagani's approach (Lagani et al., 2022a), we refined the selectivity by applying a cosine similarity activation function to these WTA results:

$$\text{sim}(y, w) = \frac{\langle y, w \rangle}{\|y\|_2 \cdot \|w\|_2} \quad (9)$$

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.

### 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 anti-Hebbian learning for all neurons except the maximally activated one, creating a more nuanced form of competition than the binary selection of Hard-WTA.

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

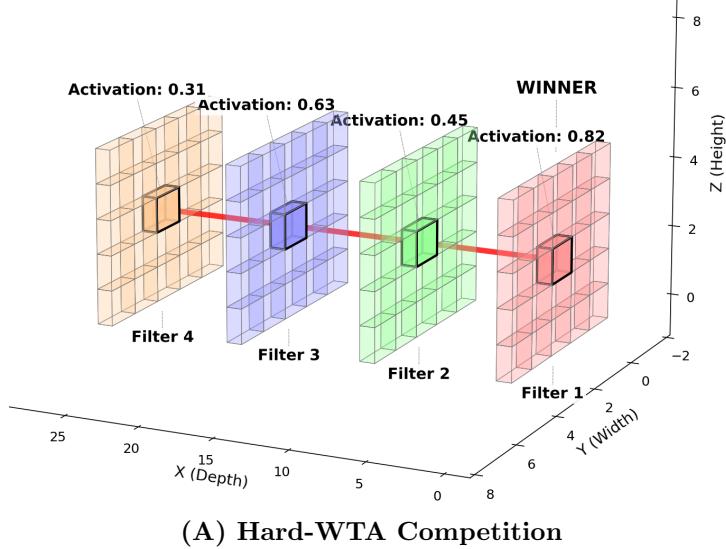


Figure 2: Implementation of Hard-WTA competition in CNNs influences neurons situated at the same spatial position across various filters.

### 3.4. BCM Learning Rule

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

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

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

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

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

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

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

### 3.5. Temporal and Homeostatic Competition

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

We developed two selection mechanisms following Lagani's proposals. Our Temporal Selection implementation tracked neuronal activity through an activation history buffer  $H_t$  of size  $T$  for each neuron:

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

From this buffer (with default size of 500), we computed a temporal threshold  $\theta_t$  as the median of historical activations:

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

This threshold-based approach promoted neurons demonstrating consistent activity over time, as only neurons whose activation  $y$  exceeded  $\theta_t$  are permitted to update their weights.

For Homeostatic Selection, we applied an input-driven competition mechanism that calculates normalised similarity scores between weights and input:

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

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

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

where  $\mu_S$  and  $\sigma_S$  are the mean and standard deviation of similarities respectively, and  $k$  is the competition factor (typically set to 2). This adaptive threshold dynamically adjusted neural plasticity, increasing when input patterns produced high similarity scores to prevent overactivation, and decreasing with low scores to facilitate more neural updates. Only neurons with post-synaptic activities  $y$  exceeding  $\theta_{adaptive}$  updated their weights.

Both selection mechanisms were designed to be compatible with additional competition methods like Hard-WTA to further promote efficient specialisation and sparse connectivity.

### 3.6. Pre-synaptic Competition

Pre-synaptic competition was enforced using synaptic couplings, modifying the weights  $w$  before calculating the postsynaptic activity. Three competition modes were developed across the input channel, where different input features compete to influence the same output. Each mode first computed an inverse weight magnitude ( $m = \frac{1}{|w|+\epsilon}$ ) to promote competition between strong synapses, where  $\epsilon$  is a small constant (1e-6) to prevent division by zero:

1. **Linear Competition, defined as:**

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

It normalises the weights to values between 0-1.

2. **Softmax Competition, defined as:**

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

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

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

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

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

### 3.7. Lateral Inhibition

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

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

where  $G_e$  and  $G_i$  represent the excitatory and inhibitory Gaussian functions, respectively, and  $\sigma_e$  and  $\sigma_i$  denote the standard deviations of the excitatory and inhibitory Gaussians (with default value of 1.2 and 1.4). This kernel strengthens synapses in the immediate spatial neighbourhood of a neuron, and weakens synapses further from this neighbourhood. When applied, it created a kernel of size 5 in all layers.

### 3.8. Architectures

To investigate biologically-plausible learning mechanisms while maintaining computational efficiency, we explored three distinct architectural variants, specifically SoftHebb along with its counterparts, the depthwise separable convolution, and the residual block. Each variant built upon SoftHebb baseline architecture while introducing specific modifications aimed at reducing parameters, improving performance, and enhancing biological realism. These architectures were evaluated independently to assess their individual contributions to the model's capabilities.

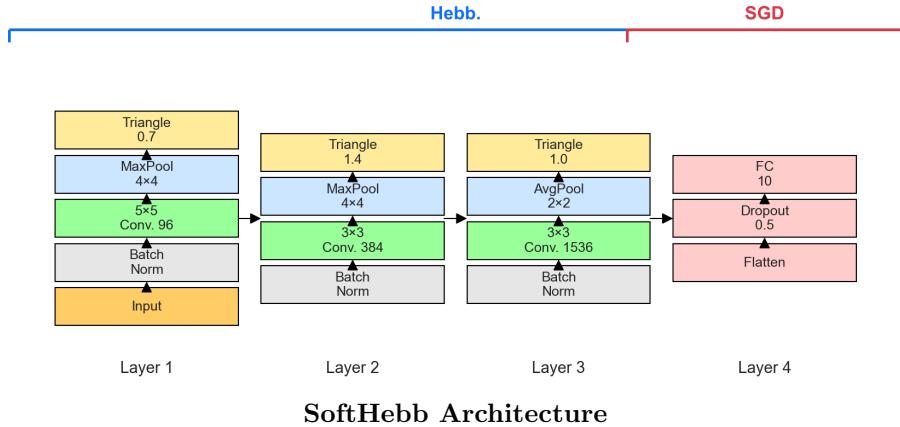


Figure 3: Visual representation of the main SoftHebb Architecture, composed of 3 Hebbian CNN layers and a final backpropagation linear classifier layer.

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

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 SoftHebb 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 SoftHebb 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 SoftHebb 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, 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  $\eta$  represents the learning rate and  $\lambda$  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.

## 4. Experimental Settings

### 4.1. Experimental Configurations

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

The configurations were built upon two fundamental architectural approaches established in previous research: the SoftHebb architecture from (Journé et al., 2022), and the Lagani architecture from (Lagani et al., 2022a). The **SoftHebb-SoftWTA-Instar** and **Lagani\HardWTA/Cos-Instar/Lagani\Deep-HardWTA/Cos-Instar** configurations directly replicate these approaches, serving as benchmarks for our enhanced implementations.

Building upon these foundations, we developed a series of novel configurations that integrate additional biologically-sound mechanisms or architectural improvements. Our configurations explored combinations of Hard-WTA competition with presynaptic competition (**SoftHebb-Pre/HardWTA/Cos-Instar**), lateral inhibition (**SoftHebb-Surr/HardWTA/Cos-Instar**), homeostatic (**SoftHebb-Hom/HardWTA/Cos-Instar**) and temporal (**SoftHebb-Temp/HardWTA/Cos-Instar**) competition. **SoftHebb-None-Instar** evaluated the system performance with no competition mechanism.

We introduced architectural variants including depthwise separable convolutions (**Depthwise\\_SoftHebb-HardWTA/Cos-Instar**) and residual connections (**Residual\\_SoftHebb-HardWTA/Cos-Instar**) to investigate more efficient and biologically-inspired network structures. Either the Grossberg Instar or BCM rule was employed to adjust the convolutional weights.

The **Dale\\_Depthwise\\_SoftHebb-Surr/HardWTA/Cos-BCM** configuration represents our most biologically realistic implementation, uniquely combining three key elements: Dale’s Principle constraints on synaptic weights, depthwise separable convolutions, and BCM Hebbian learning with Hard-WTA competition and lateral inhibition.

To establish a direct comparison with traditional deep learning approaches, we included the **SoftHebb-Backpropation** 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.

Table 1: Hebbian Learning Initial Configurations and Features

Configuration	Architecture	Competition	Hebbian Rule
<i>Replications of Published SOTA models</i>			
SoftHebb-SoftWTA-Instar	SoftHebb	Soft WTA	GI
Lagani_HardWTA/Cos-Instar	Lagani	Hard WTA	GI
Lagani_Deep-HardWTA/Cos-Instar	Lagani	Hard WTA	GI
<i>Backpropagation Comparison</i>			
SoftHebb-Backpropagation	SoftHebb	N/A	BP
<i>Novel Implementations</i>			
SoftHebb-HardWTA/Cos-Instar	SoftHebb	Hard WTA	GI
SoftHebb-HardWTA/Cos-BCM	SoftHebb	Hard WTA	BCM
SoftHebb-Pre/HardWTA/Cos-Instar	SoftHebb	Hard WTA+Pre	GI
SoftHebb-Temp/HardWTA/Cos-Instar	SoftHebb	Hard WTA+Temp	GI
SoftHebb-Hom/HardWTA/Cos-Instar	SoftHebb	Hard WTA+Hom	GI
SoftHebb-Surr/HardWTA/Cos-Instar*	SoftHebb	Hard WTA+Surr	GI
Depthwise_SoftHebb-HardWTA/Cos-Instar†	SoftHebb	Hard WTA	GI
Residual_SoftHebb-HardWTA/Cos-Instar‡	SoftHebb	Hard WTA	GI
Dale_Depth_SoftHebb-Surr/HardWTA/Cos-BCM§†	SoftHebb	Hard WTA+Surr	BCM
SoftHebb-None-Instar	SoftHebb	None	GI

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

Bienenstock-Cooper-Munro, BP: Backpropagation, Pre: Presynaptic, Temp: Temporal, Hom: Homeostatic, Surr: Surround. Special features: \*Surround lateral inhibition, †Depthwise separable convolutions, ‡Residual connections, §Dale’s Principle.

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

Table 2: Purposes of Initial Hebbian Learning Configurations

Configuration	Purpose
<i>Replications of Published SOTA models</i>	
SoftHebb-SoftWTA-Instar	Evaluate SoftHebb research
Lagani_HardWTA/Cos-Instar	Evaluate Lagani Hard-WTA research
Lagani_Deep-HardWTA/Cos-Instar	Evaluate Lagani Hard-WTA 4-layer research
<i>Backpropagation Comparison</i>	
SoftHebb-Backpropagation	Evaluate pure backpropagation baseline
<i>Novel Implementations</i>	
SoftHebb-HardWTA/Cos-Instar	Evaluate Hard-WTA in network with padding
SoftHebb-HardWTA/Cos-BCM	Compare BCM to Grossberg-Instar rule
SoftHebb-Pre/HardWTA/Cos-Instar	Assess impact of presynaptic learning
SoftHebb-Temp/HardWTA/Cos-Instar	Evaluate temporal competition with/without WTA
SoftHebb-Hom/HardWTA/Cos-Instar	Evaluate statistical competition with Hard-WTA
SoftHebb-Surr/HardWTA/Cos-Instar	Assess effect of surround modulation kernel
Depthwise_SoftHebb-HardWTA/Cos-Instar	Evaluate depthwise equivalent of hard-WTA research
Residual_SoftHebb-HardWTA/Cos-Instar	Evaluate residual equivalent of hard-WTA research
Dale_Depth_SoftHebb-Surr/HardWTA/Cos-BCM	Evaluate improved biological realism in a CNN
SoftHebb-None-Instar	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.1) 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). 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 configurations were trained using the same predetermined random seed. 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.

## 5. Results

### 5.1. Classification Accuracy

Table 3: Optimal Test Performance Metric for all Configurations within 20 Epochs

Configuration	Accuracy (%)	F1 Score
<i>Replications of Published SOTA models</i>		
<b>SoftHebb-Soft WTA-Instar</b>	<b>79.2</b>	<b>0.79</b>
Lagani_HardWTA/Cos-Instar	59.7	0.59
Lagani_Deep-HardWTA/Cos-Instar	52.8	0.53
<i>Backpropagation Comparison</i>		
<b>SoftHebb-Backpropagation</b>	<b>77.7</b>	<b>0.77</b>
<i>Novel Implementations</i>		
SoftHebb-HardWTA/Cos-Instar	74.8	0.75
SoftHebb-HardWTA/Cos-BCM	75.3	0.75
SoftHebb-Presynaptic/HardWTA/Cos-Instar	65.1	0.65
SoftHebb-Temp/HardWTA/Cos-Instar	74.6	0.74
SoftHebb-Hom/HardWTA/Cos-Instar	68.5	0.69
SoftHebb-Surr/HardWTA/Cos-Instar	75.7	0.76
Depthwise_SoftHebb-HardWTA/Cos-Instar	71.9	0.72
Residual_SoftHebb-HardWTA/Cos-Instar	74.7	0.75
Dale_Depthwise_HardHebb-Surr/HardWTA/Cos-BCM	67.3	0.67
SoftHebb-None-Instar	19.9	0.600
<i>Optimal Hard-WTA Configuration</i>		
<b>SoftHebb-Optimal</b>	<b>76</b>	<b>0.76</b>

### 5.1.1. Replications of Published Research

Our initial configurations focused on reproducing previous research results. The **SoftHebb-SoftWTA-Instar** 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/Cos-Instar** configuration, and its deeper 4-CNN layer counterpart **Lagani\_Deep-HardWTA/Cos-Instar** (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.

### 5.1.2. Novel Implementations

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 **SoftHebb-HardWTA/Cos-Instar** 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 SoftHebb 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 **SoftHebb-Surr/HardWTA/Cos-Instar** configuration achieved 75.7% accuracy, further improving our new SOTA performance for Hard-WTA implementations. Alternative learning rules with BCM in **SoftHebb-HardWTA/Cos-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, **SoftHebb-Temp/HardWTA/Cos-Instar**, also contributed to highlight the robustness of our enhanced Hard-WTA framework. Other competition mechanisms showed varying degrees of effectiveness, with the **SoftHebb-Hom/HardWTA/Cos-Instar** and **SoftHebb-Presynaptic/HardWTA/Cos-Instar** configurations achieving 68.5% and 65.1% accuracy respectively.

Our architectural variants focused on efficiency and biological plausibility. The **Depthwise-SoftHebb-Surr/HardWTA/Cos-Instar**, which significantly reduced model parameters from 5.9M to 0.9M (a factor of 6.6), and the **Residual-SoftHebb-Surr/HardWTA/Cos-Instar** 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\SoftHebb-Surr/HardWTA/Cos-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 **SoftHebb-None-Instar** setup, which by lacking a

competition mechanism reached only 24% 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 SoftHebb architecture trained through end-to-end backpropagation (**SoftHebb-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. Optimal Setup: Hard-WTA Combined with Lateral Inhibition and BCM

The **SoftHebb-Optimal** 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 **SoftHebb-Optimal** configuration achieved the highest values: 76% accuracy with a F1-Score of 0.76.

### 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 the two published Hebbian SOTA models, **SoftHebb-SoftWTA-Instar** and **Lagani\Hard/Cos-Instar**, and the **SoftHebb-Backpropagation** against our highest performing Hard-WTA model, **SoftHebb-Optimal**.

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

#### 5.2.1. Accuracy Dynamics

Figure 4 displays accuracy per epoch during training and Figure 5 the test accuracy for the **SoftHebb-Optimal** (solid blue line), **SoftHebb-Backpropagation** (dashed orange line), **SoftHebb-SoftWTA-Instar** (dotted green line), and **Lagani-Hard/Cos-Instar** (red dash-dotted line).

Training and Test accuracy were remarkably higher and converged faster in our **SoftHebb-Optimal** than the previous Hard-WTA research **Lagani-Har**

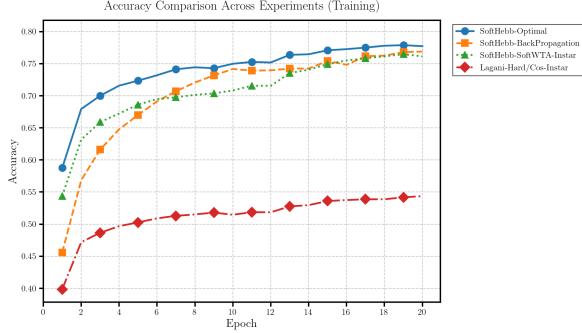


Figure 4: Accuracy values during training for Lagani-Hard/Cos-Instar (red), SoftHebb-SoftWTA-Instar (green), SoftHebb-Backpropagation (orange), and SoftHebb-Optimal (blue)

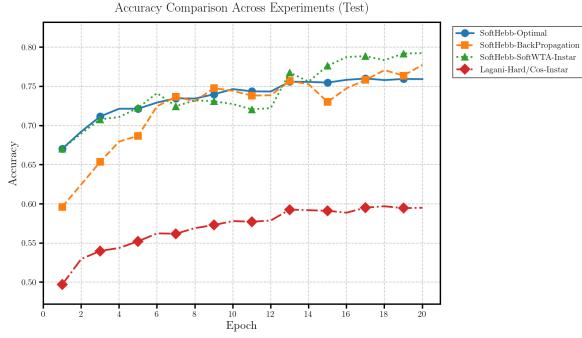


Figure 5: Accuracy values during test for Lagani-Hard/Cos-Instar (red), SoftHebb-SoftWTA-Instar (green), SoftHebb-Backpropagation (orange), and SoftHebb-Optimal (blue)

d/Cos-Instar. Moreover, the accuracy achieved by both **SoftHebb-Optimal** and **SoftHebb-SoftWTA-Instar** variants were comparable throughout training (Figure 4), with both Hebbian WTA approaches surpassing **SoftHebb-Backpropagation** at each epoch in both higher accuracy and convergence, with a significant higher initial accuracy for **SoftHebb-Optimal**. A similar pattern was observed during test (Figure 5), although the final accuracy level of the **SoftHebb-SoftWTA-Instar** was slightly higher than other configurations.

### 5.2.2. UMAP Class Clustering Embeddings

The UMAP embeddings at the final layer (Figures 6 and 7) revealed distinct patterns in feature organization. In these visualisations, dots of the same colour represent class labels, with Hebbian configurations showing clustering tendencies, where instances of the same class gather together and are separated from other classes.

The current Hebbian SOTA **SoftHebb-SoftWTA-Instar** (Figure 6 A) demonstrated highly effective class separation, with clearly delineated clusters

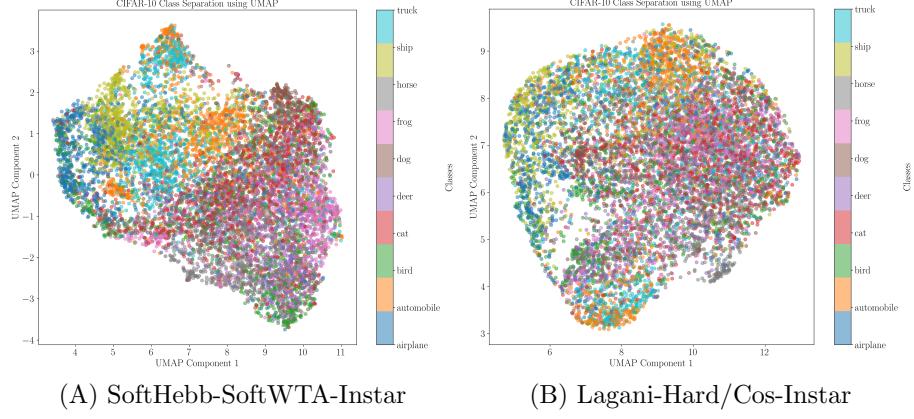


Figure 6: UMAP projection of class clusters from unsupervised feature extraction of the **SoftHebb-SoftWTA-Instar** and **Lagani-Hard/Cos-Instar** configurations.

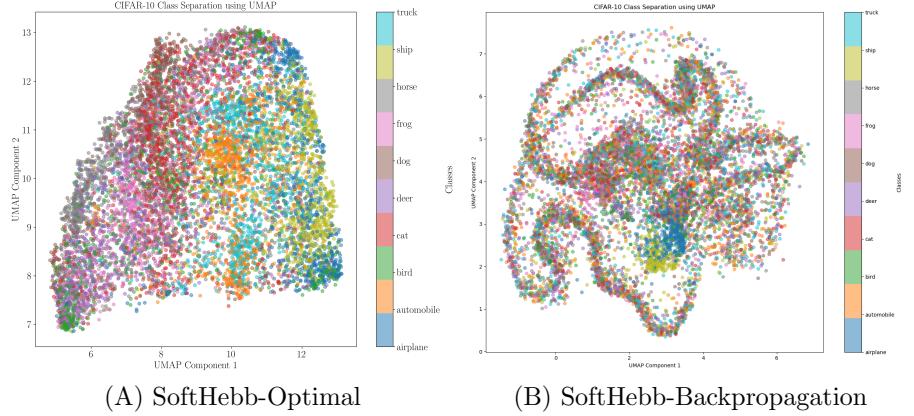


Figure 7: UMAP projection of class clusters from unsupervised feature extraction of the **SoftHebb-Optimal** and **SoftHebb-Backpropagation** configurations.

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-Hard/Cos-Instar (Figure 6 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, SoftHebb-Optimal (Figure 7 A), achieved clustering quality approaching that of SoftHebb-SoftWTA-Instar. The visual-

isation 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 SoftHebb-Backpropagation (Figure 7 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 cluster visually, rather than the discrete clustering preferred by competitive Hebbian learning. The findings suggest that Hebbian learning offers greater interpretability and explainability compared to backpropagation.

### 5.2.3. Weight Distributions

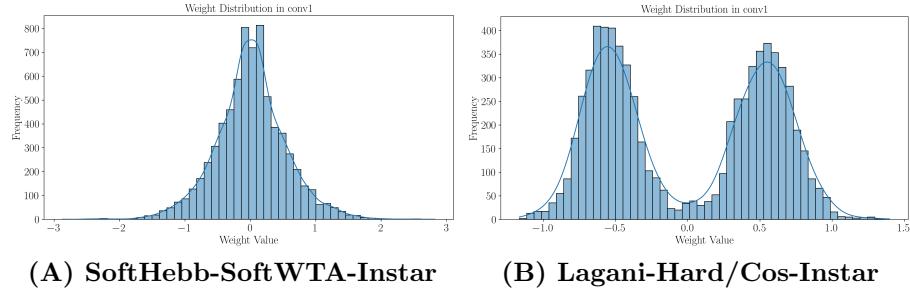


Figure 8: Weight distribution of **SoftHebb-SoftWTA-Instar** and **Lagani\_short-Hard/Cos-Instar** configurations at Layer 1.

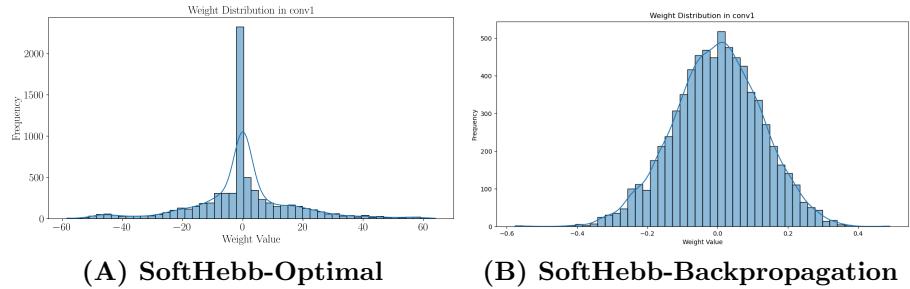


Figure 9: Weight distribution of **SoftHebb-Optimal** and **SoftHebb-Backpropagation** configurations at Layer 1.

Analysis of weight distributions (Figures 8 and 9) revealed distinct patterns

across learning approaches. **SoftHebb-SoftWTA-Instar** (Figure 8A) maintained a broad normal distribution across all layers, reflecting its soft competition approach. In contrast, **Lagani-Hard/Cos-Instar** (Figure 8B) showed a bimodal distribution with two distinct peaks, suggesting more binary weight patterns emerging from the hard competition mechanism.

Our **SoftHebb-Optimal**'s BCM learning rule (Figure 9A) produced a highly sparse distribution with a pronounced peak near zero and long tails, closely mirroring the sparse connectivity observed in biological neural networks (Hawkins et al., 2016). **SoftHebb-Backpropagation** (Figure 9B) displayed a classic normal distribution, characteristic of the global nature of gradient descent optimization.

#### 5.2.4. PGA Receptive Fields

**SoftHebb-SoftWTA-Instar** (Figure 10) demonstrated features that deviate from traditional Gabor-like patterns. Layer 1 exhibited diverse patterns including solid colour detectors (visible in bright orange/green squares), chequerboard patterns capturing local contrast and various colour-mixing effects. Layer 2 showed a transition to more complex combinations with fragmented patterns, smaller pixel clusters, and intricate colour interactions. Layer 3 developed highly abstract, sparse representations with noise-like features and scattered colour points, suggesting specialised detectors for complex distributed patterns.

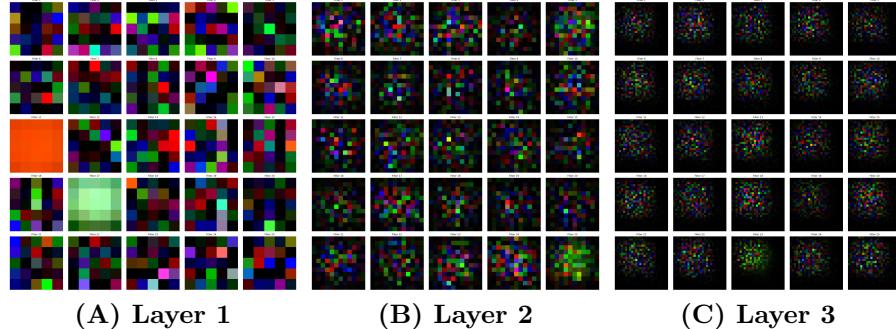


Figure 10: PGA Receptive fields for first 25 neurons at each layer of the **SoftHebb-SoftWTA-Instar** configuration.

**Lagani\_Hard/Cos-Instar** (Figure 11) exhibited classical biological neural network characteristics with Gabor-like edge detection. Layer 1 showed strong binary patterns with sharp edges and clear geometric shapes, where black-and-white regions and isolated pure colours (purple, blue, orange) demonstrate decisive edge detection from hard-WTA competition. Layer 2 maintained structured organisation with discernible orientation shapes and subtle colour blending, indicating effective combination of edge detectors with colour information. Layer 3 preserved defined shapes compared to SoftHebb's diffuse activations, reflecting maintained feature selectivity in deeper layers.

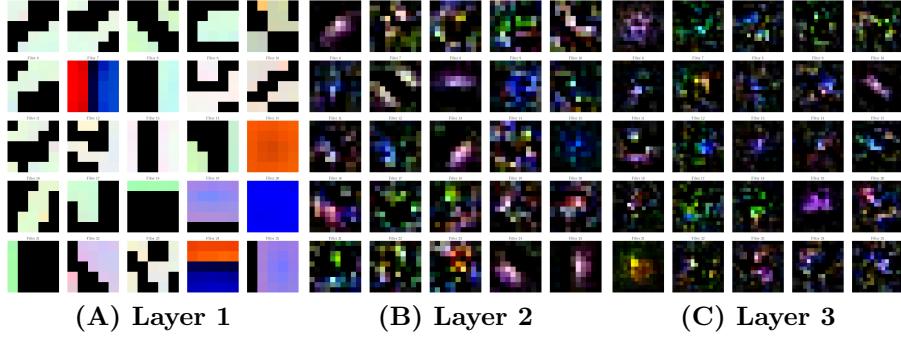


Figure 11: PGA Receptive fields for first 25 neurons at each layer of the **Lagani\_HardWTA/Cos-Instar** configuration.

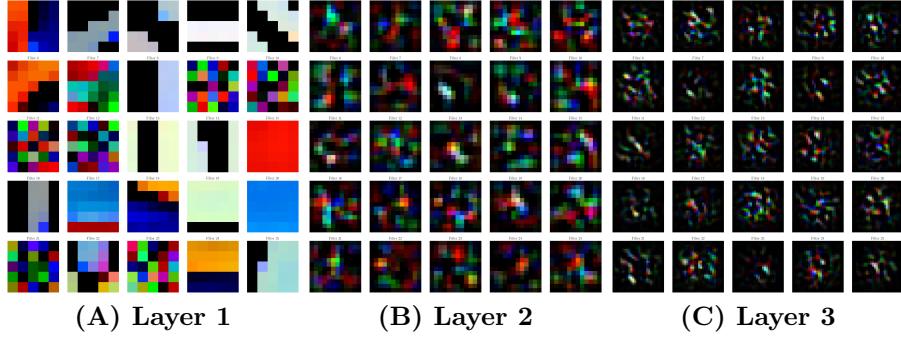


Figure 12: PGA Receptive fields for first 25 neurons at each layer of the **SoftHebb-Optimal** configuration.

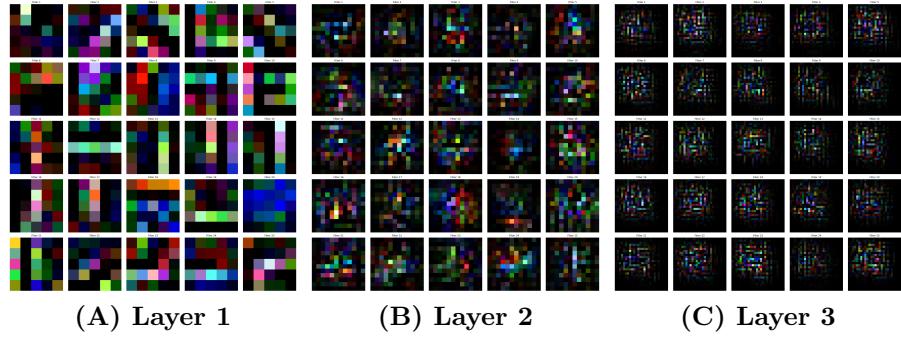


Figure 13: PGA Receptive fields for first 25 neurons at each layer of the **SoftHebb-backpropagation** configuration.

**SoftHebb-Optimal** (Figure 12) showcased successful integration of hard-WTA, lateral inhibition and BCM learning. Layer 1 combined sharp geometric boundaries comparable to Lagani's Gabor-like edge detection, features that

characterises biological neural networks, with more sophisticated colour processing, demonstrated through solid or multiple-colour colour detectors and complex edge patterns with smoother gradients and colour information. Layer 2 exhibited refined feature organisation with distinct colour combinations and organised spatial arrangements, where lateral inhibition promoted coherent transitions between spatial regions. Layer 3 developed specialised detectors with sparse but organised patterns maintaining structural coherence, suggesting effective preservation of feature selectivity while enabling complex pattern detection.

**SoftHebb-Backpropagation** (Figure 13) revealed characteristics of gradient-based optimisation. Layer 1 demonstrated strong directional selectivity with vertical and horizontal structures alongside systematic colour organisation, appearing more regularly than in Hebbian models. Layer 2 developed complex features while maintaining organisational principles, with uniform pattern and colour distribution suggesting global optimisation benefits. Layer 3 exhibited distinctive grid-like patterns with architectural regularity, reflecting systematic feature 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 32x32 pixel images and 3-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 CIFAR-10, and generalisation to other datasets, limited training data scenarios and adversarial attacks remains to be explored.

## 6. Conclusion

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 76% 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 (77%), albeit only in shallow networks, through careful integration of lateral inhibition and BCM learning rule.

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, enhanced

noise robustness, and reduced computational overhead. Our implementation demonstrates that enforcing strict competition through Hard-WTA, rather than the distributed activations of Soft-WTA, better captures the efficient coding principles observed in biological visual systems.

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 on GitHub: <https://github.com/J 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, 185–196.
- Amato, G., Carrara, F., Falchi, F., Gennaro, C., 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, Springer. pp. 324–334.

Apparaju, A., Arandjelović, O., 2022. Towards new generation, biologically plausible deep neural network learning. *Sci* 4, 46.

Babaiee, Z., Kiasari, P.M., Rus, D., Grosu, R., 2024. Neural echos: Depthwise convolutional filters replicate biological receptive fields, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 8216–8225.

Barth, A.L., Poulet, J.F., 2012. Experimental evidence for sparse firing in the neocortex. *Trends in neurosciences* 35, 345–355.

Bienenstock, E.L., Cooper, L.N., Munro, P.W., 1982. Theory for the development of neuron selectivity: orientation specificity and binocular interaction in visual cortex. *Journal of Neuroscience* 2, 32–48.

Blakemore, C., Van Sluyters, R.C., Movshon, J.A., 1976. Synaptic competition in the kitten's visual cortex, in: Cold Spring Harbor Symposia on Quantitative Biology, Cold Spring Harbor Laboratory Press. pp. 601–609.

Bredenberg, C., Williams, E., Savin, C., Richards, B., Lajoie, G., 2023. Formalizing locality for normative synaptic plasticity models, in: Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., Levine, S. (Eds.), Advances in Neural Information Processing Systems, Curran Associates, Inc.. pp. 5653–5684. URL: [https://proceedings.neurips.cc/paper\\_files/paper/2023/file/120339238f293d4ae53a7167403abc4b-Paper-Conference.pdf](https://proceedings.neurips.cc/paper_files/paper/2023/file/120339238f293d4ae53a7167403abc4b-Paper-Conference.pdf).

Choe, Y., 2022. Anti-hebbian learning. *Encyclopedia of Computational Neuroscience* , 213–216.

Chollet, F., 2017. Xception: Deep learning with depthwise separable convolutions, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1251–1258.

Cornford, J., Kalajdzievski, D., Leite, M., Lamarquette, A., Kullmann, D.M., Richards, B., 2020. Learning to live with dale's principle: Anns with separate excitatory and inhibitory units. *bioRxiv* , 2020–11.

Demidovskij, A., Kazyulina, M., Salnikov, I., Tugaryov, A., Trutnev, A., Pavlov, S., 2023. Implementation challenges and strategies for hebbian learning in convolutional neural networks. *Optical Memory and Neural Networks* 32, S252–S264.

Eccles, J.C., 1976. From electrical to chemical transmission in the central nervous system: the closing address of the sir henry dale centennial symposium cambridge, 19 september 1975. *Notes and records of the Royal Society of London* 30, 219–230.

- Fukushima, K., 1980. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics* 36, 193–202.
- Gabbott, P., Somogyi, P., 1986. Quantitative distribution of gaba-immunoreactive neurons in the visual cortex (area 17) of the cat. *Experimental brain research* 61, 323–331.
- Gerstner, W., Kistler, W.M., 2002. Mathematical formulations of hebbian learning. *Biological cybernetics* 87, 404–415.
- Grinberg, L., Hopfield, J., Krotov, D., 2019. Local unsupervised learning for image analysis, in: Real Neurons & Hidden Units: Future directions at the intersection of neuroscience and artificial intelligence@ NeurIPS 2019.
- Grossberg, S., 1976. Adaptive pattern classification and universal recoding: I. parallel development and coding of neural feature detectors. *Biological cybernetics* 23, 121–134.
- Gupta, M., Modi, S.K., Zhang, H., Lee, J.H., Lim, J.H., 2022. Is bio-inspired learning better than backprop? benchmarking bio learning vs. backprop. arXiv preprint arXiv:2212.04614 .
- Hasani, H., Soleymani, M., Aghajan, H., 2019. Surround modulation: A bio-inspired connectivity structure for convolutional neural networks. *Advances in neural information processing systems* 32.
- Hawkins, J., Ahmad, S., Purdy, S., Lavin, A., 2016. Biological and machine intelligence (bami). Initial online release 0.4 .
- He, K., Zhang, X., Ren, S., Sun, J., 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, in: Proceedings of the IEEE international conference on computer vision, pp. 1026–1034.
- He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770–778.
- Hebb, D.O., 1949. The organization of behavior: A neuropsychological theory. Psychology press.
- Hinton, G., 2022. The forward-forward algorithm: Some preliminary investigations. arXiv preprint arXiv:2212.13345 .
- Hopfield, J.J., 1982. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences* 79, 2554–2558.
- Journé, A., Rodriguez, H.G., Guo, Q., Moraitis, T., 2022. Hebbian deep learning without feedback, in: The Eleventh International Conference on Learning Representations.

- Krithivasan, S., Sen, S., Venkataramani, S., Raghunathan, A., 2022. Accelerating dnn training through selective localized learning. *Frontiers in Neuroscience* 15, 759807.
- Krizhevsky, A., Hinton, G., et al., 2009. Learning multiple layers of features from tiny images .
- Krotov, D., 2023. A new frontier for hopfield networks. *Nature Reviews Physics* 5, 366–367.
- Krotov, D., Hopfield, J.J., 2019. Unsupervised learning by competing hidden units. *Proceedings of the National Academy of Sciences* 116, 7723–7731.
- Lagani, G., 2022. Recent advancements on bio-inspired hebbian learning for deep neural networks., in: SEBD, pp. 610–615.
- Lagani, G., 2024. Bio-inspired deep learning research directions. Google Docs Document found in: <https://docs.google.com/document/d/1mRNCOAEkrW7tcztbNuXdoz428ZCwbBW8kA2FZKZBh2o/edit>.
- Lagani, G., Falchi, F., Gennaro, C., Amato, G., 2022a. Comparing the performance of hebbian against backpropagation learning using convolutional neural networks. *Neural Computing and Applications* 34, 6503–6519.
- Lagani, G., Falchi, F., Gennaro, C., Amato, G., 2023. Synaptic plasticity models and bio-inspired unsupervised deep learning: A survey. arXiv preprint arXiv:2307.16236 .
- Lagani, G., Gennaro, C., Fassold, H., Amato, G., 2022b. Fasthebb: Scaling hebbian training of deep neural networks to imagenet level, in: International Conference on Similarity Search and Applications, Springer. pp. 251–264.
- LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86, 2278–2324.
- Lee, K., Lee, K., Lee, H., Shin, J., 2018. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *Advances in neural information processing systems* 31.
- Lillicrap, T.P., Santoro, A., Marris, L., Akerman, C.J., Hinton, G., 2020. Backpropagation and the brain. *Nature Reviews Neuroscience* 21, 335–346.
- Miconi, T., 2017. Biologically plausible learning in recurrent neural networks reproduces neural dynamics observed during cognitive tasks. *Elife* 6, e20899.
- Miconi, T., 2021. Hebbian learning with gradients: Hebbian convolutional neural networks with modern deep learning frameworks. arXiv preprint arXiv:2107.01729 .

- Morris, R., 2006. Elements of a neurobiological theory of hippocampal function: the role of synaptic plasticity, synaptic tagging and schemas. *European Journal of Neuroscience* 23, 2829–2846.
- Nowlan, S., 1989. Maximum likelihood competitive learning. *Advances in neural information processing systems* 2.
- Oja, E., 1982. Simplified neuron model as a principal component analyzer. *Journal of mathematical biology* 15, 267–273.
- Rasmussen, C.E., Willshaw, D.J., 1993. Presynaptic and postsynaptic competition in models for the development of neuromuscular connections. *Biological cybernetics* 68, 409–419.
- Rojas, R., Rojas, R., 1996. The backpropagation algorithm. *Neural networks: a systematic introduction* , 149–182.
- Rumelhart, D.E., Hinton, G.E., Williams, R.J., 1986. Learning representations by back-propagating errors. *nature* 323, 533–536.
- Rumelhart, D.E., Zipser, D., 1985. Feature discovery by competitive learning. *Cognitive science* 9, 75–112.
- Schuman, C.D., Kulkarni, S.R., Parsa, M., Mitchell, J.P., Kay, B., et al., 2022. Opportunities for neuromorphic computing algorithms and applications. *Nature Computational Science* 2, 10–19.
- Sexton, R.S., Gupta, J.N., 2000. Comparative evaluation of genetic algorithm and backpropagation for training neural networks. *Information sciences* 129, 45–59.
- Song, Y., Lukasiewicz, T., Xu, Z., Bogacz, R., 2020. Can the brain do backpropagation?—exact implementation of backpropagation in predictive coding networks. *Advances in neural information processing systems* 33, 22566–22579.
- Strubell, E., Ganesh, A., McCallum, A., 2020. Energy and policy considerations for modern deep learning research, in: *Proceedings of the AAAI conference on artificial intelligence*, pp. 13693–13696.
- Tomen, N., Pintea, S.L., Van Gemert, J., 2021. Deep continuous networks, in: *International Conference on Machine Learning*, PMLR. pp. 10324–10335.
- Turrigiano, G.G., Nelson, S.B., 2004. Homeostatic plasticity in the developing nervous system. *Nature reviews neuroscience* 5, 97–107.
- Wang, Q., Li, Y., Li, R., 2024. Ecological footprints, carbon emissions, and energy transitions: the impact of artificial intelligence (ai). *Humanities and Social Sciences Communications* 11, 1–18.

Werbos, P.J., 1994. The Roots of Backpropagation: From Ordered Derivatives to Neural Networks and Political Forecasting. John Wiley & Sons.

Yoshida, T., Ohki, K., 2020. Natural images are reliably represented by sparse and variable populations of neurons in visual cortex. *Nature communications* 11, 872.

Zarkeshian, P., Kergan, T., Ghobadi, R., Nicola, W., Simon, C., 2022. Photons guided by axons may enable backpropagation-based learning in the brain. *Scientific Reports* 12, 20720.

## Appendix A. Details of Methods

### A.1. SoftWTA Weight Initialisation

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

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

$$W = \text{weight\_range} \cdot X$$

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

## Appendix B. Centre-Surround Filters through Dale’s Principle

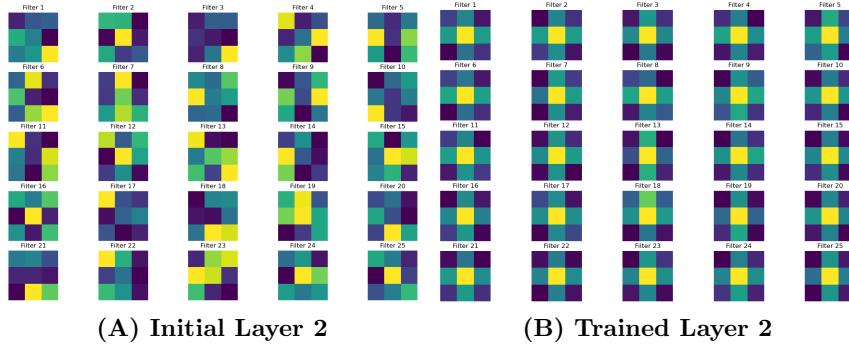


Figure B.14: Direct weight visualisation of filters in Layer 2 of **Dale\_Depthwise\_SoftHebb-Surr/HardWTA/Cos-BCM** before and after training. Note the exclusive formation of centre-surround filters.

The depthwise network (**Dale\_Depthwise\SoftHebb-Surr/HardWTA/Cos-BCM**) which fully respected Dale’s principle exhibited exclusively centre-surround patterns in its depthwise filters at Layers 2-3 (seen through the direct filter visualisation in Figure B.14), further accentuating the biological realism of combining the depthwise architecture with Hebbian Learning and its biologically-inspired competition mechanisms.

## Appendix C. Network Architectures

Table C.4: SoftHebb CNN Architecture (Non-Depthwise)

<b>Layer</b>	<b>Type</b>	<b>Output Shape</b>	<b>Kernel</b>	<b>Stride</b>	<b>Padding</b>	<b>Activation</b>
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.5: Lagani 3-layer CNN Architecture (Non-Depthwise)

<b>Layer</b>	<b>Type</b>	<b>Output Shape</b>	<b>Kernel</b>	<b>Stride</b>	<b>Padding</b>	<b>Activation</b>
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.6: 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.7: SoftHebb 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.8: 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)	—	—	—	—
<b>HebbianResidualBlock Internal Structure</b>						
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