Local Learning in Neural Networks Without Global Backpropagation

Open research specification 3 of 7

Krrish Choudhary

Verace Pvt. Ltd.

This document was first drafted in the form of a patent specification, which is why it speaks of an invention and of claims. No patent application was filed and none is intended. Where the text mentions a related or co-pending patent application, it means a companion document in this set, and none of them was filed. The work is published as open research.

Published under the PolyForm Noncommercial License 1.0.0

Copyright (c) 2026 Verace Pvt. Ltd. (https://verace.in)

Contents

1 TITLE **4**

2 FIELD **4**

3 BACKGROUND **4**

3.1 Limitations of Global Backpropagation 4

3.2 Prior Art in Local Learning 5

3.3 Deficiencies of the Prior Art 5

4 OBJECTS **6**

5 SUMMARY **7**

6 DESCRIPTION OF THE FIGURES **9**

7 DETAILED DESCRIPTION **37**

7.1 System Overview 38

7.2 Learning Mechanism 1: Precision-Weighted Predictive Coding Optimizer . . . 38

7.2.1 Prediction Error Computation 38

7.2.2 Precision-Weighted Loss 39

7.2.3 Precision Computation 39

7.2.4 Analytical Weight Updates for $\mathbf{W}_{\text{predict}}$ 39

7.2.5 Analytical Weight Updates for $\mathbf{W}_{\sigma}$ 40

7.3 Learning Mechanism 2: BCM Hebbian Optimizer with Anti-Hebbian Decorrelation 40

7.3.1 BCM-Modulated Hebbian Learning 40

7.3.2 Error-Modulated Hebbian Learning 41

7.3.3 Anti-Hebbian Lateral Decorrelation 41

7.3.4 Sigma-Squared Adaptive Learning Rate 41

7.3.5 BCM Threshold Update 41

7.4 Learning Mechanism 3: Neuromodulatory Plasticity Gate 42

7.4.1 Plasticity Computation 42

7.4.2 Developmental Modulation 42

7.5 Learning Mechanism 4: Brainstem Projection Learning 42

7.5.1 DA Projection 43

7.5.2 NE Projection 43

7.5.3 ACh Projection 43

7.5.4 5-HT Projection 43

7.6 Learning Mechanism 5: Subcortical Hebbian Learning 43

7.6.1 Thalamic Relay Learning . . . 43
7.6.2 Cortical Workspace Learning . . . 44
7.6.3 Cerebellar Forward Model Learning . . . 44
7.6.4 Lateral Inter-Column Learning . . . 44
7.7 Learning Mechanism 6: L5 Dendritic Backpropagation . . . 44
7.7.1 Error Signal Selection . . . 44
7.7.2 Analytical Chain Rule Through SwiGLU . . . 45
7.7.3 Dendritic Gate Updates . . . 45
7.7.4 Bootstrap-Safe Error Signal . . . 45
7.8 Learning Mechanism 7: Reward-Modulated Three-Factor Learning . . . 45
7.9 Learning Mechanism 8: Micro-Downscaling . . . 46
7.10 Learning Mechanism 9: Sigma-Squared Emergency Guard . . . 46
7.11 Learning Mechanism 10: Local Output Correction . . . 46
7.11.1 Analytical Cross-Entropy Gradient . . . 46
7.11.2 Output Bias Update . . . 47
7.11.3 Low-Rank Production Pathway . . . 47
7.12 Local Parameter Updates for Modulatory Circuits . . . 47
7.12.1 L4 Input Projection . . . 47
7.12.2 L1 Top-Down Modulation . . . 47
7.12.3 L6 Thalamic Feedback . . . 47
7.12.4 Oscillation Phase-Amplitude Coupling . . . 48
7.12.5 Interneuron Weights . . . 48
7.12.6 Working Memory Gates and Weights . . . 48
7.13 Coordination and Sequencing . . . 48

8 EXAMPLES 49
8.1 Example 1: Predictive Coding Update Execution . . . 49
8.2 Example 2: BCM Hebbian with Error Modulation . . . 49
8.3 Example 3: Neuromodulatory Gating During Surprising Input . . . 50
8.4 Example 4: L5 Dendritic Backpropagation with Bootstrap . . . 50
8.5 Example 5: Local Output Correction for Language Model Head . . . 51
8.6 Example 6: Micro-Downscaling Selectivity . . . 51

9 INDUSTRIAL APPLICABILITY 51
9.1 GPU Memory Reduction for Large Language Model Training . . . 52
9.2 Edge Device Training . . . 52
9.3 Continual Learning Systems . . . 52
9.4 Neuromorphic Hardware Acceleration . . . 52
9.5 Adaptive Language Processing . . . 52

10 CLAIMS **54**

1. TITLE

Local Learning in Neural Networks Without Global Backpropagation

2. FIELD

The present invention relates to machine learning and artificial neural networks. More particularly, the invention relates to systems and methods for training neural network architectures using exclusively local learning rules derived from neuroscience, wherein all synaptic weight updates are computed analytically from locally available signals at each processing unit without requiring global backpropagation of error gradients through the network. The invention encompasses predictive coding optimizers, Bienenstock–Cooper–Munro (BCM) Hebbian learning with anti-Hebbian decorrelation, neuromodulatory plasticity gating, brainstem projection learning, subcortical Hebbian learning, dendritic backpropagation within individual processing columns, reward-modulated three-factor learning, micro-downscaling for continuous homeostasis, precision variance emergency stabilization, and analytical local output correction for language model heads.

3. BACKGROUND

3.1. Limitations of Global Backpropagation

Since the seminal work of Rumelhart, Hinton, and Williams (1986), global backpropagation has been the dominant training algorithm for artificial neural networks. In backpropagation, an error signal computed at the output layer is propagated backwards through the entire network using the chain rule of calculus, and weight updates at every layer depend on gradients computed with respect to this global loss function. While effective in practice, global backpropagation suffers from several fundamental limitations.

First, backpropagation requires maintaining a complete computation graph of the forward pass, storing all intermediate activations for use during the backward pass. For deep networks with hundreds of layers and billions of parameters, this memory overhead constitutes a dominant fraction of total GPU VRAM consumption, often exceeding the memory required for the model weights themselves by a factor of three to ten.

Second, backpropagation imposes a strict sequential dependency: the backward pass cannot begin until the forward pass completes, and gradient updates at earlier layers must wait for gradient computation at later layers. This “backward locking” prevents true parallelism across layers and creates a computational bottleneck that scales linearly with network depth.

Third, and most critically for biological plausibility, backpropagation violates Dale’s law and the principle of locality that governs biological synaptic plasticity. In biological neural circuits, a synapse has access only to signals available at its pre-synaptic and post-synaptic terminals,

supplemented by diffuse neuromodulatory signals. The synapse has no access to error gradients computed at distant neurons multiple layers away. This biological implausibility is not merely an aesthetic concern; it suggests that backpropagation may not discover the learning algorithms that evolution found effective over hundreds of millions of years of optimization.

3.2. Prior Art in Local Learning

Donald Hebb (1949) proposed the first local learning rule: "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." This "fire together, wire together" principle forms the basis of associative learning but suffers from instability — pure Hebbian learning leads to unbounded weight growth.

Bienenstock, Cooper, and Munro (1982) introduced the BCM rule, which adds a sliding threshold θ_{BCM} such that synaptic modifications are potentiating when post-synaptic activity exceeds θ_{BCM} and depressing when below. This prevents runaway potentiation but does not address how prediction errors should guide learning.

Rao and Ballard (1999) proposed predictive coding as a model of cortical computation, wherein each cortical area generates predictions of activity in the area below it, and the resulting prediction errors drive local weight updates. Whittington and Bogacz (2017) showed that predictive coding can approximate backpropagation under certain conditions. However, these prior works remain theoretical and do not disclose a complete, implementable system for training large-scale neural networks using predictive coding combined with multiple complementary local learning rules.

Schultz (1997) demonstrated that midbrain dopamine neurons encode reward prediction errors, establishing a biological basis for reinforcement learning. However, prior art does not disclose how dopaminergic and other neuromodulatory signals can be integrated with predictive coding and Hebbian learning in a unified local learning framework for language modeling.

Sacramento et al. (2018) proposed dendritic error propagation as a biologically plausible alternative to backpropagation, wherein error signals propagate through dendritic compartments of pyramidal neurons rather than through a separate backward pathway. However, this work does not address integration with neuromodulatory gating, BCM homeostasis, or subcortical learning systems.

3.3. Deficiencies of the Prior Art

The prior art fails to disclose:

(a) A complete system that trains a large-scale neural network using exclusively local learning rules with zero global backpropagation, achieving competitive language modeling performance.

(b) A precision-weighted predictive coding optimizer wherein per-column prediction errors are normalized by learned precision estimates (σ^2) computed through a softplus-transformed linear projection, with all updates computed analytically under `torch.no_grad()`.

(c) Integration of BCM-modulated Hebbian learning with error-modulated Hebbian learning, anti-Hebbian decorrelation, sigma-squared-adaptive learning rates, and metaplasticity-modulated learning rates in a single unified optimizer.

(d) A neuromodulatory plasticity gate that multiplicatively combines four neuromodulator signals (dopamine, acetylcholine, norepinephrine, serotonin) to gate all learning in the network.

(e) Analytical Hebbian learning rules for brainstem neuromodulator projections, wherein each projection weight learns to correlate with a specific aspect of prediction error.

(f) Subcortical Hebbian learning rules for thalamic relay, cortical workspace, cerebellar forward model, and lateral inter-column connections.

(g) Dendritic backpropagation within a single cortical column through SwiGLU nonlinearity with analytical chain rule computation for basal and apical dendritic gates.

(h) Micro-downscaling as continuous homeostatic regulation of synaptic weights applied selectively to association weights while protecting gates, norms, and biases.

(i) Sigma-squared emergency guards that detect and correct precision parameter divergence.

(j) Analytical local output correction for language model heads using the delta rule on softmax error, with low-rank weight decomposition, computed without autograd.

4. OBJECTS

It is a primary object of the present invention to provide a system and method for training neural networks using exclusively local learning rules that require no global backpropagation of error gradients, thereby eliminating the need to store a complete computation graph and reducing memory consumption by a factor of approximately three compared to standard backpropagation training.

It is a further object of the invention to provide a precision-weighted predictive coding optimizer in which each processing column independently minimizes its own prediction error using analytically computed weight updates, wherein the error signal is normalized by a learned precision estimate to enable automatic calibration of confidence.

It is another object of the invention to provide a BCM-modulated Hebbian learning system with anti-Hebbian decorrelation and error modulation, wherein self-organizing lateral associations

arise from purely local correlational learning modulated by a sliding threshold that prevents runaway potentiation.

It is yet another object of the invention to provide a neuromodulatory plasticity gate that multiplicatively combines four neuromodulator signals to determine when, what, how broadly, and over what timescale learning occurs, thereby implementing reward-guided learning without global backpropagation.

It is a still further object of the invention to provide analytical Hebbian learning rules for brainstem neuromodulator projections, subcortical relay systems, and inter-column lateral connections, such that the entire neural network, including all subcortical processing modules, learns through purely local rules.

It is another object of the invention to provide dendritic backpropagation within individual processing columns, wherein top-down error signals propagate analytically through SwiGLU feedforward networks and dendritic gates, enabling learning in deep column internal pathways without requiring global gradient flow.

It is a further object of the invention to provide continuous homeostatic regulation through micro-downscaling of synaptic weights and emergency stabilization of precision parameters, ensuring training stability without external learning rate schedulers.

It is yet a further object of the invention to provide an analytical local output correction method for language model output heads, wherein the cross-entropy gradient is computed directly from softmax error and applied via the delta rule with low-rank weight decomposition, achieving language model training with zero autograd dependency.

5. SUMMARY

The present invention provides a system and method for training neural networks using a coordinated ensemble of local learning rules that collectively replace global backpropagation. The system comprises ten interoperating learning mechanisms, each computing weight updates from signals locally available at the synapse being modified, supplemented only by globally broadcast neuromodulatory signals.

In a first aspect, the invention provides a **predictive coding optimizer** that operates on a hierarchical network of processing columns. Each column l generates a prediction $\hat{x}_l$ of its own input and computes a prediction error $\epsilon_l = x_l - \hat{x}_l$. A prediction weight matrix $\mathbf{W}_{\text{predict}}$ is updated according to:

$$\Delta \mathbf{W}_{\text{predict}} = -\frac{\eta}{N} \left(\frac{\epsilon_l}{\sigma_l^2} \right) \cdot h_{\text{norm}}^{\top} \tag{1}$$

where σ_l^2 is a learned precision parameter computed via $\sigma^2 = \text{softplus}(\mathbf{W}_\sigma \cdot h_{\text{norm}}) + \varepsilon$, and h_{norm} is the layer-normalized hidden state. The precision weight $\mathbf{W}_\sigma$ is updated via the chain

rule through softplus:

$$\Delta \mathbf{W}_\sigma = -\frac{\eta}{N}\left(\frac{\partial \mathcal{L}}{\partial \sigma^2} \cdot \text{sigmoid}(z)\right) \cdot h_{\text{norm}}^\top \tag{2}$$

All updates are computed under `torch.no_grad()` with explicit gradient norm clipping.

In a second aspect, the invention provides a **BCM Hebbian optimizer** with three complementary learning rules: (i) BCM-modulated Hebbian learning $\Delta w \propto y(y - \sqrt{\theta_{\text{BCM}}})x$ for association weights, (ii) anti-Hebbian lateral learning $\Delta \mathbf{W}_{\text{lateral}} = \eta(Y^\top Y - I)/N$ for decorrelation, and (iii) error-modulated Hebbian learning that blends correlational and error-driven signals with a configurable weight. The optimizer further incorporates sigma-squared-adaptive learning rates wherein uncertain columns learn faster, and metaplasticity-modulated learning rates that track weight change history.

In a third aspect, the invention provides a **neuromodulatory plasticity gate** that computes:

$$\text{plasticity} = \text{DA} \times \text{ACh} \times (0.5 + 0.5 \times \text{NE}) \times (0.5 + 0.5 \times \text{5-HT}) \tag{3}$$

where DA (dopamine) gates whether to learn, ACh (acetylcholine) gates what to learn, NE (norepinephrine) modulates learning breadth, and 5-HT (serotonin) modulates temporal horizon. This plasticity signal multiplicatively gates all predictive coding and Hebbian updates.

In a fourth aspect, the invention provides **brainstem projection learning** wherein neuromodulator projections learn analytical Hebbian correlations: DA correlates with error magnitude, NE with prediction uncertainty (σ^2), ACh with inverse error (confidence), and 5-HT with error temporal stability.

In a fifth aspect, the invention provides **subcortical Hebbian learning** for thalamic relay weights, cortical workspace broadcast weights, cerebellar temporal prediction weights, and lateral inter-column connection weights, each using error-correlation Hebbian rules.

In a sixth aspect, the invention provides **dendritic backpropagation** within individual columns, wherein top-down error propagates analytically through SwiGLU feedforward networks via $\delta_{\text{ff}} = \delta_{\text{out}} \cdot W_{\text{down}}$, $\delta_{\text{gate}} = \delta_{\text{ff}} \cdot u \cdot \text{SiLU}'(g)$, and $\delta_{\text{up}} = \delta_{\text{ff}} \cdot \text{SiLU}(g)$, updating W_{down}, W_{gate}, W_{up}, basal dendritic gate W_{dend}, and apical dendritic gate W_{apical}.

In a seventh aspect, the invention provides **reward-modulated three-factor learning** $\Delta w = \text{pre} \times \text{post} \times \text{reward}$ with hippocampal surprise modulation.

In an eighth aspect, the invention provides **micro-downscaling** applied every k steps: $W \leftarrow W \times (1 - 0.05 \times s)$ for synaptic association weights only, protecting gates, norms, and biases from downscaling.

In a ninth aspect, the invention provides a **sigma-squared emergency guard** that detects divergence of precision parameters and applies corrective scaling.

In a tenth aspect, the invention provides **local output correction** for language model heads

using the analytical delta rule:

$$\Delta W_{\text{head}} = -\frac{1}{N}(\text{softmax}(\text{logits}) - \text{one_hot}(\text{targets}))^\top \cdot h \tag{4}$$

with gradient clipping and low-rank decomposition ΔA, ΔB for the production pathway.

6. DESCRIPTION OF THE FIGURES

Figure P3-1

is a system architecture diagram showing the local learning system of Claim 1, illustrating all six subsystems — predictive coding optimizer, BCM Hebbian optimizer, neuromodulatory plasticity gate, dendritic error propagation, local output correction, and the no-gradient context — and their interconnections within a hierarchical stack of processing columns.

Figure P3-2

is a method flow diagram showing the training method of Claim 2, illustrating the sequential steps of forward pass, prediction error computation, neuromodulatory plasticity signal computation, analytical weight updates, BCM Hebbian updates, dendritic error propagation, output correction, micro-downscaling, and precision monitoring, all within a no-gradient context.

Figure P3-3

is a block diagram showing the non-transitory computer-readable medium of Claim 3, illustrating the six instruction groups stored on the medium and their execution flow on a processor coupled to a graphics processing unit.

Figure P3-4

is a detailed diagram of the precision estimate computation and $\mathbf{W}_\sigma$ update of Claim 4, showing the chain rule through softplus, sigmoid clamping to $[0.01, 0.99]$, and the MSE-based gradient with dimension-invariant scaling.

Figure P3-5

is a detailed diagram of the BCM-modulated Hebbian learning rule of Claim 5, showing the BCM modulation function $y(y - \sqrt{\theta_{\text{BCM}}})$, the metaplasticity multiplier m_{meta}, and the sigma-squared-adaptive learning rate multiplier $s_{\sigma^2} = 1 + \min(\overline{\sigma^2}, 10)$.

Figure P3-6

is a diagram of the error-modulated Hebbian mode of Claim 6, showing the blending of correlational BCM and error-driven signals via the configurable error weight parameter w_e.

Figure P3-7

is a block diagram of the neuromodulatory plasticity gate of Claim 7, showing the four neuromodulator inputs (DA, ACh, NE, 5-HT), their multiplicative combination, and the affine transformations f_{NE} and $f_{\text{5-HT}}$ that ensure learning never drops below half rate.

Figure P3-8

is a diagram of the brainstem module of Claim 8, showing the four learned projection weights and their analytical Hebbian correlation rules: DA with error magnitude, NE with prediction uncertainty (σ^2), ACh with inverse error, and 5-HT with error temporal stability.

Figure P3-9

is a diagram of the brainstem weight constraints of Claim 9, showing the $[-1, 1]$ clamping applied after each projection update and the fractional learning rate relationship to the Hebbian learning rate.

Figure P3-10

is a block diagram of the thalamic relay learning module of Claim 10, showing the relay input and output weight updates via error-correlation Hebbian rules and the aggregate error signals from served processing columns.

Figure P3-11

is a block diagram of the cortical workspace learning module of Claim 11, showing surprise-weighted Hebbian learning for write weights and error-reduction Hebbian learning for read weights.

Figure P3-12

is a diagram of the cerebellar learning module of Claim 12, showing the Marr–Albus–Ito delta rule on temporal prediction weights, with the climbing fiber error signal and parallel fiber–Purkinje cell synapse modification.

Figure P3-13

is a diagram of the lateral inter-column learning module of Claim 13, showing the send and receive weight updates between adjacent processing columns via error-correlation Hebbian rules.

Figure P3-14

is a detailed diagram of the dendritic gate learning of Claim 14, showing the basal dendritic gate weight (feedforward input from L4) and apical dendritic gate weight (top-down context), both modulating the feedforward network via sigmoid gating.

Figure P3-15

is a flowchart of the bootstrap-safe error signal selection of Claim 15, showing the detection of near-zero down-projection weight norm and the substitution of mean error magnitude broadcast signal during early training.

Figure P3-16

is a block diagram of the low-rank output correction of Claim 16, showing the production pathway matrices $\Delta A \in \mathbb{R}^{V \times r}$ and $\Delta B \in \mathbb{R}^{r \times D}$, their analytical updates, and gradient norm clipping.

Figure P3-17

is a diagram of the precision weight update method of Claim 17, showing the chain rule through softplus for computing $\Delta \mathbf{W}_\sigma$ with MSE-based gradient and dimension-invariant scaling.

Figure P3-18

is a diagram of the sigma-squared-adaptive learning rate and metaplasticity multiplier of Claim 18, showing how columns with high prediction uncertainty learn faster via $s = 1 + \min(\overline{\sigma^2}, 10)$ and how the metaplasticity multiplier tracks recent weight change magnitude.

Figure P3-19

is a diagram of the brainstem neuromodulator projection updates of Claim 19, showing the analytical Hebbian correlation rules for all four projections (DA, NE, ACh, 5-HT) at a fraction of the Hebbian learning rate.

Figure P3-20

is a diagram of the local modulatory circuit parameter updates of Claim 20, showing updates for L4 input projection, L1 top-down modulation, L6 thalamic feedback, oscillatory coupling, and interneuron weights.

Figure P3-21

is a diagram of the working memory gate and weight updates of Claim 21, showing surprise-proportional gate opening when prediction error exceeds a threshold, surprise-driven Hebbian write weight correlation, and error-reduction Hebbian read weight updates.

Figure P3-22

is a flowchart of the bootstrap-safe error signal selection method of Claim 22, showing the conditional substitution of mean error magnitude for chain-rule error when the down-projection weight norm falls below a threshold.

Figure P3-23

is a timing diagram of the micro-downscaling of Claim 23, showing the selective multiplication of synaptic weight matrices by $(1 - 0.05 \times (1 - s))$ every k steps, with protected parameters (gates, norms, biases) excluded.

Figure P3-24

is a diagram of the precision parameter divergence monitor of Claim 24, showing the threshold check on $\max(|b_\sigma|)$ exceeding 20.0 and the corrective multiplicative scaling of precision weight matrix and bias by 0.9.

Figure P3-25

is a diagram of the elastic weight consolidation of Claim 25, showing the elastic pull of column parameters toward anchor values weighted by per-parameter importance estimates for continual learning protection.

Figure P3-26

is a comprehensive diagram of the subcortical system learning of Claim 26, showing the analytical Hebbian rules for brainstem neuromodulator projections, thalamic relay weights, workspace broadcast weights, cerebellar prediction weights, and lateral inter-column connection weights.

ARCHITECTURAL OVERVIEW FIGURES

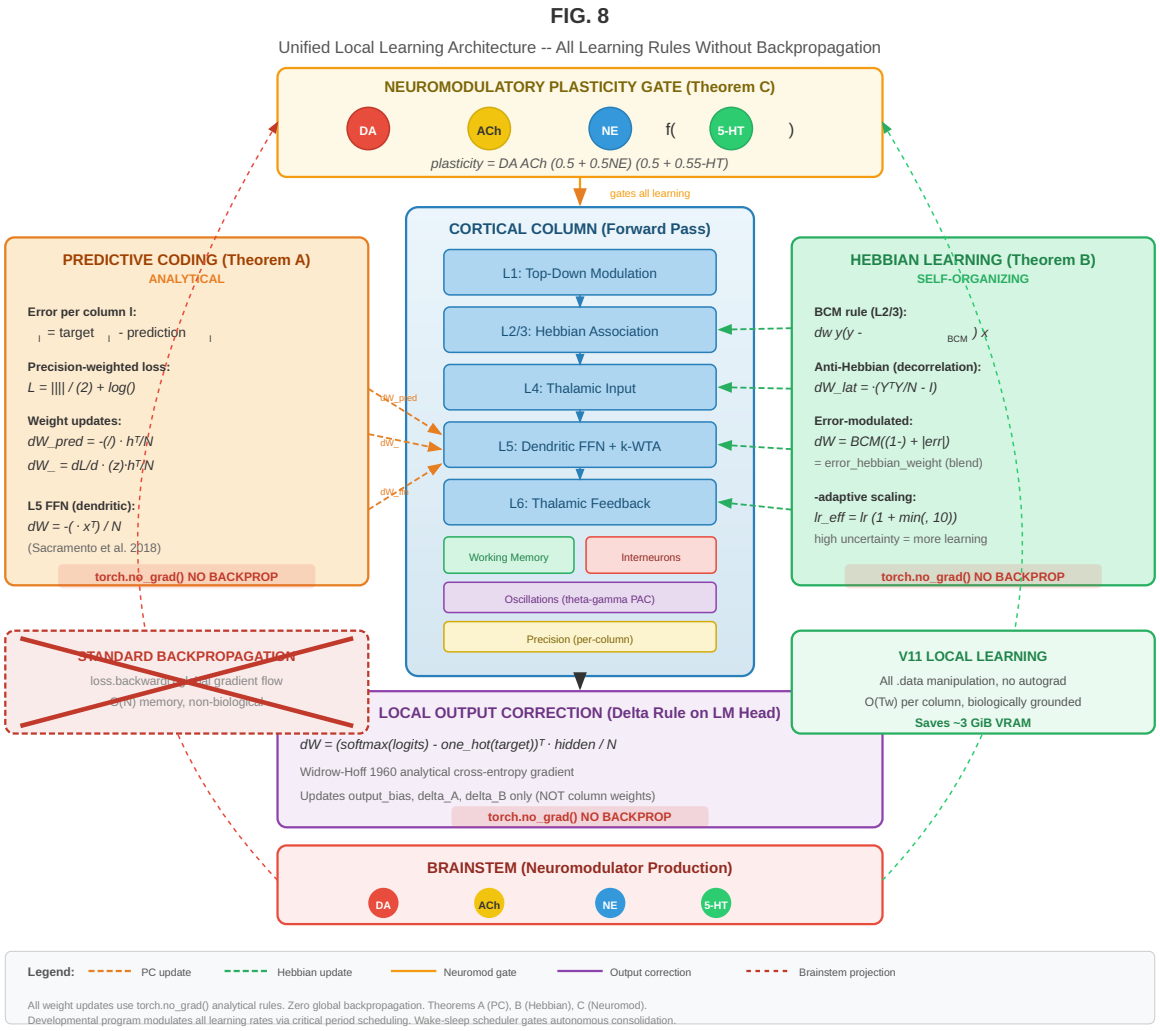


Figure 1: FIG. 1 — Ten Local Learning Mechanisms and Their Interconnections

FIG. 9

Neuromodulatory Plasticity Gate -- Four-Signal Learning Control

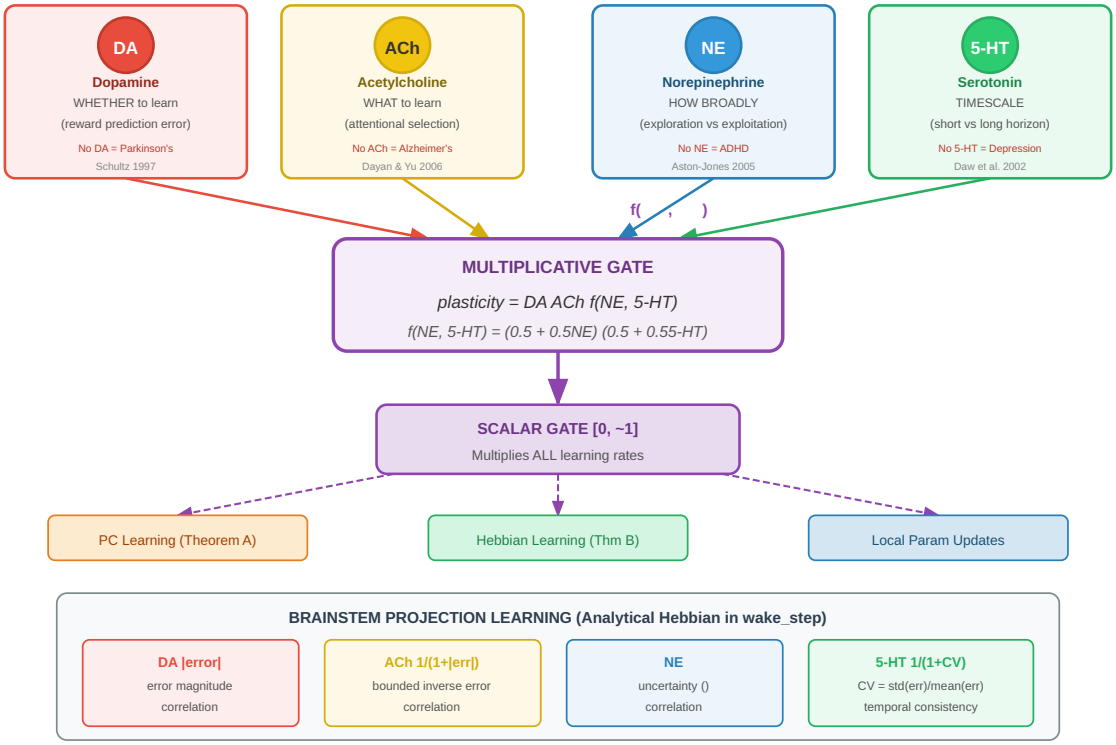


Figure 2: FIG. 2 — Neuromodulatory Plasticity Gate (DA, ACh, NE, 5-HT)

FIG. 10

Subcortical Learning Rules -- All Analytical, No Backpropagation

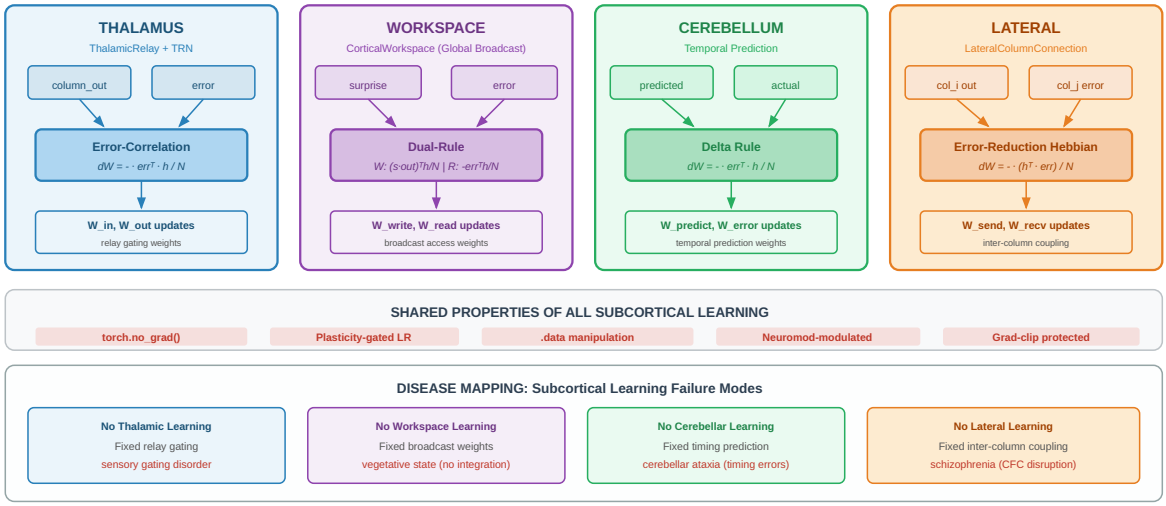
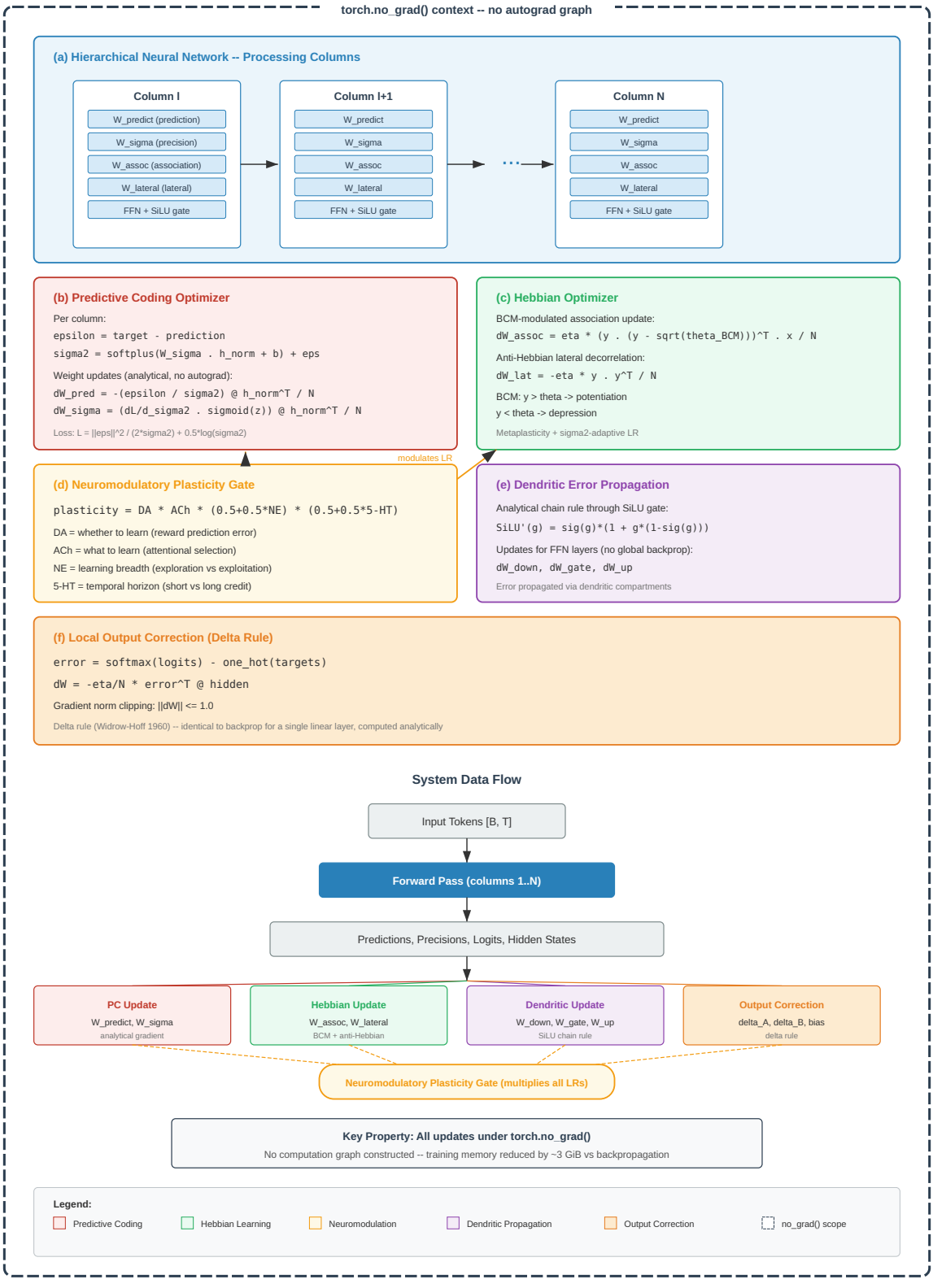


Figure 3: FIG. 3 — Subcortical Hebbian Learning Pathways

CLAIM ILLUSTRATION SHEETS

FIG. P3-1

System for Training a Neural Network Using Local Learning Rules Without Global Backpropagation



Patent 3 -- Claim 1 (System)

Figure 4: System for training a neural network using local learning rules without global backpropagation (Claim 1).

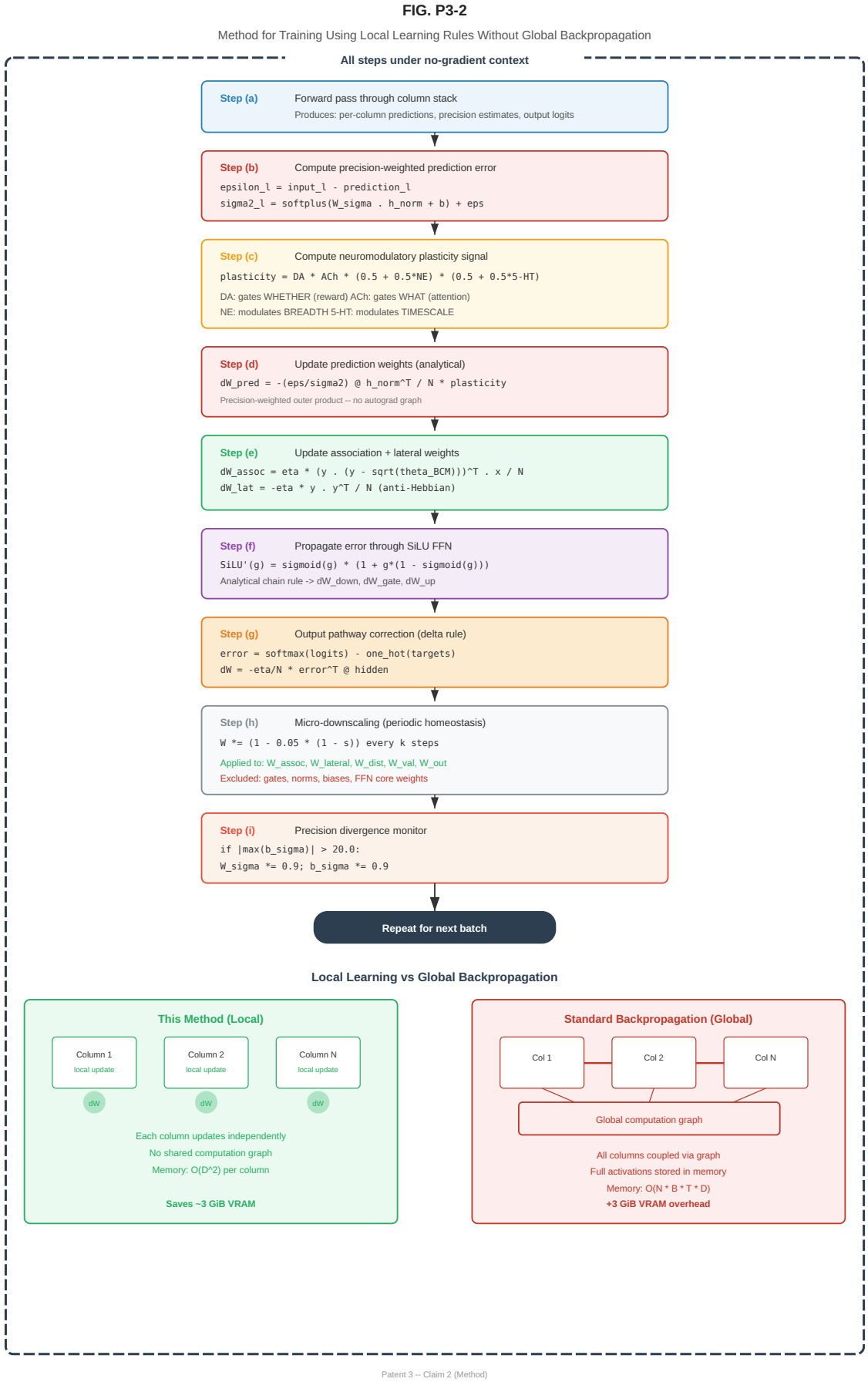


Figure 5: Method for training using local learning rules without global backpropagation (Claim 2).

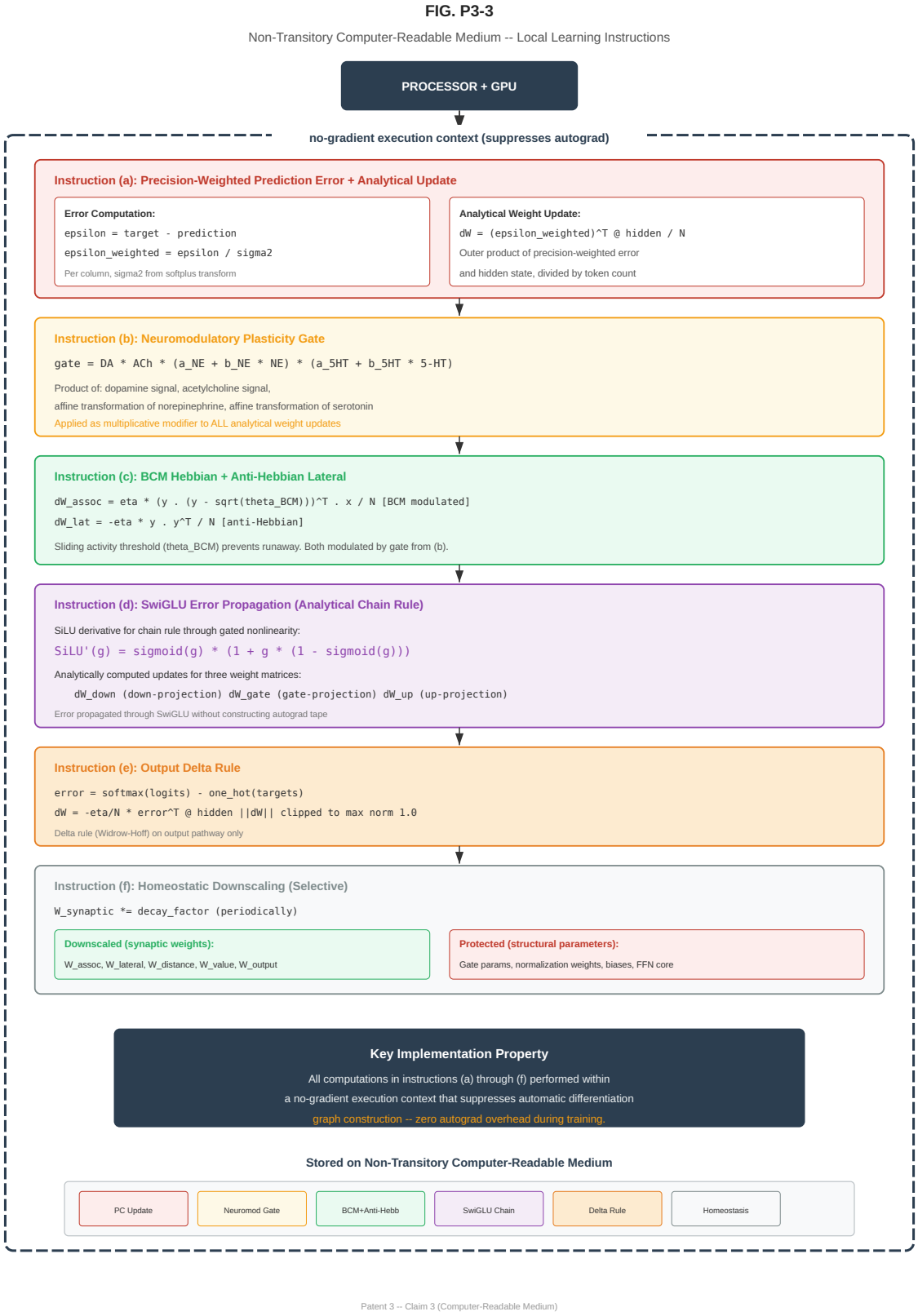


Figure 6: Non-transitory computer-readable medium storing local learning instructions (Claim 3).

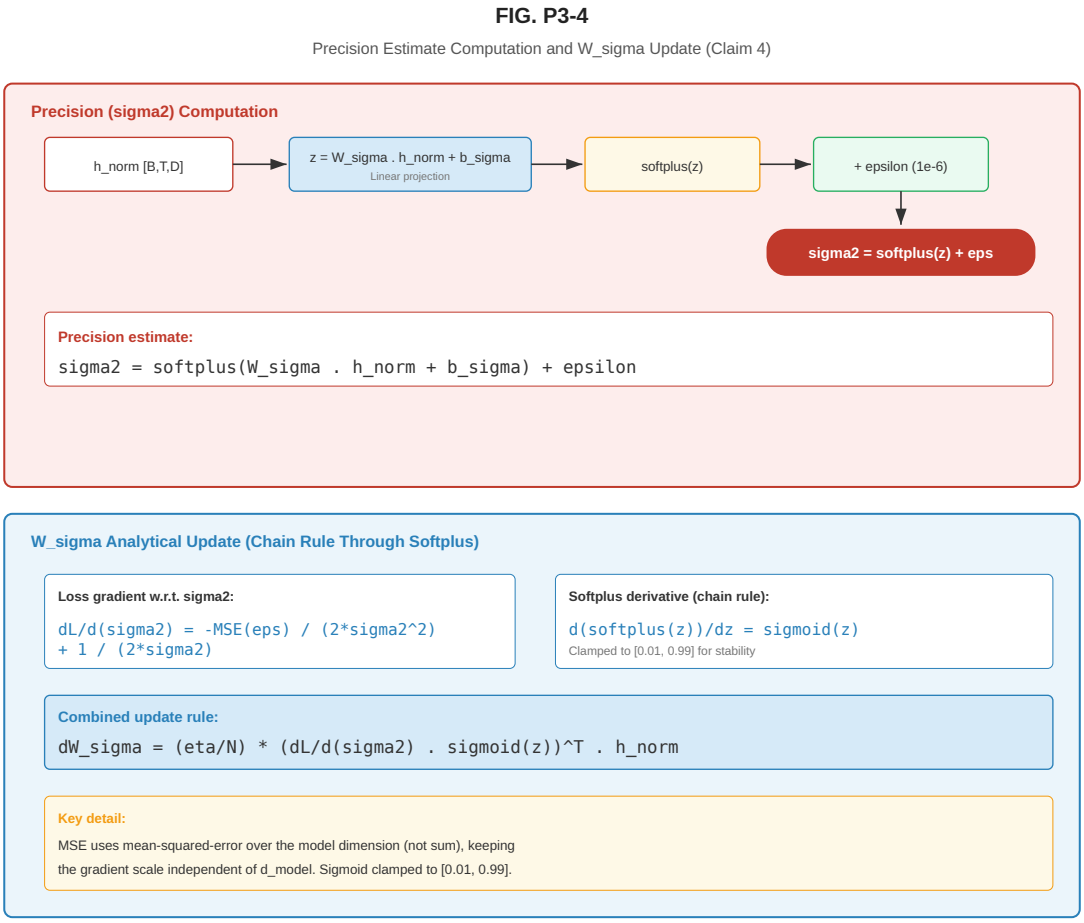


Figure 7: Precision estimate computation and $\mathbf{W}_\sigma$ update via chain rule through softplus (Claim 4).

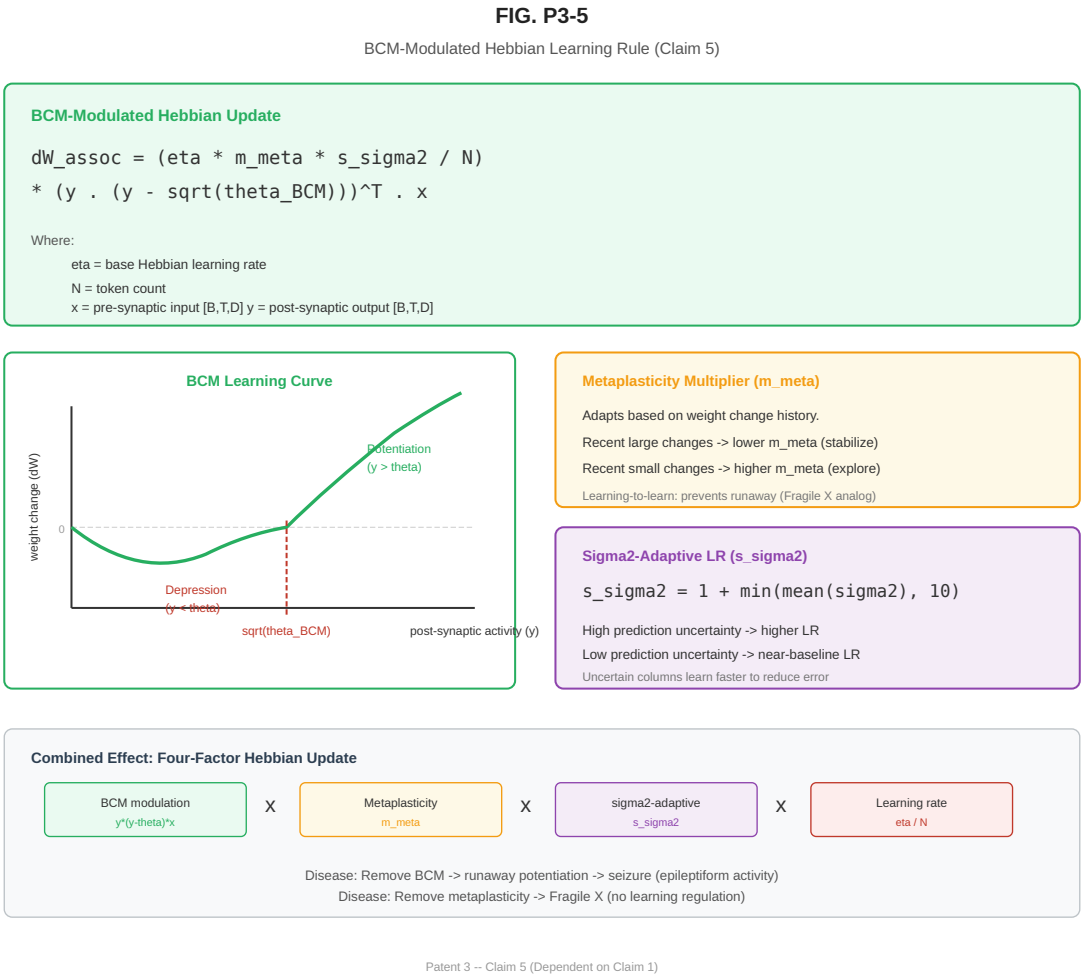


Figure 8: BCM-modulated Hebbian learning rule with metaplasticity and σ^2-adaptive rate (Claim 5).

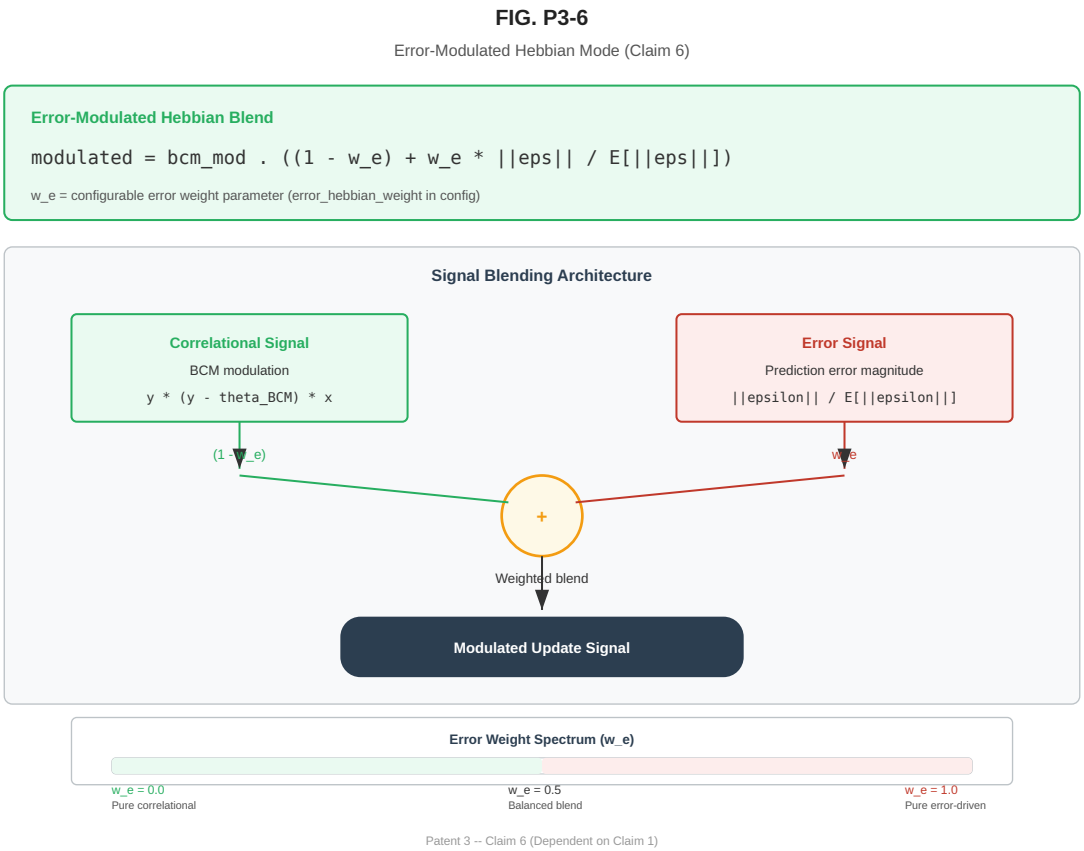


Figure 9: Error-modulated Hebbian mode blending correlational and error-driven signals (Claim 6).

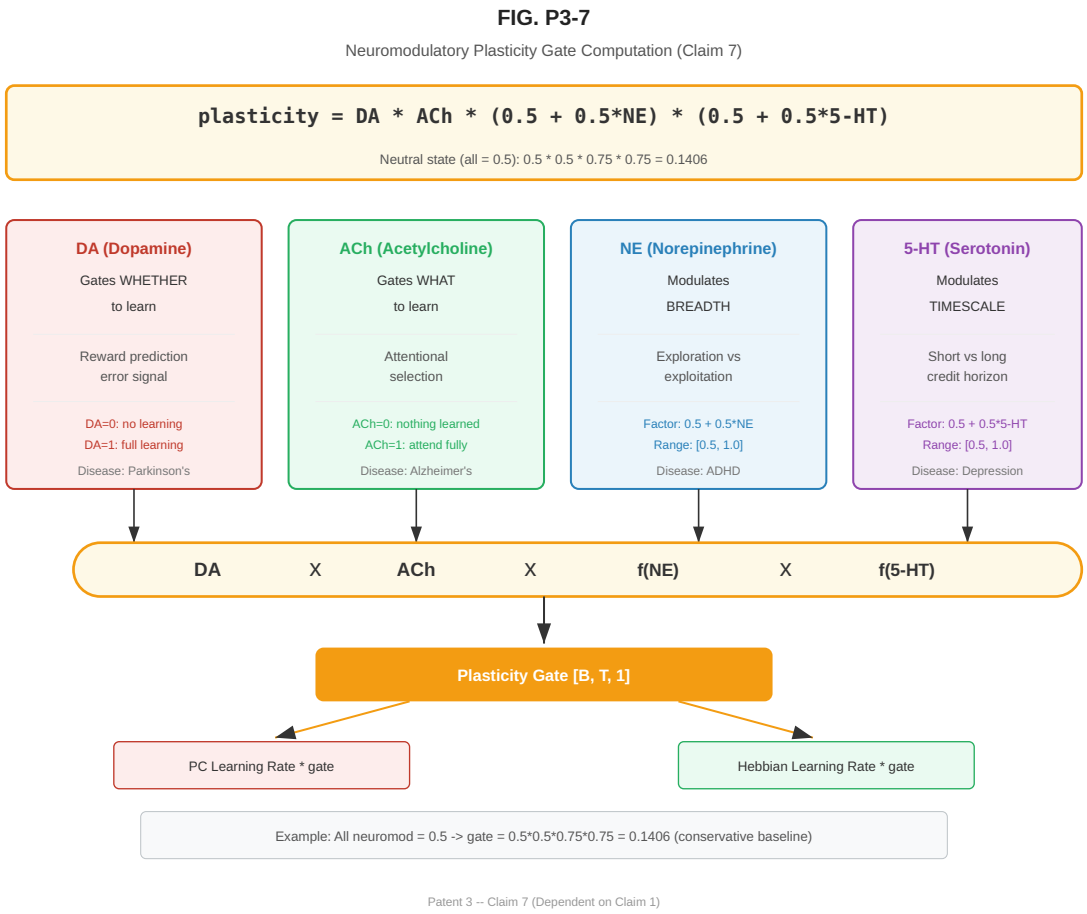


Figure 10: Neuromodulatory plasticity gate: DA $\times$ ACh $\times$ f(NE) $\times$ f(5-HT) (Claim 7).

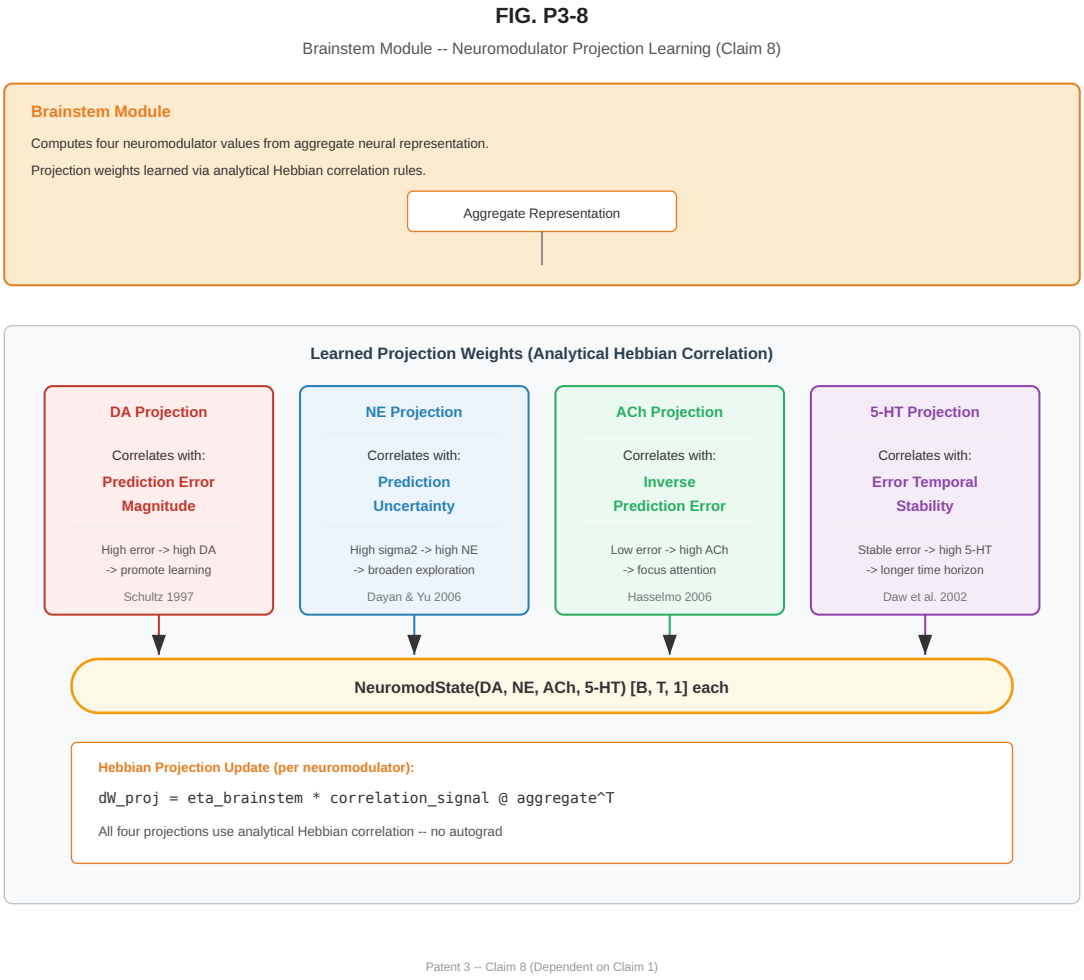


Figure 11: Brainstem module with four learned neuromodulator projection weights (Claim 8).

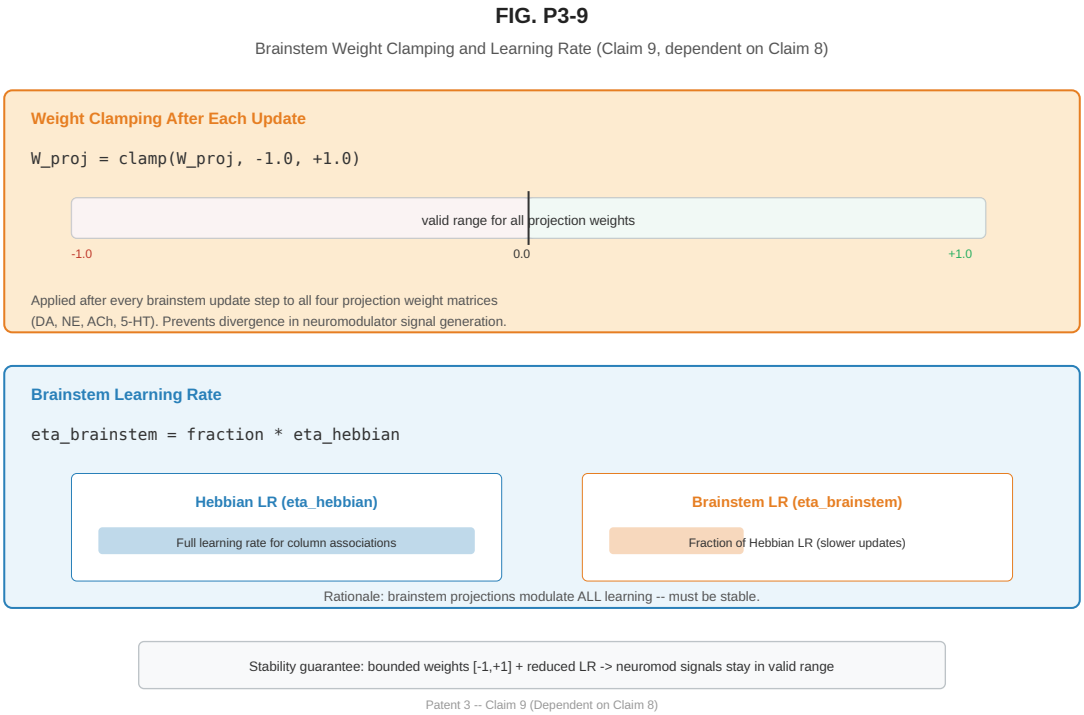


Figure 12: Brainstem weight clamping to $[-1, 1]$ and fractional learning rate (Claim 9).

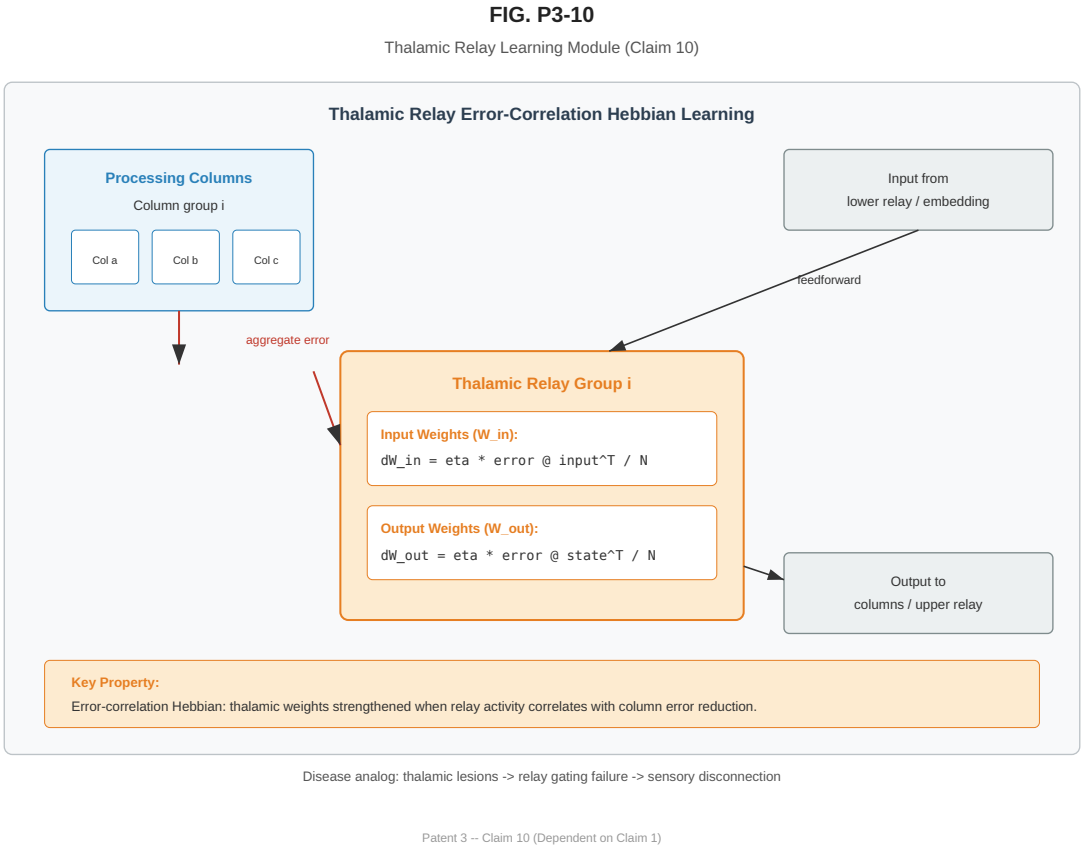


Figure 13: Thalamic relay learning via error-correlation Hebbian rules (Claim 10).

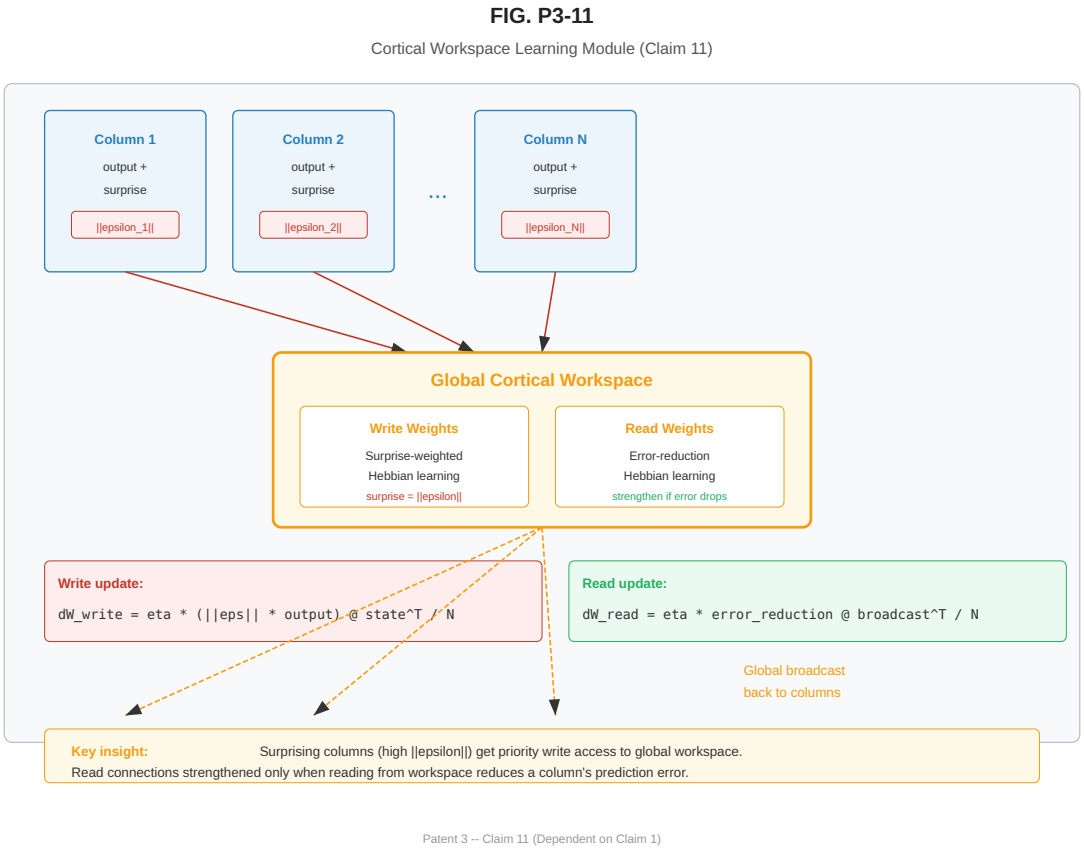
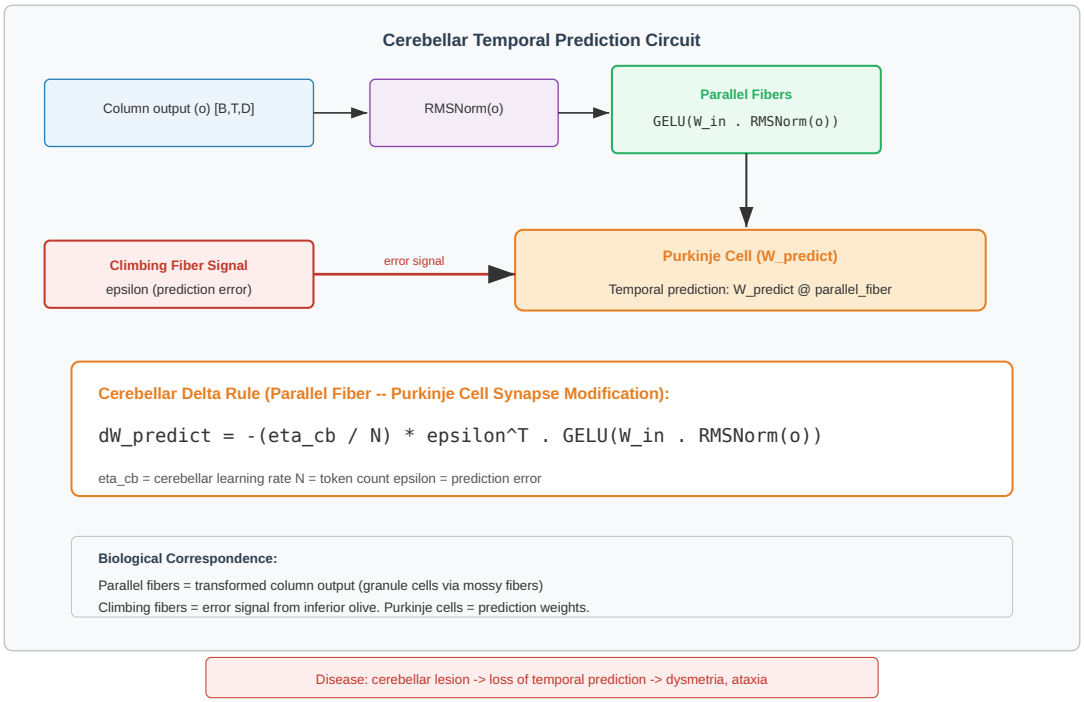


Figure 14: Cortical workspace learning: surprise-weighted write, error-reduction read (Claim 11).

FIG. P3-12

Cerebellar Learning Module -- Delta Rule on Temporal Predictions (Claim 12)

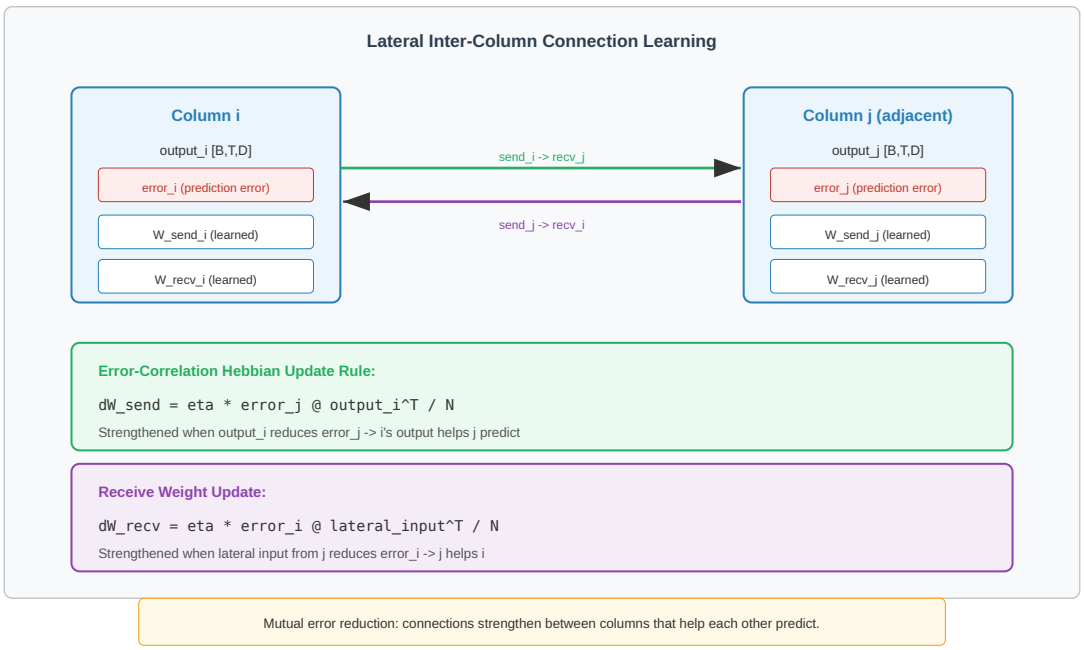


Patent 3 -- Claim 12 (Dependent on Claim 1)

Figure 15: Cerebellar learning module implementing the Marr–Albus–Ito delta rule (Claim 12).

FIG. P3-13

Lateral Inter-Column Learning (Claim 13)



Patent 3 -- Claim 13 (Dependent on Claim 1)

Figure 16: Lateral inter-column learning via error-correlation Hebbian rules (Claim 13).

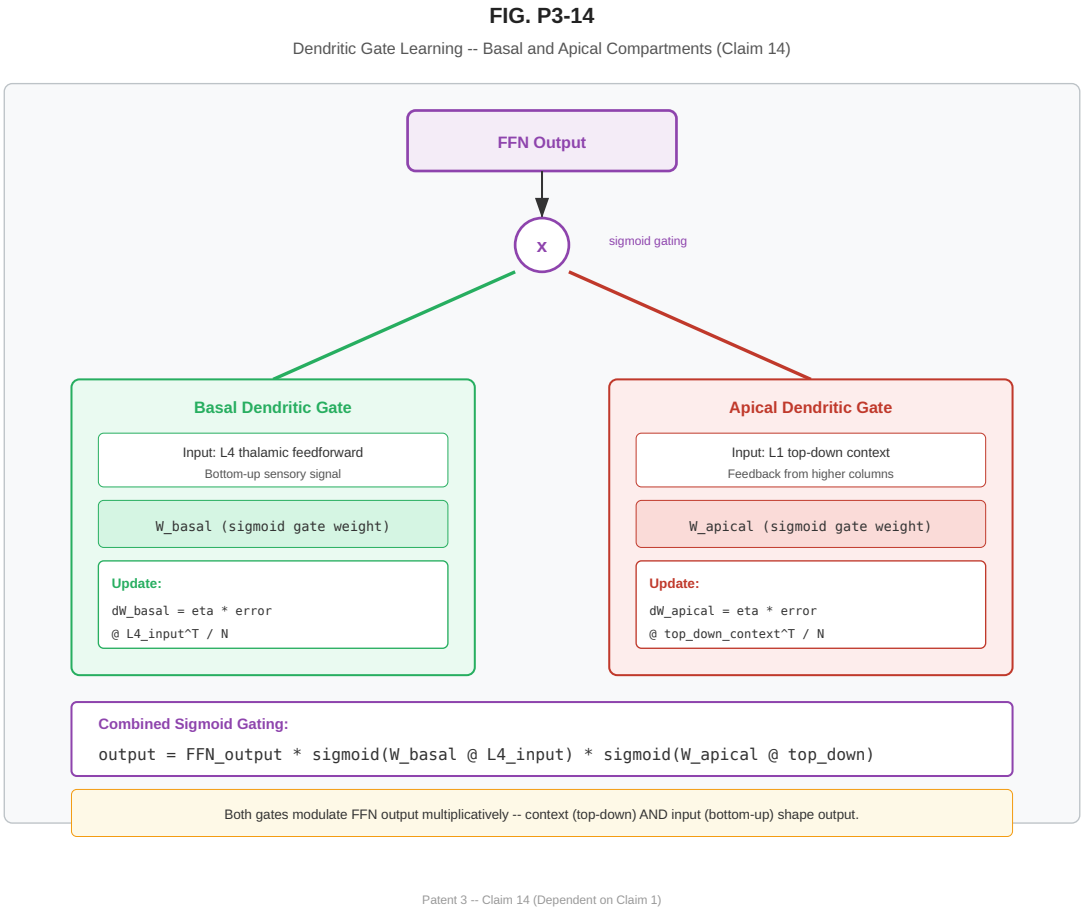


Figure 17: Dendritic gate learning: basal (L4 feedforward) and apical (top-down) compartments (Claim 14).

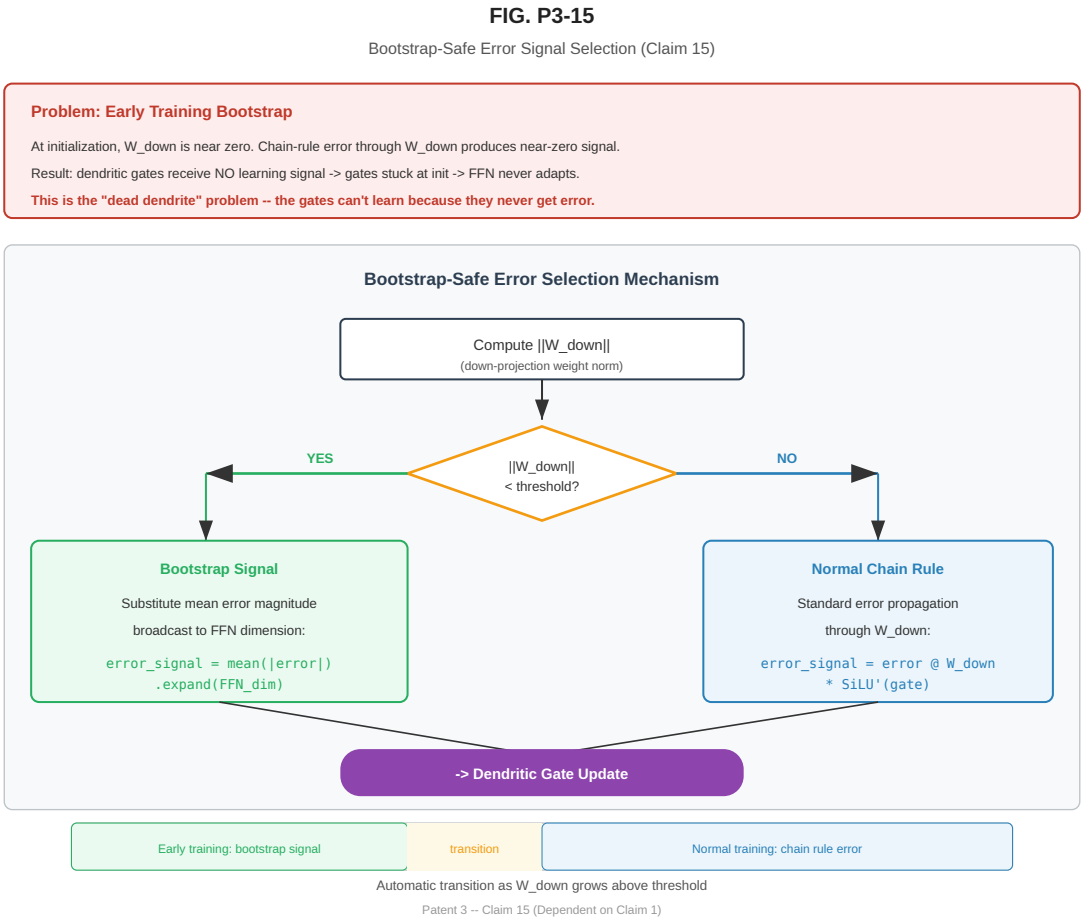


Figure 18: Bootstrap-safe error signal selection for early training (Claim 15).

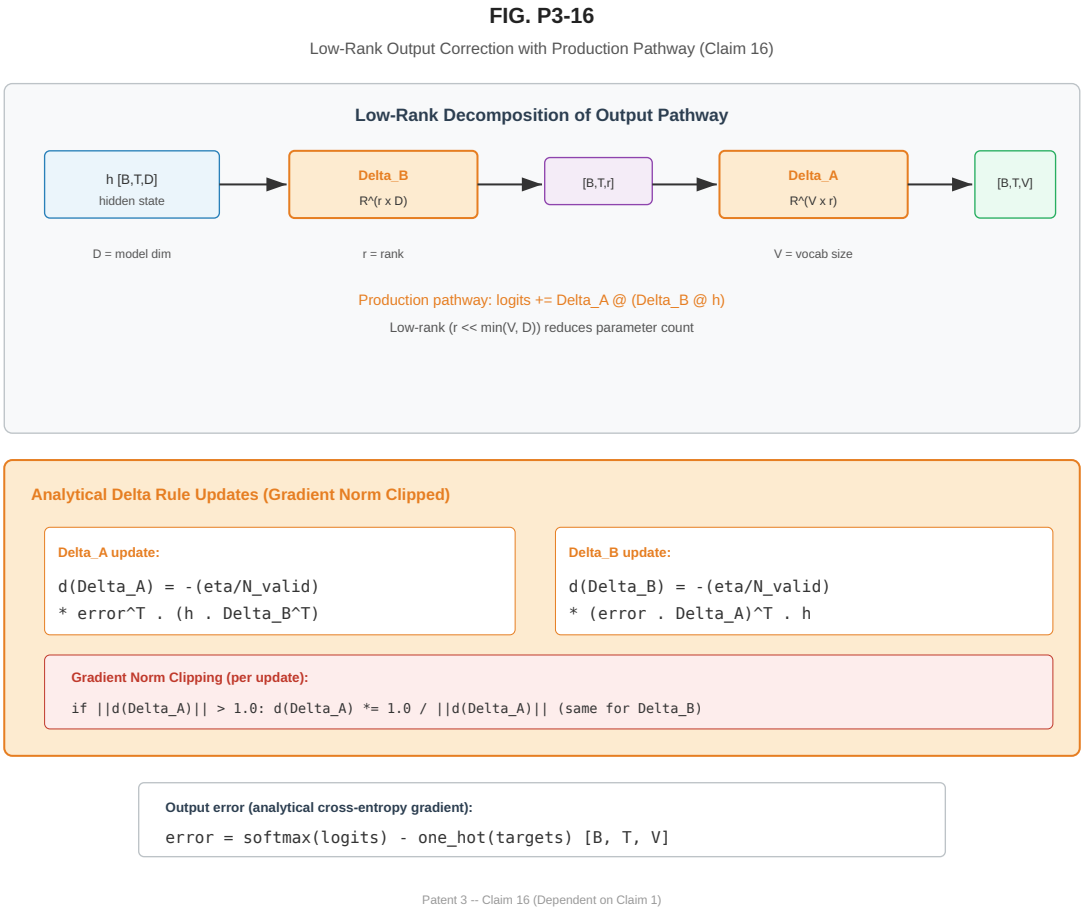


Figure 19: Low-rank output correction with production pathway matrices ΔA, ΔB (Claim 16).

FIG. P3-17

Precision Weight Update via Chain Rule Through Softplus (Claim 17, Method)

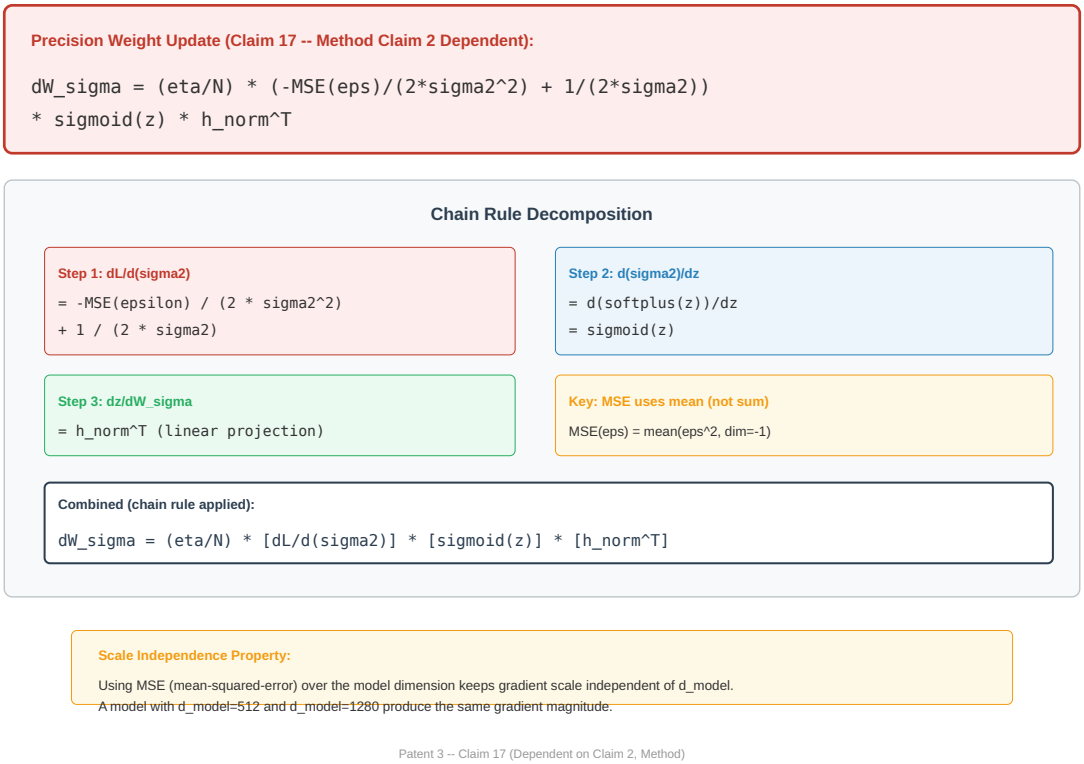


Figure 20: Precision weight update method: chain rule through softplus (Claim 17).

FIG. P3-18

Sigma2-Adaptive LR and Metaplasticity Multiplier for BCM (Claim 18)

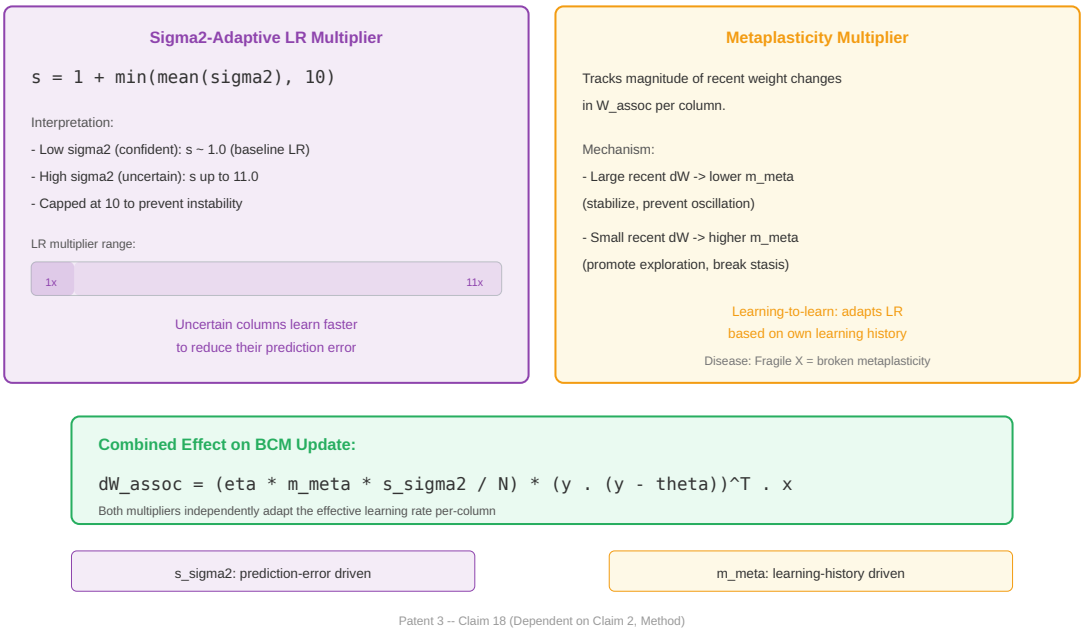


Figure 21: σ^2-adaptive learning rate and metaplasticity multiplier for BCM updates (Claim 18).

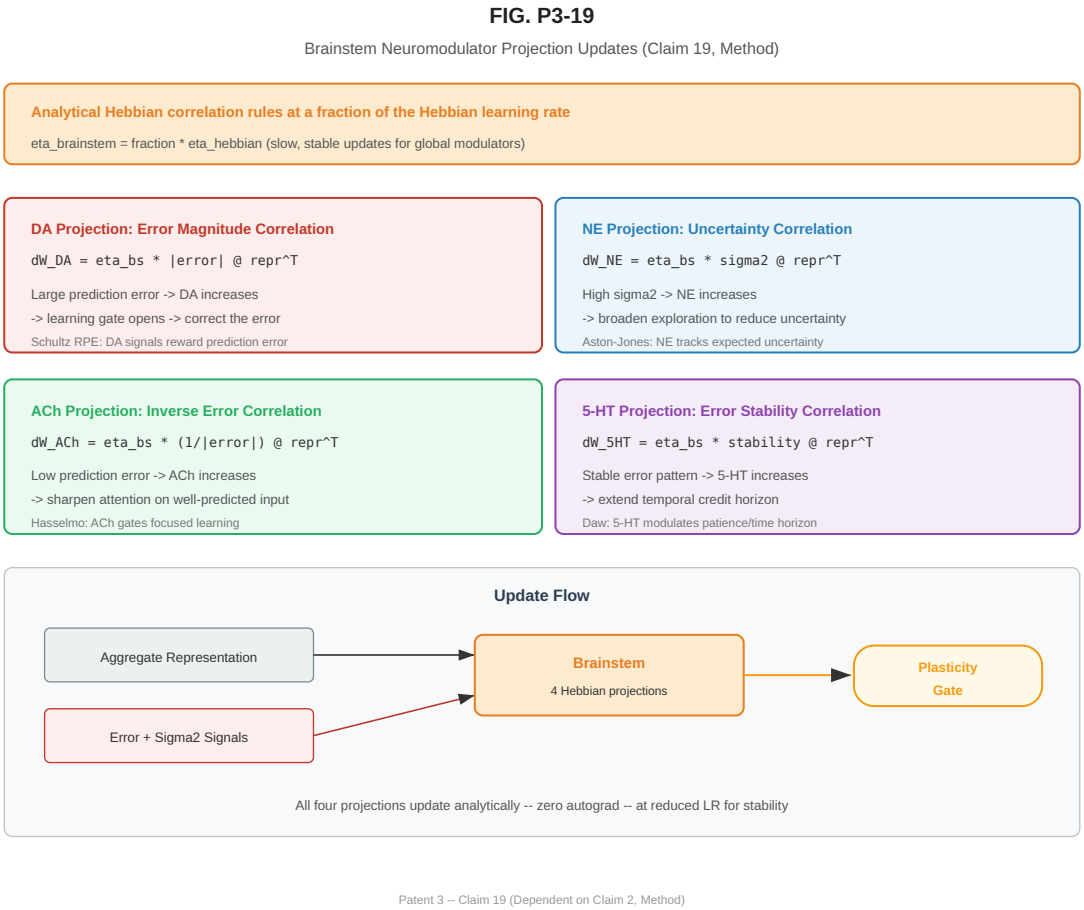


Figure 22: Brainstem neuromodulator projection updates via analytical Hebbian rules (Claim 19).

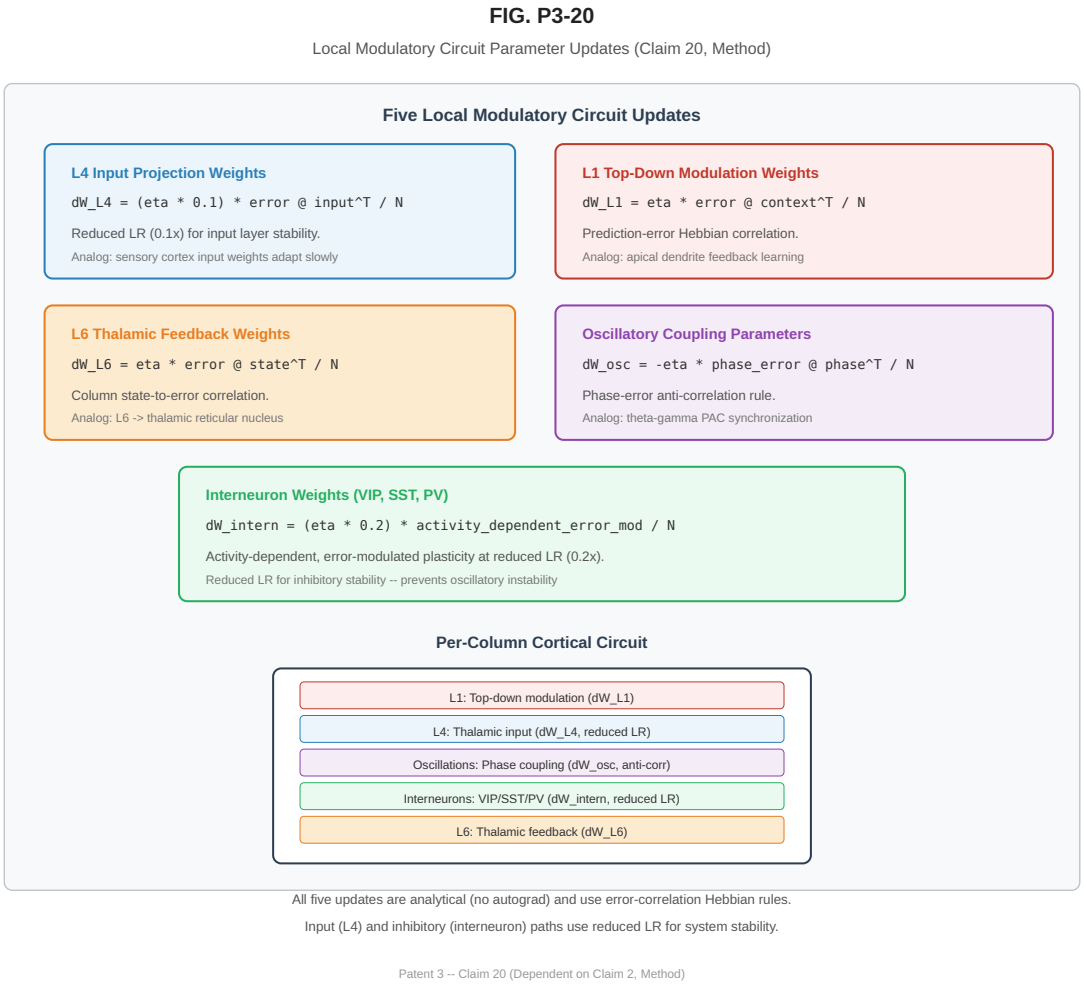
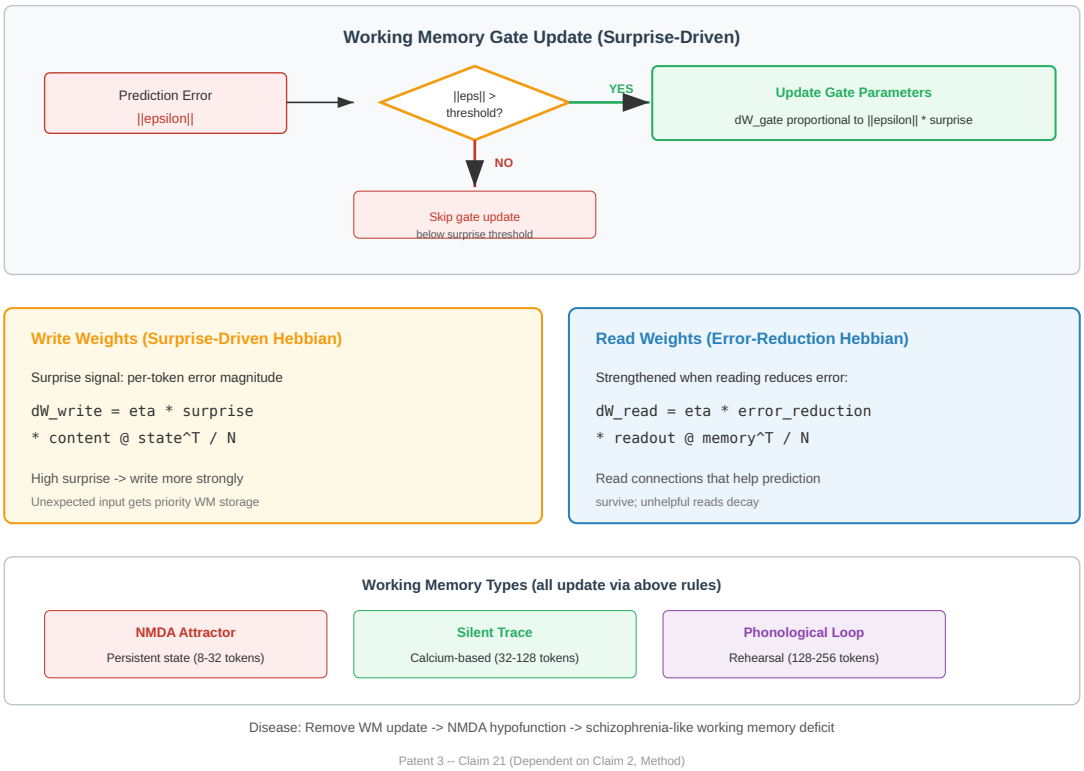


Figure 23: Local modulatory circuit parameter updates: L4, L1, L6, oscillatory coupling, interneurons (Claim 20).

FIG. P3-21

Working Memory Gate and Weight Updates (Claim 21, Method)



Disease: Remove WM update -> NMDA hypofunction -> schizophrenia-like working memory deficit

Patent 3 -- Claim 21 (Dependent on Claim 2, Method)

Figure 24: Working memory gate and weight updates driven by prediction error surprise (Claim 21).

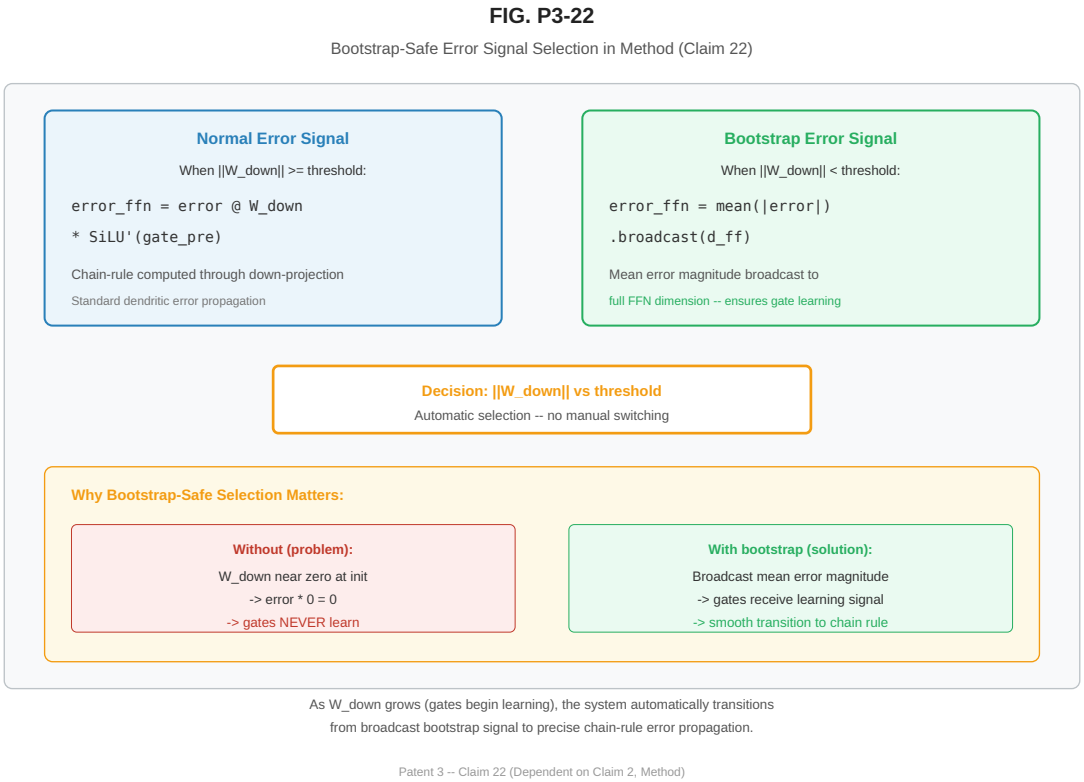


Figure 25: Bootstrap-safe error signal selection method (Claim 22).

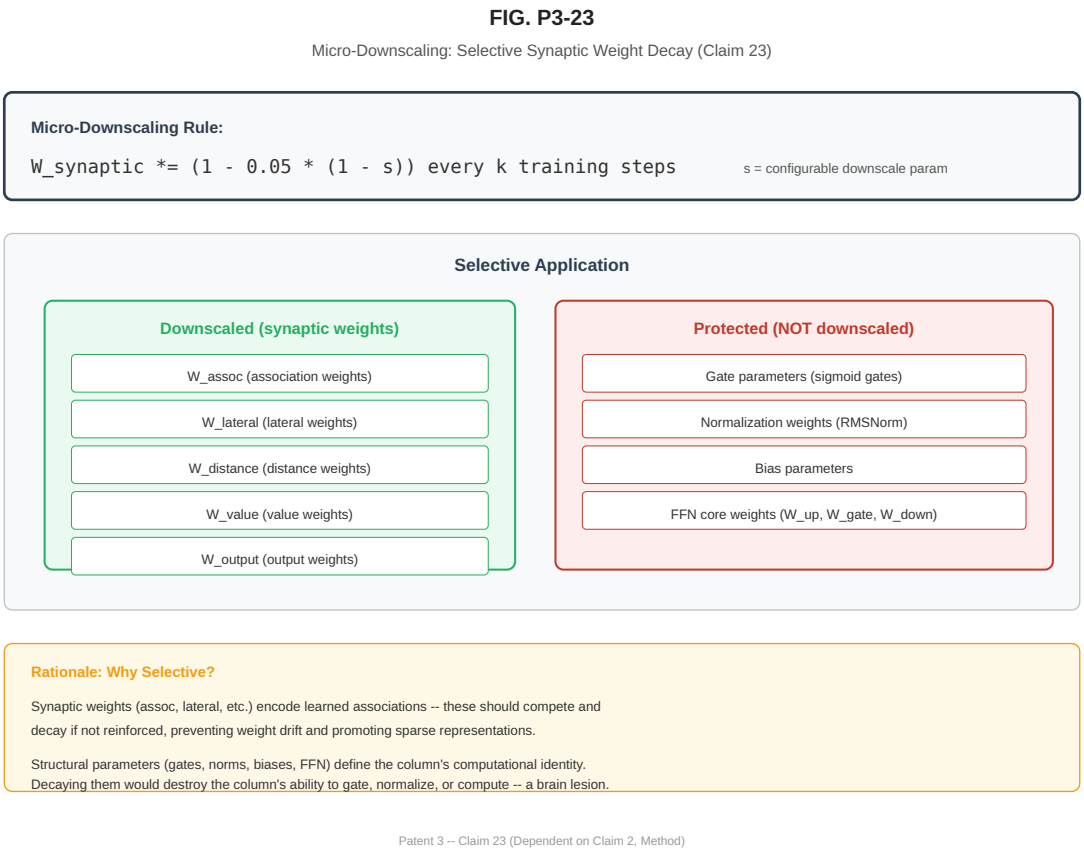


Figure 26: Micro-downscaling: selective synaptic weight decay with protected parameters (Claim 23).

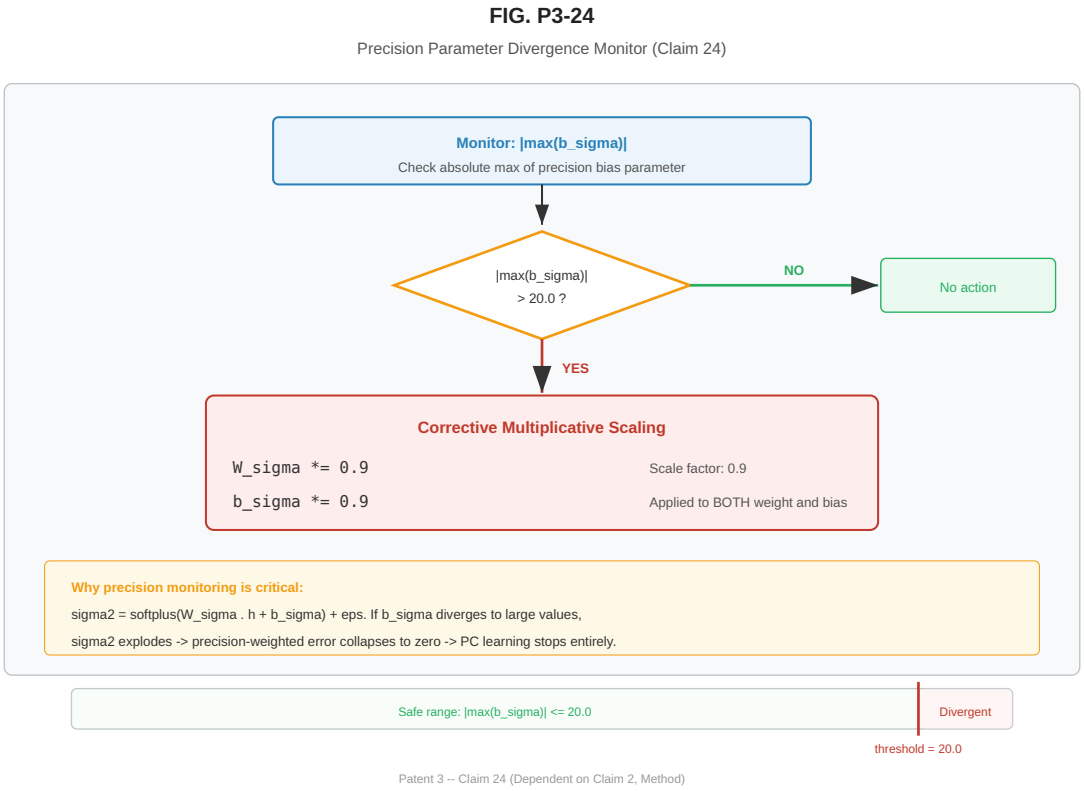


Figure 27: Precision parameter divergence monitor with corrective scaling (Claim 24).

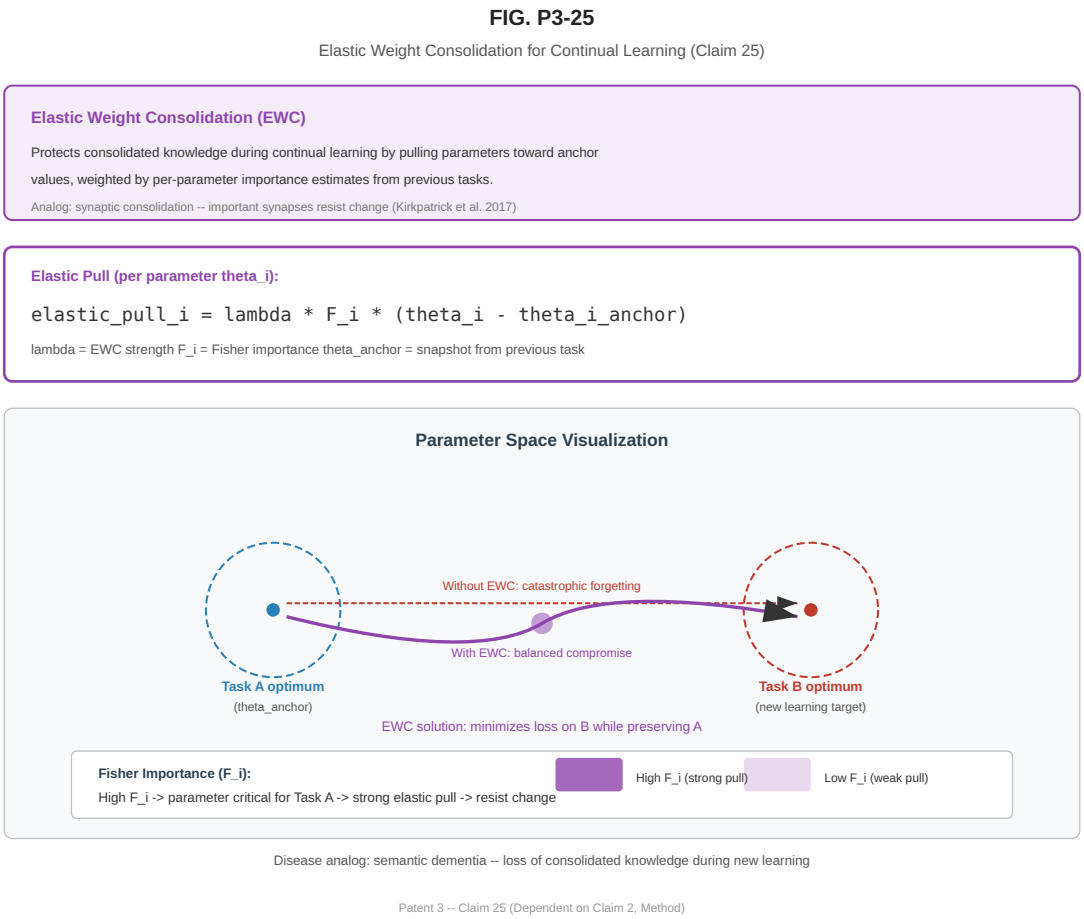


Figure 28: Elastic weight consolidation for continual learning protection (Claim 25).

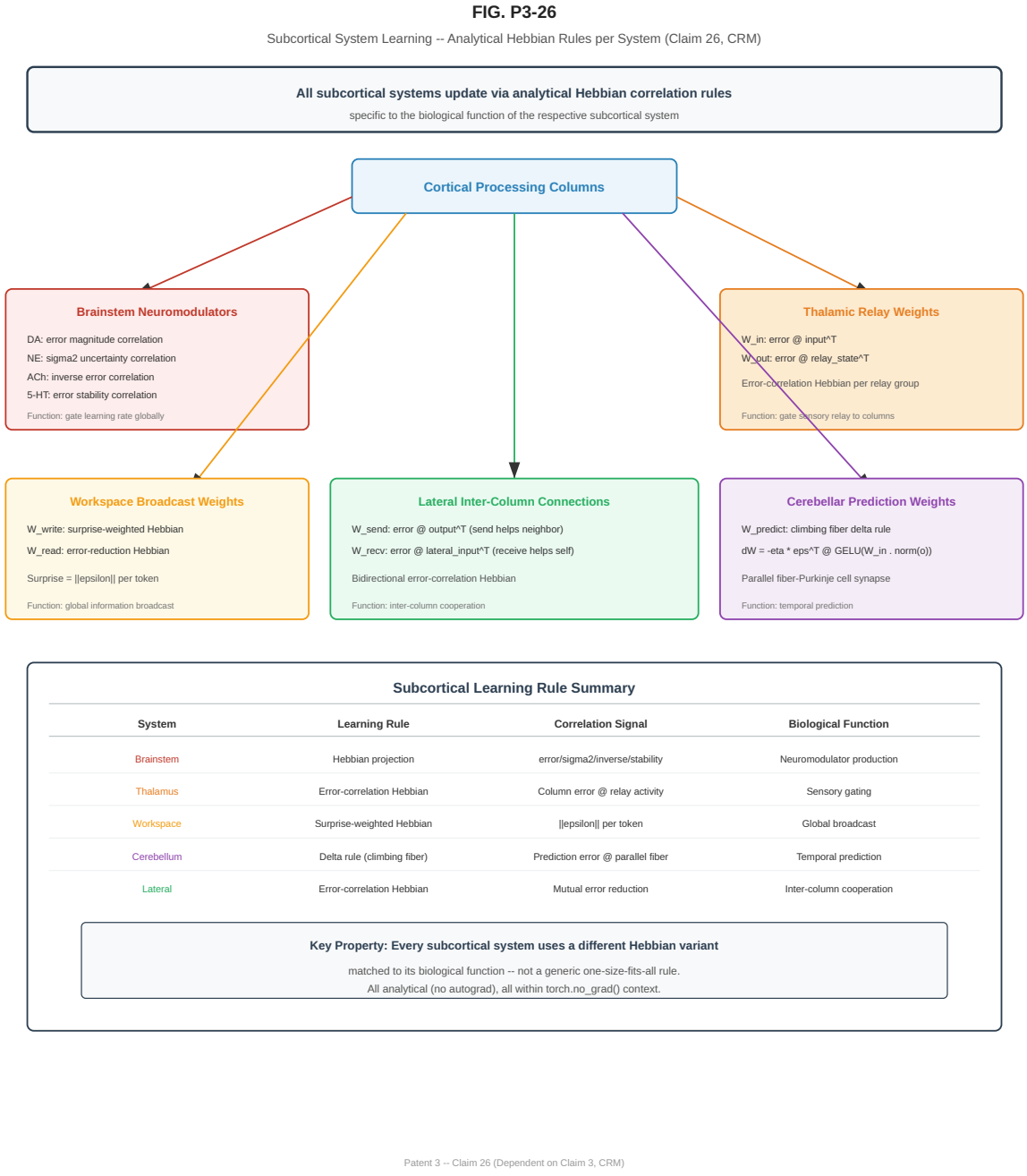


Figure 29: Subcortical system learning: analytical Hebbian rules for all five subsystems (Claim 26).

7. DETAILED DESCRIPTION

The following detailed description discloses the preferred embodiment of the present invention. Reference is made to the accompanying drawings, which form a part hereof, and in which are shown by way of illustration specific embodiments in which the invention may be practiced.

7.1. System Overview

The disclosed system comprises a learning manager module that orchestrates ten distinct local learning mechanisms operating on a hierarchical network of processing columns (cortical columns) and associated subcortical processing modules. The learning manager executes a wake-step procedure on every training iteration, applying learning rules in a prescribed sequence. All weight updates are computed under `torch.no_grad()` — meaning the automatic differentiation graph is never constructed, never stored, and never traversed. This eliminates the approximately three-fold memory overhead that backpropagation imposes for storing intermediate activations and gradient tensors.

The ten learning mechanisms are:

1. Precision-weighted predictive coding optimizer
2. BCM Hebbian optimizer with anti-Hebbian decorrelation and error modulation
3. Neuromodulatory plasticity gate
4. Brainstem projection learning
5. Subcortical Hebbian learning
6. L5 dendritic backpropagation
7. Reward-modulated three-factor learning
8. Micro-downscaling (continuous homeostasis)
9. Sigma-squared emergency guard
10. Local output correction (delta rule on language model head)

7.2. Learning Mechanism 1: Precision-Weighted Predictive Coding Optimizer

Referring to Figure 7, the predictive coding optimizer implements local error minimization per processing column. The system operates on a hierarchical stack of L processing columns, where each column l receives input x_l from the column below and generates a prediction $\hat{x}_l$ of that input.

7.2.1. Prediction Error Computation

For each column l, the prediction error is computed as:

$$\epsilon_l = x_l^{\text{detach}} - \hat{x}_l \tag{5}$$

where the superscript "detach" indicates that the target is detached from the computation graph, enforcing locality of learning. This detachment ensures that gradient information does not flow between columns through the error signal.

7.2.2. Precision-Weighted Loss

The local loss for column l is:

$$\mathcal{L}_l = \frac{||\epsilon_l||^2}{2\sigma_l^2} + \frac{1}{2}\log(\sigma_l^2) \tag{6}$$

where σ_l^2 is the column's precision parameter. The first term penalizes prediction error inversely weighted by precision (uncertain columns are penalized less for errors), and the second term prevents the trivial solution of infinite variance.

7.2.3. Precision Computation

The precision σ_l^2 is computed from the layer-normalized hidden state h_{norm} via:

$$z = \mathbf{W}_\sigma \cdot h_{\text{norm}} + b_\sigma, \quad \sigma_l^2 = \text{softplus}(z) + \varepsilon \tag{7}$$

where $\mathbf{W}_\sigma$ is a linear projection from d_{col} to 1, b_σ is a bias parameter, and $\varepsilon = 10^{-6}$ prevents division by zero. The softplus function ensures $\sigma^2 > 0$ without hard clamping.

7.2.4. Analytical Weight Updates for $\mathbf{W}_{predict}$

The weight update for the prediction matrix is derived analytically:

$$\Delta\mathbf{W}_{\text{predict}} = -\frac{\eta_{\text{PC}} \cdot p}{N}\left(\frac{\epsilon_l}{\sigma_l^2}\right) \cdot h_{\text{norm}}^\top \tag{8}$$

where η_{PC} is the predictive coding learning rate, p is the neuromodulatory plasticity multiplier, and $N = B \times T$ is the number of tokens in the batch. The computation proceeds as follows:

1. The weighted error ϵ_l/σ_l^2 is computed in float32 precision, producing a tensor of shape $[B, T, d_{\text{model}}]$.
2. Both the weighted error and hidden state are reshaped to $[N, d_{\text{model}}]$ and $[N, d_{\text{col}}]$ respectively.
3. The outer product $\Delta W = -(\text{weighted_error}^\top \cdot h_{\text{flat}})/N$ yields the gradient of shape $[d_{\text{model}}, d_{\text{col}}]$.
4. Gradient norm clipping is applied: if $||\Delta W|| > \text{max_grad_norm}$, then $\Delta W \leftarrow \Delta W \cdot (\text{max_grad_norm}/||\Delta W||)$.
5. The weight is updated: $\mathbf{W}_{\text{predict}} \leftarrow \mathbf{W}_{\text{predict}} - \eta \cdot \Delta W$.

7.2.5. Analytical Weight Updates for $\mathbf{W}_\sigma$

The precision parameter update is computed via the chain rule through softplus:

$$\frac{\partial \mathcal{L}}{\partial \sigma^2} = -\frac{\text{MSE}(\epsilon_l)}{2(\sigma_l^2)^2} + \frac{1}{2\sigma_l^2} \tag{9}$$

$$\frac{\partial \sigma^2}{\partial z} = \text{sigmoid}(z) \tag{10}$$

$$\frac{\partial \mathcal{L}}{\partial z} = \frac{\partial \mathcal{L}}{\partial \sigma^2} \cdot \text{sigmoid}(z) \tag{11}$$

where $\text{MSE}(\epsilon_l) = ||\epsilon_l||^2 / d_{\text{model}}$ uses mean (not sum) squared error to keep the gradient scale independent of model dimension. The sigmoid is clamped to $[0.01, 0.99]$ for numerical stability. The weight and bias updates are:

$$\Delta \mathbf{W}_\sigma = \frac{1}{N} (\partial \mathcal{L} / \partial z)^\top \cdot h_{\text{norm}} \tag{12}$$

$$\Delta b_\sigma = \frac{1}{N} \sum_n \partial \mathcal{L} / \partial z_n \tag{13}$$

7.3. Learning Mechanism 2: BCM Hebbian Optimizer with Anti-Hebbian Decorrelation

Referring to Figure 8, the Hebbian optimizer operates on the lateral association layer (L2/3) of each processing column, implementing self-organizing representation learning.

7.3.1. BCM-Modulated Hebbian Learning

The association weight matrix $\mathbf{W}_{\text{assoc}}$ is a per-head matrix of shape $[H, d_h, d_h]$ where H is the number of association heads and d_h is the head dimension. The BCM update rule is:

$$\Delta \mathbf{W}_{\text{assoc}}^{(h)} = \frac{\eta_{\text{Hebb}} \cdot m_{\text{meta}} \cdot s_{\sigma^2}}{N} \cdot (\text{bcm_mod}^\top \cdot \text{pre}) \tag{14}$$

where:

- $\text{bcm_mod} = \text{post} \odot (\text{post} - \sqrt{\theta_{\text{BCM}}})$ is the BCM modulation, with potentiation when post-synaptic activity exceeds $\sqrt{\theta_{\text{BCM}}}$ and depression when below.
- m_{meta} is the metaplasticity learning rate multiplier, which adapts based on the history of weight changes.
- $s_{\sigma^2} = 1 + \min(\overline{\sigma^2}, 10)$ is the sigma-squared-adaptive multiplier, which increases learning rate for columns with high prediction uncertainty.
- The diagonal is zeroed after each update to prevent self-reinforcement.
- Structural plasticity masks are applied to respect pruned connections.

- Weight norm is clamped to a maximum of 5.0 per head.

7.3.2. Error-Modulated Hebbian Learning

When prediction errors are available, the BCM update is blended with error-driven modulation:

$$\text{modulated} = \text{bcm_mod} \odot \left((1 - w_e) + w_e \cdot \frac{||\epsilon||}{E[||\epsilon||]} \right) \tag{15}$$

where $w_e \in [0, 1]$ is a configurable error weight parameter. At $w_e = 0$, learning is pure BCM (blind correlation). At $w_e = 1$, learning is fully error-driven. The normalized error magnitude ensures scale invariance.

7.3.3. Anti-Hebbian Lateral Decorrelation

The lateral weight matrix $\mathbf{W}_{\text{lateral}}$ is updated via anti-Hebbian learning:

$$\Delta \mathbf{W}^{(h)}_{\text{lateral}} = \frac{\eta}{N}(Y_h^\top Y_h - I) \tag{16}$$

where Y_h is the output of head h reshaped to $[N, d_h]$. The subtraction of the identity matrix drives outputs toward orthogonality, preventing representational collapse. Weight norm is clamped to 3.0.

7.3.4. Sigma-Squared Adaptive Learning Rate

The effective learning rate for each column is modulated by prediction uncertainty:

$$\eta_{\text{eff}} = \eta_{\text{Hebb}} \cdot m_{\text{meta}} \cdot (1 + \min(\overline{\sigma^2}, 10)) \tag{17}$$

Columns with high σ^2 (high uncertainty) learn faster because they have the most to learn, while columns with low σ^2 (well-calibrated predictions) learn more conservatively.

7.3.5. BCM Threshold Update

The BCM threshold θ_{BCM} is updated as an exponential moving average of post-synaptic activity:

$$\theta_{\text{BCM}} \leftarrow (1 - \alpha) \cdot \theta_{\text{BCM}} + \alpha \cdot \overline{|y|}^2 \tag{18}$$

This sliding threshold is the mechanism that prevents runaway Hebbian learning. High activity raises the threshold, making further potentiation harder (homeostatic regulation).

7.4. Learning Mechanism 3: Neuromodulatory Plasticity Gate

Referring to Figure 10, the neuromodulatory plasticity gate computes a multiplicative signal that gates all learning in the system.

7.4.1. Plasticity Computation

The plasticity gate is computed from four neuromodulator signals:

$$\text{plasticity} = \text{DA} \times \text{ACh} \times \underbrace{(0.5 + 0.5 \cdot \text{NE})}_{f_{\text{NE}}} \times \underbrace{(0.5 + 0.5 \cdot \text{5-HT})}_{f_{\text{5-HT}}} \tag{19}$$

where:

- **DA (dopamine)**: gates *whether* to learn. DA is high when prediction error is high (surprise), signaling that the current input deviates from expectations and should drive learning.
- **ACh (acetylcholine)**: gates *what* to learn. ACh is high during focused attention, selecting which synapses are eligible for modification.
- **NE (norepinephrine)**: modulates *how broadly* to learn. High NE broadens learning (exploration); low NE focuses learning (exploitation). The factor $f_{\text{NE}} \in [0.5, 1.0]$ ensures learning never drops below half rate.
- **5-HT (serotonin)**: modulates *temporal horizon*. High 5-HT extends credit assignment over longer sequences; low 5-HT focuses on immediate outcomes. The factor $f_{\text{5-HT}} \in [0.5, 1.0]$ operates analogously to NE.

At neutral neuromodulatory state (all signals at 0.5), the plasticity gate produces $0.5^2 \times 0.75 \times 0.75 \approx 0.14$, providing a moderate baseline learning rate.

7.4.2. Developmental Modulation

The plasticity gate is further modulated by a developmental stage multiplier, which implements critical periods in the network's learning:

$$\text{plasticity}_{\text{eff}} = \text{plasticity} \times m_{\text{dev}} \tag{20}$$

where m_{dev} varies across training according to a developmental program that controls learning rate, Hebbian weighting, inhibitory strength, and structural plasticity rate.

7.5. Learning Mechanism 4: Brainstem Projection Learning

Referring to Figure 11, the brainstem module produces neuromodulatory signals from learned linear projections of the current representation. These projections learn via analytical Hebbian updates:

7.5.1. DA Projection

The dopamine projection W_{DA} learns to correlate with prediction error magnitude:

$$\Delta W_{\text{DA}} = \frac{\eta_{\text{BS}}}{N} \cdot ||\epsilon||^{\top} \cdot h_{\text{flat}} \tag{21}$$

where $||\epsilon||$ is the per-token error magnitude $[N, 1]$ and h_{flat} is the last column's output $[N, d]$.

7.5.2. NE Projection

The norepinephrine projection learns to correlate with prediction uncertainty:

$$\Delta W_{\text{NE}} = \frac{0.5 \cdot \eta_{\text{BS}}}{N} \cdot \sigma^{2\top} \cdot h_{\text{flat}} \tag{22}$$

7.5.3. ACh Projection

The acetylcholine projection learns to correlate with inverse error (confidence):

$$\Delta W_{\text{ACh}} = \frac{0.5 \cdot \eta_{\text{BS}}}{N} \cdot \left(\frac{1}{1 + ||\epsilon||} \right)^{\top} \cdot h_{\text{flat}} \tag{23}$$

7.5.4. 5-HT Projection

The serotonin projection learns to correlate with error temporal stability:

$$\Delta W_{\text{5-HT}} = \frac{0.3 \cdot \eta_{\text{BS}}}{N} \cdot \underbrace{\frac{1}{1 + \text{std}(||\epsilon||)/(\text{mean}(||\epsilon||) + \varepsilon)}}_{\text{stability}} \cdot \bar{h} \tag{24}$$

where $\bar{h}$ is the mean hidden representation. All projection weights are clamped to $[-1, 1]$.

7.6. Learning Mechanism 5: Subcortical Hebbian Learning

Referring to Figure 29, each subcortical system uses a biologically grounded local learning rule.

7.6.1. Thalamic Relay Learning

For each thalamic relay group serving a subset of columns, the relay input weight W_{in} and output weight W_{out} are updated via error-correlation Hebbian:

$$\Delta W_{\text{out}} = -\frac{\eta_{\text{thal}}}{N} \cdot \epsilon_{\text{relay}}^{\top} \cdot h_{\text{relay}} \tag{25}$$

$$\Delta W_{\text{in}} = -\frac{0.5 \cdot \eta_{\text{thal}}}{N} \cdot h_{\text{relay}}^{\top} \cdot \epsilon_{\text{relay}} \tag{26}$$

where ϵ_{relay} is the aggregate error for the relay's columns and $h_{\text{relay}} = W_{\text{in}}(\bar{o}_{\text{columns}})$. The relay gate opens bidirectionally based on aggregate error magnitude.

7.6.2. Cortical Workspace Learning

The workspace write weights learn via surprise-weighted Hebbian:

$$\Delta W_{\text{write}}^{(c)} = \frac{\eta_{\text{ws}}}{N} \cdot (\text{surprise} \odot o_c)^\top \cdot h_{\text{ws}} \tag{27}$$

where surprise $= ||\epsilon_c||$ per token. The workspace read weights learn via error-reduction Hebbian:

$$\Delta W_{\text{read}}^{(c)} = -\frac{\eta_{\text{ws}}}{N} \cdot \epsilon_c^\top \cdot h_{\text{ws}} \tag{28}$$

7.6.3. Cerebellar Forward Model Learning

The cerebellum implements the Marr–Albus–Ito learning rule: the climbing fiber carries prediction error, and the parallel fiber–Purkinje cell synapse is modified via the delta rule:

$$\Delta W_{\text{predict}} = -\frac{\eta_{\text{cb}}}{N} \cdot \epsilon^\top \cdot \text{GELU}(W_{\text{in}} \cdot \text{RMSNorm}(o)) \tag{29}$$

$$\Delta W_{\text{error}} = -\frac{\eta_{\text{cb}}}{N} \cdot \epsilon^\top \cdot o \tag{30}$$

7.6.4. Lateral Inter-Column Learning

For each column i and its neighbors $\{i-1, i+1\}$:

$$\Delta W_{\text{send}}^{(j)} = -\frac{\eta_{\text{lat}}}{N} \cdot h_{\text{send}}^\top \cdot \epsilon_i \tag{31}$$

$$\Delta W_{\text{recv}}^{(i)} = -\frac{\eta_{\text{lat}}}{N} \cdot \epsilon_i^\top \cdot h_{\text{send}} \tag{32}$$

where $h_{\text{send}} = W_{\text{send}}^{(j)}(o_j)$ is the neighboring column's projected output.

7.7. Learning Mechanism 6: L5 Dendritic Backpropagation

Referring to Figure 17, the L5 feedforward network within each column uses a SwiGLU nonlinearity: $\text{ff} = \text{SiLU}(W_{\text{gate}} \cdot h) \odot (W_{\text{up}} \cdot h)$, followed by $o = W_{\text{down}} \cdot \text{ff}$. Top-down error from the column above propagates analytically:

7.7.1. Error Signal Selection

For column l, the teaching signal is the prediction error from column $l+1$ (the column above), telling column l how predictable its output was. For the last column, its own error is used. The error is precision-weighted: $\delta_{\text{out}} = \epsilon/\sigma^2$.

7.7.2. Analytical Chain Rule Through SwiGLU

$$\Delta W_{\text{down}} = -\frac{\eta}{N} \cdot \delta_{\text{out}}^{\top} \cdot h_{\text{ff}} \tag{33}$$

$$\delta_{\text{ff}} = \delta_{\text{out}} \cdot W_{\text{down}} \tag{34}$$

$$\text{SiLU}'(g) = \text{sigmoid}(g) \cdot (1 + g \cdot (1 - \text{sigmoid}(g))) \tag{35}$$

$$\delta_{\text{gate}} = \delta_{\text{ff}} \odot u \odot \text{SiLU}'(g) \tag{36}$$

$$\delta_{\text{up}} = \delta_{\text{ff}} \odot \text{SiLU}(g) \tag{37}$$

$$\Delta W_{\text{gate}} = -\frac{\eta}{N} \cdot \delta_{\text{gate}}^{\top} \cdot h_{\text{norm}} \tag{38}$$

$$\Delta W_{\text{up}} = -\frac{\eta}{N} \cdot \delta_{\text{up}}^{\top} \cdot h_{\text{norm}} \tag{39}$$

where g is the gate projection, u is the up projection, and h_{norm} is the input.

7.7.3. Dendritic Gate Updates

The basal dendritic gate W_{dend} modulates SwiGLU output based on feedforward input (L4 output). The apical dendritic gate W_{apical} modulates based on top-down context. Both learn via:

$$\Delta W_{\text{dend}} = -\frac{\eta_{\text{dend}}}{N} \cdot (\delta_{\text{dend}} \odot \text{ff}_{\text{pre}} \odot \sigma'(\text{dend_logit}))^{\top} \cdot \text{l4_out} \tag{40}$$

with an analogous expression for W_{apical} using the top-down signal.

7.7.4. Bootstrap-Safe Error Signal

Early in training, W_{down} is near zero (identity-at-init), making $\delta_{\text{ff}} \approx 0$ via normal chain rule. The system detects this condition by checking $||W_{\text{down}}||$ and, when below threshold 0.01, substitutes a bootstrap error signal: the mean error magnitude per token broadcast to the feedforward dimension. This ensures dendritic gates receive a learning signal even before the main pathway has learned.

7.8. Learning Mechanism 7: Reward-Modulated Three-Factor Learning

The system supports reward-modulated Hebbian learning for hippocampal memory:

$$\Delta w = \text{pre} \times \text{post} \times \text{reward} \tag{41}$$

Positive reward boosts hippocampal surprise signals, strengthening encoding of rewarding experiences. Negative reward suppresses surprise, reducing encoding of aversive outcomes. This three-factor rule requires only locally available signals: pre-synaptic activity, post-synaptic activity, and a globally broadcast reward signal (dopamine).

7.9. Learning Mechanism 8: Micro-Downscaling

Referring to Figure 26, the system implements continuous homeostatic regulation via micro-downscaling applied every 10 training steps:

$$W \leftarrow W \times (1 - 0.05 \times (1 - s_{\text{downscale}})) \tag{42}$$

where $s_{\text{downscale}}$ is a configurable synaptic downscale parameter. Critically, downscaling is applied selectively:

- **Downscaled**: W_{assoc}, W_{lateral}, W_{dist}, W_{val}, W_{out} — these are synaptic weights that are subject to Hebbian potentiation and require homeostatic regulation.
- **Protected**: gate parameters, normalization weights, biases, decay time constants — these control circuit dynamics and must not be attenuated.

A weight is only downscaled if its absolute maximum exceeds 10^{-8}, preventing downscaling of weights that are effectively zero.

7.10. Learning Mechanism 9: Sigma-Squared Emergency Guard

The system monitors precision parameters for divergence:

$$\text{if } \max(|b_\sigma|) > 20: \quad b_\sigma \leftarrow 0.9 \cdot b_\sigma, \quad \mathbf{W}_\sigma \leftarrow 0.9 \cdot \mathbf{W}_\sigma \tag{43}$$

This guard detects the pathological condition where precision parameters grow unbounded, which would cause the precision-weighted prediction error to approach either zero (infinite confidence, ignoring errors) or infinity (zero confidence, amplifying noise). The multiplicative correction of 0.9 gently reduces the parameter toward a safe range.

7.11. Learning Mechanism 10: Local Output Correction

Referring to Figure 19, the local output correction implements the analytical delta rule for the language model head.

7.11.1. Analytical Cross-Entropy Gradient

For a linear language model head logits $= W_{\text{head}} \cdot h + b$, the gradient of cross-entropy loss with respect to the logits is:

$$\text{error} = \text{softmax}(\text{logits}) - \text{one_hot}(\text{targets}) \tag{44}$$

This is computed directly without autograd. Padding tokens (target $= -1$) are masked with a valid token mask.

7.11.2. Output Bias Update

$$\Delta b_{\text{out}} = -\frac{\eta_{\text{LOC}}}{N_{\text{valid}}} \sum_n \text{error}_n \tag{45}$$

with gradient norm clipping: if $||\Delta b|| > 1.0$, then $\Delta b \leftarrow \Delta b \cdot (1.0/||\Delta b||)$.

7.11.3. Low-Rank Production Pathway

The production pathway uses a low-rank decomposition $\Delta W = \Delta A \cdot \Delta B$ where $\Delta A \in \mathbb{R}^{V \times r}$ and $\Delta B \in \mathbb{R}^{r \times D}$:

$$h_{\text{proj}} = h \cdot \Delta B^\top \tag{46}$$

$$\Delta(\Delta A) = -\frac{\eta_{\text{LOC}}}{N_{\text{valid}}} \cdot \text{error}^\top \cdot h_{\text{proj}} \tag{47}$$

$$\text{error}_{\text{proj}} = \text{error} \cdot \Delta A \tag{48}$$

$$\Delta(\Delta B) = -\frac{\eta_{\text{LOC}}}{N_{\text{valid}}} \cdot \text{error}_{\text{proj}}^\top \cdot h \tag{49}$$

Both updates are gradient-norm clipped to MAX_GRAD_NORM $= 1.0$.

7.12. Local Parameter Updates for Modulatory Circuits

In addition to the ten primary mechanisms, the Hebbian optimizer provides local updates for parameters in modulatory circuits that do not participate in the main prediction pathway:

7.12.1. L4 Input Projection

The thalamic input layer L4 learns at $\eta_{\text{L4}} = 0.05 \times \eta_{\text{base}}$ (very slow for input layer stability):

$$\Delta W_{\text{L4}} = -\frac{\eta_{\text{L4}}}{N} \cdot (W_{\text{L4}} \cdot \epsilon)^\top \cdot x_{\text{input}} \tag{50}$$

7.12.2. L1 Top-Down Modulation

The top-down projection L1 learns via prediction-error Hebbian:

$$\Delta W_{\text{L1}} = -\frac{\eta}{N} \cdot (W_{\text{L4}} \cdot \epsilon)^\top \cdot \epsilon \tag{51}$$

The L1 gate opens bidirectionally based on error magnitude, with a threshold centered at 1.0.

7.12.3. L6 Thalamic Feedback

The feedback projection L6 learns column state-to-error correlation:

$$\Delta W_{\text{L6}} = -\frac{\eta}{N} \cdot \epsilon^\top \cdot s \tag{52}$$

where s is the column state vector. The L6 gate is bidirectionally driven analogously to L1.

7.12.4. Oscillation Phase-Amplitude Coupling

Oscillatory coupling parameters (κ_1, κ_2, κ_3) learn via phase-error anti-correlation: coupling is reduced where error aligns with oscillatory phase, and the oscillation output projection $W_{\text{osc_out}}$ learns via error-correlation Hebbian.

7.12.5. Interneuron Weights

VIP interneurons learn to detect novelty (error-modulated activity), SST interneurons learn input suppression (inverse-error modulated), and PV interneurons learn veto thresholds (error-weighted activity). All at very slow learning rates ($0.05 \times \eta_{\text{base}}$) for inhibitory stability.

7.12.6. Working Memory Gates and Weights

When prediction error exceeds a threshold, working memory gates (NMDA, silent trace, phonological loop) open in proportion to surprise. Write weights learn via surprise-driven Hebbian correlation, and read weights learn via error-reduction Hebbian. All working memory weights are norm-guarded to 5.0.

7.13. Coordination and Sequencing

The learning manager applies all mechanisms in a specific sequence during each wake step:

1. Developmental modulation (critical period adjustment)
2. Neuromodulatory plasticity gate computation
3. Predictive coding updates (vertical connections)
4. L5 dendritic backpropagation (intra-column deep pathway)
5. Hebbian updates (lateral L2/3 associations)
6. Local parameter updates (L1/L4/L6/oscillations/interneurons/WM)
7. Structural plasticity (pruning and synaptogenesis)
8. Hebbian weight norm guards
9. EWC elastic pull (continual learning protection)
10. Micro-downscaling (homeostatic regulation)
11. Brainstem projection learning

12. Subcortical weight learning
13. Sigma-squared emergency guard
14. Wake step tracking

This ordering ensures that neuromodulatory gating is computed before it modulates learning, prediction errors are computed before they drive Hebbian and subcortical updates, and homeostatic regulation follows potentiation.

8. EXAMPLES

8.1. Example 1: Predictive Coding Update Execution

Consider a network with 12 processing columns, batch size $B = 96$, sequence length $T = 512$, and model dimension $d = 512$. At training step 100:

1. Column 5 generates prediction $\hat{x}_5$ of its input x_5.
2. Prediction error: $\epsilon_5 = x_5 - \hat{x}_5$, shape $[96, 512, 512]$.
3. Precision: $\sigma_5^2 = \text{softplus}(\mathbf{W}_\sigma \cdot h_{\text{norm}}) + 10^{-6}$, shape $[96, 512, 1]$. Suppose $\sigma_5^2 \approx 2.3$ (the column is moderately uncertain).
4. Weighted error: $\epsilon_5/2.3$, shape $[96, 512, 512]$.
5. Reshape to $[49152, 512]$ for outer product computation.
6. $\Delta\mathbf{W}_{\text{predict}} = -(49152^{-1})(\epsilon/\sigma^2)^\top h_{\text{norm}}$, shape $[512, d_{\text{col}}]$.
7. Gradient norm: suppose $||\Delta W|| = 3.7 > 1.0$. Clip: $\Delta W \leftarrow \Delta W \times (1.0/3.7)$.
8. Update: $\mathbf{W}_{\text{predict}} \leftarrow \mathbf{W}_{\text{predict}} - 0.001 \times 0.14 \times \Delta W$ where 0.14 is the plasticity gate.

The entire computation uses no autograd graph. Memory consumption is proportional to the tensors being manipulated, not to the full network depth.

8.2. Example 2: BCM Hebbian with Error Modulation

Column 3 performs error-modulated Hebbian learning with $w_e = 0.5$:

1. Pre-synaptic input: x_{pre}, shape $[96, 512, 512]$. Reshape to heads: $[96, 8, 512, 64]$.
2. Post-synaptic output: y_{post}, same shape.
3. BCM threshold: $\theta_{\text{BCM}} = 0.82$ (running average of squared activity).

4. BCM modulation: $y \odot (y - \sqrt{0.82})$.
5. Error modulation: error_mag $= ||\epsilon_3||_2$ per token. Normalize to mean 1.
6. Blended: bcm_mod $\odot\, (0.5 + 0.5 \times$ error_scale$)$.
7. Update per head: $\Delta W^{(h)} = (\text{modulated}^\top \cdot \text{pre})/N$.
8. Zero diagonal, apply structural plasticity mask, clamp norm to 5.0.

Simultaneously, anti-Hebbian update: $\Delta \mathbf{W}^{(h)}_{\text{lateral}} = (Y_h^\top Y_h - I)/N$, clamped to norm 3.0.

8.3. Example 3: Neuromodulatory Gating During Surprising Input

At step 500, the network encounters an unexpected token sequence:

1. Brainstem computes: DA $= 0.87$ (high surprise $\rightarrow$ high dopamine), NE $= 0.71$ (elevated arousal), 5-HT $= 0.35$ (short patience — predictions failing), ACh $= 0.62$ (moderate attention).
2. Plasticity gate: $0.87 \times 0.62 \times (0.5 + 0.5 \times 0.71) \times (0.5 + 0.5 \times 0.35) = 0.87 \times 0.62 \times 0.855 \times 0.675 \approx 0.312$.
3. This is more than double the neutral baseline of ≈ 0.14, meaning all learning rules fire with increased strength for this surprising input.
4. Brainstem DA projection weight increases (correlating with the high error magnitude).
5. Brainstem 5-HT projection decreases (the instability of error reduces the serotonin stability signal).

8.4. Example 4: L5 Dendritic Backpropagation with Bootstrap

At training step 10 (early training), column 7:

1. W_{down} was initialized to zeros. $||W_{\text{down}}|| = 0.003 < 0.01$ (bootstrap threshold).
2. Normal chain rule path: $\delta_{\text{ff}} = \delta_{\text{out}} \cdot W_{\text{down}} \approx 0$ — dendritic gates would receive no learning signal.
3. Bootstrap fallback: $\delta_{\text{dend}} = ||\delta_{\text{out}}||_2$ per token, broadcast to d_{ff}.
4. Dendritic gate receives a meaningful error signal from step 1, enabling early learning.
5. At step 200, suppose $||W_{\text{down}}|| = 0.15 > 0.01$: the system switches to normal chain rule $\delta_{\text{ff}} = \delta_{\text{out}} \cdot W_{\text{down}}$, which is now non-trivial.

8.5. Example 5: Local Output Correction for Language Model Head

At step 300, the LM head processes 49,152 tokens over a vocabulary of 13,817:

1. Forward: logits $= W_{\text{emb}} \cdot h + \Delta A \cdot \Delta B \cdot h + b_{\text{out}}$, shape $[49152, 13817]$.
2. probs $=$ softmax(logits), computed in float32.
3. one_hot: scatter targets into one-hot, shape $[49152, 13817]$.
4. error $=$ probs $-$ one_hot, shape $[49152, 13817]$.
5. Mask padding: error $\leftarrow$ error $\odot$ valid_mask.
6. Bias update: $\Delta b = -$error.sum(0)$/N_{\text{valid}}$, clip norm to 1.0.
7. ΔA update: project hidden through ΔB, compute outer product with error.
8. ΔB update: project error through ΔA, compute outer product with hidden.
9. All updates applied. Zero autograd. Memory: only the tensors being manipulated.

8.6. Example 6: Micro-Downscaling Selectivity

At step 100 (divisible by 10), with $s_{\text{downscale}} = 0.95$:

1. Scale factor: $1 - 0.05 \times (1 - 0.95) = 1 - 0.0025 = 0.9975$.
2. For each column: $W_{\text{assoc}} \leftarrow 0.9975 \times W_{\text{assoc}}$ (if max $|W| > 10^{-8}$).
3. Similarly for W_{lateral}, W_{dist}, W_{val}, W_{out}.
4. `gate_raw`: NOT downscaled (intrinsic excitability, not a synapse).
5. `norm.weight`: NOT downscaled (normalization scale).
6. W_{down}, W_{gate}, W_{up}: NOT downscaled (L5 SwiGLU core — updated by dendritic backpropagation, not Hebbian).

9. INDUSTRIAL APPLICABILITY

The present invention has broad industrial applicability in any domain where neural networks require training with reduced memory consumption, improved biological plausibility, or enhanced continual learning capability.

9.1. GPU Memory Reduction for Large Language Model Training

The elimination of global backpropagation reduces training VRAM consumption by approximately a factor of three, as no intermediate activations need to be stored for the backward pass and no gradient computation graph is maintained. For a 106 million parameter model, this reduces VRAM usage from approximately 6 GB (with backpropagation) to approximately 2 GB (with local learning only). This enables training of larger models on commodity GPU hardware, including laptop-class GPUs with 6–8 GB VRAM.

9.2. Edge Device Training

The local learning rules disclosed herein enable on-device training and adaptation without the memory overhead of backpropagation, making the system suitable for edge deployment on mobile devices, embedded systems, and IoT hardware where VRAM is severely constrained.

9.3. Continual Learning Systems

The combination of micro-downscaling, sigma-squared emergency guards, EWC elastic pull, and neuromodulatory gating provides inherent resistance to catastrophic forgetting, enabling deployment in continual learning scenarios where the model must adapt to new data without losing previously learned capabilities.

9.4. Neuromorphic Hardware Acceleration

The local learning rules disclosed herein are inherently compatible with neuromorphic computing architectures (e.g., Intel Loihi, IBM TrueNorth, BrainChip Akida), which implement local Hebbian-like learning rules in silicon. The system can serve as a software simulation of algorithms intended for eventual neuromorphic hardware deployment.

9.5. Adaptive Language Processing

The neuromodulatory plasticity gating enables context-dependent learning rates: the system learns rapidly from surprising inputs (high dopamine), slowly from predictable inputs, and modulates exploration versus exploitation (norepinephrine) and temporal horizon (serotonin) dynamically. This is applicable to dialogue systems, real-time translation, and adaptive content generation.

ABSTRACT

A system and method for training neural networks using exclusively local learning rules without global backpropagation. The system comprises a learning manager that orchestrates ten interoperating local learning mechanisms: (1) a precision-weighted predictive coding optimizer

computing analytical weight updates from per-column prediction errors normalized by learned precision estimates; (2) a BCM Hebbian optimizer with anti-Hebbian decorrelation and error modulation for self-organizing lateral associations; (3) a neuromodulatory plasticity gate multiplicatively combining dopamine, acetylcholine, norepinephrine, and serotonin signals to gate all learning; (4) brainstem projection learning via analytical Hebbian correlation with error signals; (5) subcortical Hebbian learning for thalamic relay, cortical workspace, cerebellar forward model, and lateral connections; (6) dendritic backpropagation within individual columns through SwiGLU nonlinearity with bootstrap-safe error signals; (7) reward-modulated three-factor learning; (8) micro-downscaling for continuous homeostatic regulation of synaptic weights; (9) sigma-squared emergency guards for precision parameter stabilization; and (10) analytical local output correction for language model heads using the delta rule with low-rank decomposition. All weight updates are computed analytically under `torch.no_grad()`, eliminating the gradient computation graph and reducing training memory consumption by approximately three-fold compared to backpropagation.

10. CLAIMS

We claim:

Claim 1. A system for training a neural network using local learning rules without global backpropagation, the system comprising:

(a) a hierarchical neural network comprising a plurality of processing columns arranged in a stack, each processing column comprising a prediction weight matrix, a precision weight matrix, an association weight matrix, a lateral weight matrix, and a feedforward network with a gated nonlinearity;

(b) a predictive coding optimizer configured to, for each processing column, compute a prediction error as the difference between the column's input and the column's prediction of that input, normalize the prediction error by a learned precision estimate computed via a softplus-transformed linear projection of the column's hidden state, and update the prediction weight matrix and the precision weight matrix using analytical gradient expressions computed without constructing or traversing an automatic differentiation graph;

(c) a Hebbian optimizer configured to update the association weight matrix of each processing column using a BCM-modulated Hebbian learning rule wherein the weight change is proportional to a post-synaptic activity modulated by a sliding threshold, and to update the lateral weight matrix using an anti-Hebbian decorrelation rule;

(d) a neuromodulatory plasticity gate configured to compute a multiplicative plasticity signal from four neuromodulator values representing dopamine, acetylcholine, norepinephrine, and serotonin, and to apply the plasticity signal as a multiplier to the learning rates of both the predictive coding optimizer and the Hebbian optimizer;

(e) a dendritic error propagation module configured to propagate error signals analytically through the gated nonlinearity of each processing column's feedforward network, computing weight updates for down-projection, gate-projection, and up-projection weights using the analytical chain rule through a SiLU activation derivative, without global backpropagation; and

(f) a local output correction module configured to compute an output error as the difference between a softmax of output logits and a one-hot encoding of target tokens, and to update output pathway weights using the delta rule applied to the output error and hidden representations, with gradient norm clipping, without constructing an automatic differentiation graph;

wherein all weight updates across the system are computed under a no-gradient context that prevents construction of a computation graph, thereby reducing training memory consumption compared to backpropagation-based training.

Claim 2. A method for training a neural network using local learning rules without global backpropagation, the method comprising:

(a) performing a forward pass through a hierarchical neural network comprising a plurality of processing columns to produce per-column predictions, precision estimates, and output logits;

(b) computing, for each processing column, a prediction error as the difference between the column's input and the column's prediction, and normalizing the prediction error by a precision estimate derived from a softplus-transformed projection of the column's hidden state;

(c) computing a neuromodulatory plasticity signal by multiplicatively combining four neuromodulator values, wherein a first neuromodulator gates whether learning occurs, a second neuromodulator gates what is learned, a third neuromodulator modulates learning breadth, and a fourth neuromodulator modulates temporal learning horizon;

(d) updating prediction weight matrices of the processing columns using analytically computed precision-weighted outer products of normalized errors and hidden states, modulated by the plasticity signal, without constructing an automatic differentiation graph;

(e) updating association weight matrices of the processing columns using a BCM-modulated Hebbian rule wherein weight changes are proportional to post-synaptic activity modulated by a sliding threshold, and updating lateral weight matrices using an anti-Hebbian decorrelation rule;

(f) propagating top-down error signals analytically through a gated feedforward nonlinearity within each processing column to compute weight updates for the feedforward network using the analytical chain rule through a SiLU activation derivative;

(g) computing an output error as the difference between a softmax distribution over output logits and a one-hot target distribution, and updating output pathway weights using the delta rule applied to the output error;

(h) applying micro-downscaling to synaptic weight matrices at periodic intervals while excluding gate parameters, normalization parameters, and bias parameters from downscaling; and

(i) monitoring precision parameters for divergence and applying corrective multiplicative scaling when a precision parameter exceeds a divergence threshold;

wherein all weight update steps are performed under a no-gradient context without constructing or traversing an automatic differentiation graph.

Claim 3. A non-transitory computer-readable medium storing instructions that, when executed by a processor coupled to a graphics processing unit, cause the processor to perform a method for training a neural network using local learning rules without global backpropagation, the method comprising:

(a) computing, for each processing column in a hierarchical neural network, a precision-weighted prediction error by dividing a prediction error tensor by a learned precision estimate tensor, and computing an analytical weight update as an outer product of the precision-weighted error and a hidden state, divided by a token count;

(b) computing a neuromodulatory plasticity gate as a product of a dopamine signal, an acetylcholine signal, and affine transformations of a norepinephrine signal and a serotonin signal, and applying the gate as a multiplicative modifier to all analytical weight updates;

(c) updating association weights via a BCM-modulated Hebbian rule with a sliding activity threshold and updating lateral weights via an anti-Hebbian decorrelation rule, both modulated by the neuromodulatory plasticity gate;

(d) propagating error signals through a SwiGLU feedforward nonlinearity within each column by computing a SiLU derivative as $\text{sigmoid}(g) \cdot (1 + g \cdot (1 - \text{sigmoid}(g)))$ and analytically computing weight updates for down-projection, gate-projection, and up-projection matrices;

(e) computing an output error as a softmax distribution minus a one-hot target distribution and updating output pathway weights via the delta rule with gradient norm clipping; and

(f) periodically applying homeostatic downscaling selectively to synaptic weight matrices while protecting gate, normalization, and bias parameters;

wherein all computations in steps (a) through (f) are performed within a no-gradient execution context that suppresses automatic differentiation graph construction.

Claim 4. The system of Claim 1, wherein the predictive coding optimizer computes the precision estimate as:

$$\sigma^2 = \text{softplus}(\mathbf{W}_\sigma \cdot h_{\text{norm}} + b_\sigma) + \varepsilon$$

and updates $\mathbf{W}_\sigma$ via the chain rule through softplus:

$$\Delta \mathbf{W}_\sigma = \frac{\eta}{N} \left(\frac{\partial \mathcal{L}}{\partial \sigma^2} \cdot \text{sigmoid}(z) \right)^\top \cdot h_{\text{norm}}$$

where $\frac{\partial \mathcal{L}}{\partial \sigma^2} = -\frac{\text{MSE}(\epsilon)}{2(\sigma^2)^2} + \frac{1}{2\sigma^2}$ and the sigmoid is clamped to $[0.01, 0.99]$.

Claim 5. The system of Claim 1, wherein the BCM-modulated Hebbian learning rule computes weight changes as:

$$\Delta \mathbf{W}_{\text{assoc}}^{(h)} = \frac{\eta \cdot m_{\text{meta}} \cdot s_{\sigma^2}}{N} \cdot (y \odot (y - \sqrt{\theta_{\text{BCM}}}))^{\top} \cdot x$$

where m_{meta} is a metaplasticity learning rate multiplier that adapts based on weight change history, and $s_{\sigma^2} = 1 + \min(\overline{\sigma^2}, 10)$ increases the learning rate for columns with high prediction uncertainty.

Claim 6. The system of Claim 1, wherein the Hebbian optimizer further comprises an error-modulated mode that blends correlational and error-driven signals:

$$\text{modulated} = \text{bcm_mod} \odot \left((1 - w_e) + w_e \cdot \frac{||\epsilon||}{\mathbb{E}[||\epsilon||]} \right)$$

where w_e is a configurable error weight parameter.

Claim 7. The system of Claim 1, wherein the neuromodulatory plasticity gate computes:

$$\text{plasticity} = \text{DA} \times \text{ACh} \times (0.5 + 0.5 \times \text{NE}) \times (0.5 + 0.5 \times \text{5-HT})$$

such that at neutral neuromodulatory state with all signals at 0.5, the plasticity gate produces approximately 0.14.

Claim 8. The system of Claim 1, further comprising a brainstem module with learned projection weights, wherein the brainstem module computes the four neuromodulator values from an aggregate neural representation, and wherein the projection weights are updated via analytical Hebbian correlation rules:

- a dopamine projection correlating with prediction error magnitude,
- a norepinephrine projection correlating with prediction uncertainty,
- an acetylcholine projection correlating with inverse prediction error, and
- a serotonin projection correlating with prediction error temporal stability.

Claim 9. The system of Claim 8, wherein the brainstem projection weights are clamped to a range $[-1, 1]$ after each update, and wherein the brainstem learning rate is a fraction of the Hebbian learning rate.

Claim 10. The system of Claim 1, further comprising a thalamic relay learning module configured to update thalamic input and output weight matrices via error-correlation Hebbian rules using aggregate error signals from the processing columns served by each relay group.

Claim 11. The system of Claim 1, further comprising a cortical workspace learning module configured to update write weights via surprise-weighted Hebbian learning, wherein the surprise

signal is the per-token error magnitude, and to update read weights via error-reduction Hebbian learning.

Claim 12. The system of Claim 1, further comprising a cerebellar learning module implementing a delta rule on temporal prediction weights, wherein a climbing fiber signal carries prediction error and parallel fiber–Purkinje cell synapses are modified via:

$$\Delta W_{\text{predict}} = -\frac{\eta_{\text{cb}}}{N} \cdot \epsilon^{\top} \cdot \text{GELU}(W_{\text{in}} \cdot \text{RMSNorm}(o))$$

Claim 13. The system of Claim 1, further comprising a lateral inter-column learning module configured to update send and receive weights between adjacent processing columns via error-correlation Hebbian rules, strengthening connections between columns whose outputs reduce each other's prediction errors.

Claim 14. The system of Claim 1, wherein the dendritic error propagation module further comprises:

- a basal dendritic gate weight updated using error signals correlated with feedforward input from a thalamic input layer, and
- an apical dendritic gate weight updated using error signals correlated with a top-down context signal,

wherein both dendritic gates modulate the feedforward network output via sigmoid gating.

Claim 15. The system of Claim 1, wherein the dendritic error propagation module implements a bootstrap-safe error signal selection mechanism that detects when a down-projection weight norm is below a threshold and substitutes a mean error magnitude signal broadcast to the feedforward dimension, ensuring dendritic gates receive a learning signal during early training when the down-projection weight is near zero.

Claim 16. The system of Claim 1, wherein the local output correction module updates output pathway weights using a low-rank decomposition:

$$\Delta(\Delta A) = -\frac{\eta}{N_{\text{valid}}} \cdot \text{error}^{\top} \cdot (h \cdot \Delta B^{\top})$$
$$\Delta(\Delta B) = -\frac{\eta}{N_{\text{valid}}} \cdot (\text{error} \cdot \Delta A)^{\top} \cdot h$$

where $\Delta A \in \mathbb{R}^{V \times r}$ and $\Delta B \in \mathbb{R}^{r \times D}$ are production pathway matrices, and each update is gradient-norm clipped to a maximum norm of 1.0.

Claim 17. The method of Claim 2, wherein the step of updating prediction weight matrices further comprises computing precision weight updates via a chain rule through a softplus function:

$$\Delta \mathbf{W}_{\sigma} = \frac{\eta}{N} \left(-\frac{\text{MSE}(\epsilon)}{2(\sigma^2)^2} + \frac{1}{2\sigma^2} \right) \cdot \text{sigmoid}(z) \cdot h_{\text{norm}}^{\top}$$

where the MSE uses mean-squared-error over the model dimension to maintain gradient scale independence.

Claim 18. The method of Claim 2, wherein the step of updating association weight matrices further comprises modulating the BCM update by a sigma-squared adaptive learning rate multiplier $s = 1 + \min(\overline{\sigma^2}, 10)$ and a metaplasticity multiplier that tracks the magnitude of recent weight changes.

Claim 19. The method of Claim 2, further comprising updating brainstem neuromodulator projection weights via analytical Hebbian correlation rules at a fraction of the Hebbian learning rate, wherein dopamine projection weights correlate with error magnitude, norepinephrine projection weights correlate with precision uncertainty, acetylcholine projection weights correlate with inverse error, and serotonin projection weights correlate with error temporal stability.

Claim 20. The method of Claim 2, further comprising updating local modulatory circuit parameters including:

- L4 input projection weights via error-correlation at a reduced learning rate for input layer stability,
- L1 top-down modulation weights via prediction-error Hebbian correlation,
- L6 thalamic feedback weights via column state-to-error correlation,
- oscillatory coupling parameters via phase-error anti-correlation, and
- interneuron weights via activity-dependent error-modulated plasticity at reduced learning rates for inhibitory stability.

Claim 21. The method of Claim 2, further comprising updating working memory gate parameters in proportion to prediction error surprise when the error exceeds a threshold, and updating working memory write weights via surprise-driven Hebbian correlation and read weights via error-reduction Hebbian correlation.

Claim 22. The method of Claim 2, wherein the step of propagating top-down error signals further comprises a bootstrap-safe error signal selection that, when a down-projection weight norm is below a threshold value, substitutes a mean error magnitude broadcast signal for the chain-rule-computed error signal.

Claim 23. The method of Claim 2, wherein the step of applying micro-downscaling comprises multiplying synaptic weight matrices by a factor $(1 - 0.05 \times (1 - s))$ every k training steps, where s is a configurable downscale parameter, and wherein the downscaling is applied only to association, lateral, distance, value, and output weight matrices, and not to gate parameters, normalization weights, biases, or feedforward core weights.

Claim 24. The method of Claim 2, wherein the step of monitoring precision parameters comprises checking whether the absolute maximum of a precision bias parameter exceeds a threshold of 20.0, and upon exceeding, multiplicatively scaling both the precision weight matrix and bias by a factor of 0.9.

Claim 25. The method of Claim 2, further comprising applying elastic weight consolidation by computing an elastic pull on column parameters toward anchor values weighted by per-parameter importance estimates, thereby protecting consolidated knowledge during continual learning.

Claim 26. The non-transitory computer-readable medium of Claim 3, wherein the method further comprises updating brainstem neuromodulator projections, thalamic relay weights, workspace broadcast weights, cerebellar prediction weights, and lateral inter-column connection weights, each using analytical Hebbian correlation rules that are specific to the biological function of the respective subcortical system.