Autonomous Self-Directed Learning in Neural Networks with Biologically-Grounded Wake-Sleep Cycles

Open research specification 4 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 FIELD **3**

2 BACKGROUND **3**

2.1 Limitations of Conventional Neural Network Training 3

2.2 Prior Art in Sleep-Based and Continual Learning 4

2.2.1 Synaptic Homeostasis Hypothesis (Tononi & Cirelli, 2006) 4

2.2.2 Elastic Weight Consolidation (Kirkpatrick et al., 2017) 4

2.2.3 Active Inference (Friston, 2010) 4

2.2.4 Hinton's Wake-Sleep Algorithm (Hinton et al., 1995) 5

2.3 Problems Solved by the Present Invention 5

3 SUMMARY **6**

3.1 Always-On Learning Module 6

3.2 Surprise-Weighted Sleep Pressure Accumulator 6

3.3 Dual Sleep Trigger Mechanism 6

3.4 Multi-Phase Sleep Consolidation 7

3.5 Imagination Engine (Dream Learning) 7

3.6 Curiosity-Driven Active Inference 8

3.7 Self-Play Learning 8

3.8 Activation-Based Continual Learning Guard 8

3.9 Confidence-Weighted Consolidation 9

3.10 Prenatal Self-Organization 9

3.11 Trend-Based Self-Evaluation and Learning Rate Adaptation 10

3.12 Developmental Critical Periods 10

4 DESCRIPTION OF THE FIGURES **10**

5 DETAILED DESCRIPTION **13**

5.1 System Architecture Overview 13

5.2 Autonomous Lifecycle State Machine 14

5.3 Always-On Learning Module 15

5.4 Surprise-Weighted Sleep Pressure Accumulator 15

5.5 Dual Sleep Trigger Mechanism 16

5.6 Multi-Phase Sleep Consolidation Cycle 16

5.6.1 Phase 1: Generative Replay 16

5.6.2 Phase 2: Dream Learning via Imagination Engine 17

5.6.3 Phase 2.5: Semantic Consolidation 18

5.6.4 Phase 3: Synaptic Downscaling 18

5.7 Curiosity-Driven Active Inference 19
5.7.1 Input Scoring 19
5.7.2 Active Prompt Generation via Cross-Episode Memory Recombination . 20
5.8 Self-Play Learning 20
5.9 Activation-Based Elastic Weight Consolidation 21
5.9.1 Importance Estimation Without Backpropagation 21
5.9.2 Importance Decay and Anchor Parameters 22
5.10 Prenatal Self-Organization 22
5.10.1 Phase 1: Spontaneous Retinal Waves (30% of prenatal steps) 22
5.10.2 Phase 2: Motor Babbling (25% of prenatal steps) 23
5.10.3 Phase 3: Competitive Differentiation (25% of prenatal steps) 23
5.10.4 Phase 4: Oscillatory Calibration (20% of prenatal steps) 23
5.10.5 Convergence Detection 24
5.10.6 Prenatal Sleep Cycles 24
5.10.7 Prenatal Hippocampal Encoding 24
5.11 Trend-Based Self-Evaluation and Learning Rate Adaptation 24
5.12 Developmental Critical Periods 25
5.12.1 Stage 1: Sensory (Steps 0–1,000) 25
5.12.2 Stage 2: Association (Steps 1,000–5,000) 25
5.12.3 Stage 3: Integration (Steps 5,000–20,000) 25
5.12.4 Stage 4: Refinement (Steps 20,000+) 25
5.13 State Persistence and Resumption 26
5.14 Implementation Considerations 26
5.14.1 Memory Safety 26
5.14.2 OOM Resilience 27
5.14.3 GPU Memory Shield Compatibility 27

TITLE

Autonomous Self-Directed Learning in Neural Networks with Biologically-Grounded Wake-Sleep Cycles

1. FIELD

The present invention relates to the field of artificial intelligence and machine learning, and more particularly to systems and methods for autonomous self-directed learning in neural networks that operate without continuous human supervision. Specifically, the invention provides a biologically-grounded autonomous learning framework comprising wake-sleep cycling with multi-phase consolidation, surprise-driven sleep pressure accumulation, imagination-based dream learning, curiosity-driven active inference for self-directed data acquisition, self-play for generative self-improvement, activation-based continual learning guards against catastrophic forgetting, confidence-weighted memory consolidation, and prenatal self-organization—all orchestrated by a unified lifecycle controller that enables a neural network to continuously and autonomously improve its performance without external intervention.

2. BACKGROUND

2.1. Limitations of Conventional Neural Network Training

Conventional deep neural networks, including transformer architectures and recurrent networks, operate under a strict dichotomy between training and inference modes. During training, all model parameters are updated via global backpropagation of error signals; during inference, all learning is disabled and the model operates as a static function. This paradigm suffers from several fundamental limitations:

(a) **No online adaptation:** Once deployed, conventional models cannot learn from new experiences without explicit retraining, requiring human-curated datasets and supervised training pipelines.

(b) **Catastrophic forgetting:** When models are retrained on new data, previously learned knowledge is catastrophically overwritten unless expensive replay buffers or regularization schemes are manually configured.

(c) **No consolidation mechanism:** Conventional training treats all gradient updates uniformly, with no mechanism to distinguish between short-term experiential learning and long-term knowledge consolidation.

(d) **No self-directed exploration:** Standard training requires externally provided data batches; the model has no capacity to identify what it does not know or to seek out informative training experiences autonomously.

(e) **No developmental staging:** Learning rate schedules in conventional training (cosine decay, warm-up) are hand-tuned hyperparameters with no principled basis, and do not model the biologically established phenomenon of critical periods that fundamentally alter the nature of learning at different developmental stages.

(f) **No self-initialization:** Neural networks begin from random parameter initialization, requiring large volumes of supervised data to bootstrap useful representations, whereas biological neural networks undergo extensive self-organization before receiving external sensory input.

2.2. Prior Art in Sleep-Based and Continual Learning

2.2.1. Synaptic Homeostasis Hypothesis (Tononi & Cirelli, 2006)

Tononi and Cirelli proposed the Synaptic Homeostasis Hypothesis (SHY), which posits that wakefulness leads to net synaptic potentiation throughout the brain, and that the function of sleep is to provide synaptic downscaling—a global, multiplicative reduction of synaptic weights that restores the brain's capacity for new learning. While SHY provides a theoretical framework for understanding synaptic homeostasis, prior art has not disclosed a complete computational implementation comprising multi-phase sleep consolidation with generative replay, contrastive consolidation, semantic memory transfer, and surprise-weighted synaptic downscaling, all driven by an adaptive sleep pressure accumulator.

2.2.2. Elastic Weight Consolidation (Kirkpatrick et al., 2017)

Kirkpatrick et al. introduced Elastic Weight Consolidation (EWC), which computes the Fisher Information Matrix to estimate parameter importance and penalizes changes to important parameters during new learning. However, computing the Fisher Information requires full backpropagation passes over held-out data, which is computationally expensive and requires memory proportional to the number of model parameters, making it impractical for large-scale models on memory-constrained hardware. The present invention discloses an activation-based importance estimation method that approximates Fisher Information using only forward-pass activations and prediction errors, requiring zero autograd computation.

2.2.3. Active Inference (Friston, 2010)

Friston's Active Inference framework proposes that biological agents minimize expected free energy by actively seeking states that reduce uncertainty. While Active Inference provides the

theoretical motivation, prior art has not disclosed a practical computational implementation that combines precision-weighted uncertainty (σ^2 at discourse scale) with prediction error to generate a scalar curiosity signal, nor a cross-episode memory recombination strategy that generates novel training prompts from hippocampal memory fragments targeting the model's areas of maximum uncertainty.

2.2.4. Hinton's Wake-Sleep Algorithm (Hinton et al., 1995)

Hinton et al. proposed the wake-sleep algorithm for training generative models, using a recognition network during the wake phase and a generative network during the sleep phase. This algorithm is limited to two-phase learning in specific generative model architectures (Helmholtz machines) and does not incorporate biologically-grounded consolidation, curiosity-driven exploration, developmental staging, or adaptive sleep pressure mechanisms.

2.3. Problems Solved by the Present Invention

The present invention addresses all of the above limitations by providing a unified autonomous learning system that:

(i) Learns continuously from every experience without distinguishing between training and inference modes, modulated by neuromodulatory signals and sleep pressure;

(ii) Accumulates sleep pressure proportional to prediction error (surprise), analogous to adenosine accumulation in biological systems, and autonomously triggers sleep cycles when consolidation is needed;

(iii) Executes multi-phase sleep consolidation comprising generative replay, contrastive learning, semantic memory transfer from episodic to cortical stores, and global synaptic downscaling;

(iv) Generates novel training experiences through imagination (dream learning) where all learning rules fire simultaneously on hippocampal memory replay;

(v) Actively seeks informative experiences through curiosity-driven exploration based on precision-weighted uncertainty estimation;

(vi) Learns from its own generated output through self-play with adaptive frequency control;

(vii) Prevents catastrophic forgetting through activation-based elastic weight consolidation that requires no backpropagation;

(viii) Prioritizes consolidation of high-confidence, schema-consistent memories through confidence-weighted training passes;

(ix) Undergoes structured prenatal self-organization through four developmental phases before external data exposure;

(x) Adapts its own learning parameters through periodic self-evaluation with trend-based learning rate adjustment; and

(xi) Models developmental critical periods that systematically alter the balance between exploration and exploitation, Hebbian and error-driven learning, and structural and parametric plasticity across the system's lifetime.

3. SUMMARY

The present invention provides a system and method for autonomous self-directed learning in neural networks, comprising a biologically-grounded autonomous learning framework that enables a neural network to continuously improve without human intervention.

3.1. Always-On Learning Module

In a first aspect, the invention provides an always-on learning module that executes local synaptic updates after every forward pass—whether during training, inference, or generation—eliminating the conventional distinction between training and evaluation modes. The effective learning rate is modulated by:

- Acetylcholine level (attention modulation): $\text{lr}_{\text{eff}} \propto (0.5 + \text{ACh})$;
- Norepinephrine level (arousal modulation): $\text{lr}_{\text{eff}} \propto (0.5 + \text{NE})$;
- Wake efficiency (fatigue modulation): $\text{lr}_{\text{eff}} \propto \max(0.1,\ 1.0 - p_{\text{sleep}})$,

where $p_{\text{sleep}} \in [0, 1]$ is the current sleep pressure.

3.2. Surprise-Weighted Sleep Pressure Accumulator

In a second aspect, the invention provides a sleep pressure accumulator that models adenosine build-up during wakefulness. Sleep pressure accumulates according to:

$$p(t) = \min\big(1.0,\ p(t-1) + \beta_{\text{base}} \cdot (1 + \min(\text{PE}, 5))\big) \tag{1}$$

where $\beta_{\text{base}} = 1 - \lambda_{\text{decay}}$ is the base increment per step and PE is the mean prediction error from the most recent forward pass. High-surprise experiences accumulate sleep pressure up to six times faster than low-surprise experiences, causing the system to autonomously trigger sleep consolidation when it has encountered particularly challenging input.

3.3. Dual Sleep Trigger Mechanism

In a third aspect, the invention provides a dual-condition sleep trigger:

1. **Primary (pressure-based):** Sleep is triggered when $p_{\text{sleep}} \geq 0.7$, modeling the physiological adenosine threshold;

2. **Failsafe (step-count):** Sleep is triggered when the number of wake steps since the last sleep phase exceeds a configurable interval, preventing pathological insomnia in cases of sustained low-surprise input.

3.4. Multi-Phase Sleep Consolidation

In a fourth aspect, the invention provides a four-phase sleep consolidation cycle:

Phase 1 — Generative Replay: Hippocampal memories are replayed through cortical columns with local predictive coding and Hebbian learning updates, enabling the cortex to learn to reconstruct stored experiences without hippocampal support.

Phase 2 — Contrastive Consolidation: Real memory patterns potentiate cortical synapses while corrupted (noise-injected) patterns depress them, sharpening stored representations through differential plasticity.

Phase 2.5 — Semantic Consolidation: Similar hippocampal episodes are clustered, centroid patterns are extracted as "gist" representations, and cortical columns are trained on these semantic summaries through slow interleaved replay, implementing the Complementary Learning Systems theory (McClelland et al., 1995). Consolidated episodes are removed from the episodic store.

Phase 3 — Synaptic Downscaling: All synaptic weight parameters are multiplicatively reduced by a configurable factor (default: 1% per sleep phase), implementing Tononi's SHY. Protected parameters—including gate set points, bias values, normalization gains, oscillatory periods, receptor thresholds, and decay constants—are excluded from downscaling as they represent intrinsic neuronal properties rather than synaptic connections.

3.5. Imagination Engine (Dream Learning)

In a fifth aspect, the invention provides an imagination engine that generates novel learning experiences from hippocampal memory replay during the dream phase. Critically, *all* learning rules fire simultaneously during dream episodes:

1. Predictive coding local updates on prediction weights and precision parameters;

2. L5 feedforward network updates (SwiGLU and dendritic compartments);

3. Local parameter updates on thalamic input, top-down modulation, thalamic feedback, and association output pathways;

4. BCM Hebbian updates with anti-Hebbian decorrelation on lateral association weights;

5. Metaplasticity tracking of weight changes induced by dream plasticity.

Dream inputs consist of hippocampal memories with added Gaussian noise (default scale 0.05), modeling the noisy reconstructive nature of biological dreaming.

3.6. Curiosity-Driven Active Inference

In a sixth aspect, the invention provides a curiosity drive implementing Friston's Active Inference principle. The curiosity score for a candidate input is computed as:

$$c = w_{\text{curiosity}} \cdot \sigma^2_{\text{discourse}} + w_{\text{exploit}} \cdot \text{PE} \tag{2}$$

where $\sigma^2_{\text{discourse}}$ is the precision estimate at the discourse processing scale and PE is the prediction error. The curiosity drive further implements active prompt generation through cross-episode memory recombination: N hippocampal memories are retrieved, their token-level representations are concatenated and randomly permuted to create novel sequences, and the resulting representations are forwarded through the model to generate training token sequences that target the model's areas of maximum uncertainty.

3.7. Self-Play Learning

In a seventh aspect, the invention provides a self-play mechanism wherein the model generates text, re-processes its own output as input, evaluates the prediction error (surprise at its own generation), and applies local learning rules to reduce self-inconsistency. The self-play module incorporates adaptive frequency control: if three consecutive self-play sessions produce loss values exceeding a threshold (default: 15.0), the self-play frequency is reduced by a factor of five to avoid wasting computation on gibberish self-evaluation during early training.

3.8. Activation-Based Continual Learning Guard

In an eighth aspect, the invention provides an activation-based Elastic Weight Consolidation method that requires zero autograd computation. Parameter importance is estimated as:

$$F_i \approx \mathbb{E}\big[|a_i|^2 \cdot |\text{PE}|^2\big] \tag{3}$$

where a_i is the parameter activation magnitude and PE is the column-level prediction error. This formulation provides a biologically-grounded analog of synaptic tagging (Frey & Morris, 1997): synapses that are highly active during high-error predictions are tagged for protection, making them resistant to overwriting during subsequent learning. The method eliminates the memory and computation overhead of Fisher Information estimation via backpropagation, enabling continual learning on memory-constrained hardware.

3.9. Confidence-Weighted Consolidation

In a ninth aspect, the invention provides confidence-weighted consolidation wherein the number of training passes and learning rate during semantic consolidation are modulated by memory confidence and schema consistency:

$$\text{priority} = \text{confidence} \times \text{schema_boost} \tag{4}$$

$$n_{\text{passes}} = \text{clamp}(5 \times \text{priority},\ 3,\ 8) \tag{5}$$

$$\text{lr}_{\text{consol}} = 10^{-4} \times \text{priority} \tag{6}$$

High-confidence patterns that are consistent with existing knowledge schemas receive up to 8 training passes with amplified learning rate, while low-confidence or schema-violating patterns receive the minimum 3 passes, ensuring that reliable knowledge is preferentially integrated into long-term cortical representations.

3.10. Prenatal Self-Organization

In a tenth aspect, the invention provides a four-phase prenatal development process that establishes functional connectivity before any external data is presented:

Phase 1 — Spontaneous Retinal Waves (30%): Temporally-correlated random walks with momentum in embedding space, teaching the association layers about local temporal structure. Activity is generated as $h_t = h_{t-1} + v_t$ where velocity $v_t = 0.9 \cdot v_{t-1} + 0.1 \cdot \mathcal{N}(0, 0.05^2)$.

Phase 2 — Motor Babbling (25%): The model generates sequences autoregressively from random seeds, then processes its own output with local learning—bootstrapping sensorimotor coordination.

Phase 3 — Competitive Differentiation (25%): Five distinct pattern types (repeated motifs, ascending sequences, alternating patterns, segmented frequency bands, pure random) are presented to induce column-level functional specialization through competitive Hebbian learning.

Phase 4 — Oscillatory Calibration (20%): Multi-frequency oscillatory patterns with phase-amplitude coupling (gamma modulated by theta phase) calibrate the network's oscillatory dynamics and cross-frequency coupling parameters.

Convergence is detected when weight coefficient of variation falls below 5% and predictive coding loss trend change falls below 2% over a sliding window.

3.11. Trend-Based Self-Evaluation and Learning Rate Adaptation

In an eleventh aspect, the invention provides periodic self-evaluation wherein the model tests itself on held-out hippocampal memories and adapts its learning strategy based on performance trends:

- **Improving** (recent loss $< 0.95\times$ older loss): Advance developmental program by 50 steps, accelerating maturation;
- **Degrading** (recent loss $> 1.05\times$ older loss): Reduce sleep interval (more frequent consolidation), reduce Hebbian learning rate by 20%;
- **Emergency** (recent loss $> 1.2\times$ older loss): Force sleep pressure to 1.0, triggering immediate emergency sleep consolidation;
- **Stagnant** (coefficient of variation $< 2\%$ over 5 evaluations): Increase Hebbian learning rate by 50% (capped at $10\times$ initial) and elevate sleep pressure to trigger explorative consolidation.

3.12. Developmental Critical Periods

In a twelfth aspect, the invention provides a developmental staging system with four critical periods that modulate learning characteristics:

Table 1: Developmental Critical Period Parameters

Stage	Steps	LR Mult.	Inhib.	Hebbian Wt.	Sleep Int. Mult.
Sensory	0–1K	2.0×	0.3	0.8	2.0×
Association	1K–5K	1.5×	0.6	0.6	1.5×
Integration	5K–20K	1.0×	0.9	0.4	1.0×
Refinement	20K+	0.5×	1.0	0.3	0.7×

The progression from high Hebbian weight (correlation-driven learning) to low Hebbian weight (error-driven learning) models the biological transition from infant-like statistical absorption to adult-like predictive refinement, while the increasing inhibitory strength models GABAergic maturation that triggers critical period closure (Hensch, 2004).

4. DESCRIPTION OF THE FIGURES

The invention will be more fully understood from the following detailed description taken in conjunction with the accompanying drawings, in which:

FIG. P4-1 is a system-level block diagram showing the autonomous self-directed learning system of Claim 1, including the neural network model with prediction and precision outputs, the always-on learning module, sleep pressure accumulator, dual sleep trigger, multi-phase sleep consolidation module (replay, dream, downscaling), and the lifecycle controller orchestrating the continuous autonomous loop;

FIG. P4-2 is a detailed diagram of the sleep pressure accumulation formula of Claim 2, showing the base increment $\beta_{\text{base}} = 1 - \lambda_{\text{decay}}$, the surprise multiplier $(1 + \min(\text{PE}, 5.0))$ ranging from $1\times$ to $6\times$, and the resulting adaptive pressure accumulation rate;

FIG. P4-3 is a flowchart of the dual sleep trigger mechanism of Claim 3, showing the pressure-based trigger at threshold 0.7 (surprise-driven pathway) and the step-count failsafe trigger (time-based pathway), connected by an OR gate to initiate sleep consolidation;

FIG. P4-4 is a block diagram showing the five concurrent learning rules applied during the dream learning phase of Claim 4: (i) predictive coding update, (ii) FFN dendritic update, (iii) local parameter update, (iv) BCM Hebbian update with anti-Hebbian decorrelation, and (v) metaplasticity tracking, all applied concurrently on each dream episode;

FIG. P4-5 is a categorized listing of the non-synaptic parameters excluded from synaptic downscaling per Claim 5, organized into gate parameters, normalization parameters, oscillatory parameters, receptor parameters, memory parameters, spatial parameters, and competition parameters;

FIG. P4-6 is a diagram showing the wake efficiency modulation of Claim 6, illustrating the formula wake_eff $= \max(0.1, 1.0 - p_{\text{sleep}})$ and how it multiplicatively scales both Hebbian and predictive coding learning rates as sleep pressure accumulates;

FIG. P4-7 is a block diagram of the curiosity drive module of Claim 7, showing the active inference score computation from discourse-level precision and prediction error, and the cross-episode memory recombination pipeline for active prompt generation;

FIG. P4-8 is a diagram of the top-k curiosity ranking mechanism of Claim 8, showing how candidate inputs are scored, ranked by curiosity score, and the most informative candidates selected for processing;

FIG. P4-9 is a flowchart of the self-play module of Claim 9, showing the four-phase pipeline: generation, re-processing, surprise evaluation, and local learning, together with the adaptive frequency controller and gibberish detection mechanism;

FIG. P4-10 is a detailed diagram of the adaptive frequency controller parameters of Claim 10, showing the gibberish threshold of 15.0, the $5\times$ interval multiplier on three consecutive exceedances, and the recovery condition;

FIG. P4-11 is a diagram of the activation-based continual learning guard of Claim 11, contrasting the conventional Fisher Information method (requiring backpropagation) with the disclosed activation-error correlation method (forward-pass only), and showing the EWC penalty computation with exponential importance decay;

FIG. P4-12 is a diagram of the hippocampal-based importance estimation of Claim 12, showing how episodic memories retrieved during wake processing provide the forward-pass activations and prediction errors used for zero-autograd importance estimation;

FIG. P4-13 is a diagram of the semantic consolidation phase of Claim 13, showing the five-step pipeline: cosine similarity clustering, centroid extraction, confidence-weighted priority scoring, multi-pass cortical training, and episodic memory clearance;

FIG. P4-14 is a detailed diagram of the semantic consolidation learning rules of Claim 14, showing working memory reset before consolidation and the four concurrent learning rules (BCM Hebbian, predictive coding, FFN update, local parameter update) applied during each training pass;

FIG. P4-15 is a four-panel diagram of the prenatal self-organization method of Claim 15, showing (a) spontaneous retinal waves with momentum-driven random walks, (b) motor babbling with autoregressive self-generation, (c) competitive differentiation with five pattern types, and (d) oscillatory calibration with phase-amplitude coupling, annotated with convergence criteria;

FIG. P4-16 is a pie chart showing the prenatal phase allocation of Claim 16: 30% spontaneous waves, 25% motor babbling, 25% competitive differentiation, and 20% oscillatory calibration;

FIG. P4-17 is a diagram of the spontaneous wave generation formula of Claim 17, showing the velocity update equation $v_t = \mu \cdot v_{t-1} + (1 - \mu) \cdot \mathcal{N}(0, \sigma^2 \cdot I_d)$, the position integration $h_t = h_{t-1} + v_t$, and layer normalization;

FIG. P4-18 is a diagram of the fetal sleep cycles during prenatal development per Claim 18, showing sleep trigger checks after each prenatal step and mandatory working memory reset between all steps to prevent cross-contamination;

FIG. P4-19 is a diagram of the prenatal hippocampal encoding of Claim 19, showing how spontaneous activity is encoded into the hippocampal store with an `is_production=False` flag distinguishing self-organized patterns from production experiences;

FIG. P4-20 is a diagram of the convergence monitoring parameters of Claim 20, showing the 50-step sliding window, the 5% coefficient of variation threshold for weight stability, and the 2% loss trend change threshold;

FIG. P4-21 is a flowchart of the trend-based adaptive self-directed learning method of Claim 21, showing the wake processing loop, periodic evaluation on held-out memories, trend comparison with four response categories (improving, degrading, emergency, stagnation), and the developmental staging system;

FIG. P4-22 is a detailed diagram of the specific trend ratios and corrective actions of Claim 22, showing the improving threshold (< 0.95), degrading threshold (> 1.05), and emergency threshold (> 1.2) with their corresponding corrective actions;

FIG. P4-23 is a diagram of the stagnation detection and correction mechanism of Claim 23, showing the coefficient of variation threshold ($< 2\%$ over 5 evaluations), the $1.5\times$ learning rate increase capped at $10\times$ initial, and the sleep pressure elevation to 0.8;

FIG. P4-24 is a timeline diagram of the four critical periods of developmental staging per Claim 24, showing the sensory period (LR $2.0\times$, inhibition 0.3, Hebbian 0.8, sleep $2.0\times$), association period (LR $1.5\times$, inhibition 0.6, Hebbian 0.6, sleep $1.5\times$), integration period (LR $1.0\times$, inhibition 0.9, Hebbian 0.4, sleep $1.0\times$), and refinement period (LR $0.5\times$, inhibition 1.0, Hebbian 0.3, sleep $0.7\times$);

FIG. P4-25 is a diagram of the periodic evaluation on hippocampal memories per Claim 25, showing the retrieval of up to 5 episodic memories, forward evaluation through cortical columns, mean squared reconstruction error computation, and the sliding window trend comparison using 3 recent vs. 3 earlier evaluations.

5. DETAILED DESCRIPTION

The following detailed description sets forth specific embodiments of the invention. It will be understood that the invention is not limited to the specific embodiments described herein, and that various modifications, substitutions, and combinations of features may be made without departing from the spirit and scope of the invention as defined by the appended claims.

5.1. System Architecture Overview

Referring to FIG. P4-1, the autonomous learning system 100 comprises a neural network model 110 (preferably a cortical column network as described in related Patent Application 1 of 7) and an autonomous loop controller 120 that orchestrates all aspects of the system's self-directed learning lifecycle.

The autonomous loop controller 120 comprises:

- An always-on learning module 130 that executes local synaptic updates after every forward pass;
- A sleep pressure accumulator 140 that tracks the system's need for consolidation;

- An imagination engine 150 that generates dream-based learning experiences;
- A curiosity drive 160 that implements active inference for self-directed data acquisition;
- A self-play module 170 that enables learning from self-generated output;
- A continual learning guard 180 that prevents catastrophic forgetting;
- A consolidation manager 190 that transfers knowledge from episodic to semantic memory;
- A prenatal development module 195 that provides structured self-organization before external data exposure;
- A developmental program 196 that manages critical period scheduling; and
- A self-evaluation module 197 that monitors performance and adapts learning parameters.

5.2. Autonomous Lifecycle State Machine

Referring to FIG. P4-1, the autonomous loop controller 120 implements a state machine with five primary states:

1. **WAKE** state (default): The system processes input data through the neural network, executes always-on learning, accumulates sleep pressure, and optionally runs curiosity-driven exploration and self-play;
2. **SLEEP** state: Entered when the dual sleep trigger fires; the system executes generative replay through hippocampal memory re-processing;
3. **DREAM** state: Following sleep, the imagination engine generates novel learning experiences from memory cues with added noise;
4. **CONSOLIDATE** state: When sufficient hippocampal memories have accumulated, semantic consolidation transfers episodic knowledge to cortical representations;
5. **EVALUATE** state: Periodically during wake, the system tests itself on held-out memories and adapts learning parameters.

The lifecycle executes as a continuous loop:

$$\text{WAKE} \xrightarrow{p_{\text{sleep}} \geq 0.7 \ \vee \ \text{steps} \geq \text{interval}} \text{SLEEP} \rightarrow \text{DREAM} \rightarrow \text{CONSOLIDATE} \rightarrow \text{WAKE} \quad (7)$$

On each wake step, the autonomous loop controller 120 executes the following sequence:

1. Forward pass through the neural network model 110;
2. Always-on learning via module 130 (if enabled);

3. EWC penalty computation via module 180 (if initialized);
4. Sleep check via dual trigger mechanism;
5. Self-play via module 170 (if scheduled);
6. Curiosity-driven exploration via module 160 (if enabled);
7. Self-evaluation via module 197 (if scheduled).

5.3. Always-On Learning Module

Referring to FIG. P4-6, the always-on learning module 130 eliminates the conventional distinction between training and inference. In biological neural systems, synaptic modification occurs continuously—there is no mechanism equivalent to a model entering "eval mode." After every forward pass, the module 130 executes local learning updates comprising both Hebbian association updates and predictive coding weight corrections.

The effective learning rate is computed as:

$$\text{lr}_{\text{eff}} = \text{lr}_{\text{base}} \times (0.5 + \text{ACh}_{\text{mean}}) \times (0.5 + \text{NE}_{\text{mean}}) \times \text{wake_eff} \tag{8}$$

where $\text{wake_eff} = \max(0.1,\ 1.0 - p_{\text{sleep}})$ models the biological decrease in synaptic plasticity as adenosine accumulates during prolonged wakefulness.

Both the Hebbian and predictive coding learning rates are modulated by wake efficiency, ensuring that all forms of synaptic plasticity diminish as the system approaches the sleep threshold. This prevents runaway potentiation during prolonged wake periods and creates a natural pressure toward consolidation.

5.4. Surprise-Weighted Sleep Pressure Accumulator

Referring to FIG. P4-2, the sleep pressure accumulator 140 models adenosine build-up in biological systems (Basheer et al., 2004). Sleep pressure is maintained as a scalar value $p_{\text{sleep}} \in [0, 1]$, initialized to 0.0 after each sleep phase.

On each wake experience, sleep pressure accumulates according to:

$$p_{\text{sleep}}(t) = \min\big(1.0,\ p_{\text{sleep}}(t{-}1) + \beta_{\text{base}} \cdot (1 + \min(\text{PE}, 5.0))\big) \tag{9}$$

where:

- $\beta_{\text{base}} = 1 - \lambda_{\text{decay}}$ is the base increment per step (default: $\lambda_{\text{decay}} = 0.999$, giving $\beta_{\text{base}} = 0.001$);
- PE is the mean prediction error from the most recent forward pass;

- The surprise multiplier $(1+\min(\text{PE}, 5.0))$ ranges from $1\times$ (zero surprise) to $6\times$ (maximum surprise), capped at 5.0 to prevent pathological pressure spikes.

This formulation ensures that:

(a) Routine, well-predicted experiences accumulate sleep pressure slowly (multiplier ≈ 1);

(b) Surprising or novel experiences accumulate pressure rapidly (multiplier up to $6\times$);

(c) The system autonomously schedules more frequent sleep after encountering challenging input;

(d) Sleep pressure is bounded in $[0, 1]$, preventing numerical overflow.

After each sleep phase, sleep pressure is reset to 0.0, analogous to the clearance of adenosine during biological sleep.

5.5. Dual Sleep Trigger Mechanism

Referring to FIG. P4-3, the dual trigger mechanism ensures robust sleep scheduling under all conditions. Sleep is triggered when *either* of two conditions is met:

1. **Pressure-based trigger (primary):** $p_{\text{sleep}} \geq 0.7$. This threshold was selected to model the biological observation that sleep onset occurs when adenosine accumulation reaches approximately 70% of its saturation level. The primary trigger ensures that sleep occurs when the system genuinely needs consolidation, regardless of step count.

2. **Step-count trigger (failsafe):** wake_steps $\geq$ sleep_interval. This failsafe prevents pathological insomnia in cases where the system processes only well-predicted (low-surprise) input for extended periods, ensuring that consolidation occurs at least once per configurable interval (default: 100 wake steps).

5.6. Multi-Phase Sleep Consolidation Cycle

Referring to FIG. P4-4, upon entering the sleep state, the system executes a four-phase consolidation cycle within a `torch.no_grad()` context (all sleep learning is analytical, requiring no autograd).

5.6.1. Phase 1: Generative Replay

The neural network model 110 executes its internal sleep phase, replaying hippocampal memories through cortical columns with local predictive coding and Hebbian updates. This corresponds to slow-wave sleep (SWS) replay observed in biological hippocampal-cortical circuits, where hippocampal sharp-wave ripples drive cortical reactivation of recently encoded experiences.

5.6.2. Phase 2: Dream Learning via Imagination Engine

The imagination engine 150 retrieves n memories from the hippocampal store (where $n = \min(\text{dream_steps}, \text{memory_size})$, default dream_steps = 10) and processes each through the following dream learning pipeline:

1. **Noise injection:** Gaussian noise with scale $\sigma_{\text{dream}} = 0.05$ is added to the retrieved memory: $x_{\text{dream}} = x_{\text{mem}} + \mathcal{N}(0, \sigma^2_{\text{dream}})$. This models the noisy reconstructive nature of biological dreaming.

2. **Column-wise forward pass:** The noisy memory is processed through all cortical columns sequentially, producing column outputs with predictions and precision estimates.

3. **Prediction error computation:** For each column l, the prediction error is computed as $e_l = t_l - \hat{y}_l$, where t_l is the target (dream input for column 0, output of column $l-1$ for subsequent columns) and $\hat{y}_l$ is the column's prediction.

4. **Precision-weighted loss:** The dream predictive coding loss is:

$$\mathcal{L}_{\text{dream}} = \sum_l \left(\frac{e_l^2}{2\sigma_l^2} + \frac{1}{2} \log \sigma_l^2 \right) \tag{10}$$

5. **Concurrent learning rule activation:** All five learning pathways fire simultaneously:

   (a) *PC local update:* Prediction weights $\mathbf{W}_{\text{predict}}$ and precision parameters $\mathbf{W}_\sigma$ are updated via analytical gradient computation;

   (b) *L5 FFN update:* SwiGLU and dendritic compartment weights are updated using precision-weighted error signals;

   (c) *Local parameter update:* Thalamic input (L4), top-down modulation (L1), thalamic feedback (L6), and association output (L2/3) pathway weights are updated via Hebbian-error hybrid rules;

   (d) *BCM Hebbian update:* BCM threshold is updated from dream activity, and Hebbian potentiation is applied to L2/3 association weights, followed by anti-Hebbian decorrelation;

   (e) *Metaplasticity tracking:* Weight change magnitude during dream phase is tracked by the metaplasticity module, ensuring that post-dream wake learning rates are correctly calibrated.

The simultaneous activation of all learning rules during dreaming is critical: REM sleep in biological systems consolidates both procedural and declarative pathways (Diekelmann & Born, 2010), and partial dream learning creates prediction-computation mismatch upon waking.

5.6.3. Phase 2.5: Semantic Consolidation

Referring to FIG. P4-13, the consolidation manager 190 implements the Complementary Learning Systems theory (McClelland et al., 1995) by transferring knowledge from the fast episodic hippocampal store to slow cortical representations.

The process comprises:

1. **Clustering:** Similar hippocampal episodes are grouped into k clusters (default $k = 8$) using cosine similarity on episode representations, with a minimum cluster size of 3;

2. **Centroid extraction:** For each cluster, the centroid pattern (the "gist") is computed as the mean activation across cluster members;

3. **Confidence-weighted training:** Each centroid pattern is presented to cortical columns for n_{passes} training passes, where:

$$\text{priority} = \text{confidence} \times \text{schema_boost} \tag{11}$$

$$n_{\text{passes}} = \text{clamp}(5 \times \text{priority},\ 3,\ 8) \tag{12}$$

$$\text{lr}_{\text{consol}} = 10^{-4} \times \text{priority} \tag{13}$$

   Each training pass applies all four learning pathways: BCM Hebbian with anti-Hebbian decorrelation, PC local updates, L5 FFN updates, and local parameter updates.

4. **Working memory reset:** Before consolidation, all working memory subsystems (NMDA, silent trace, phonological loop) are reset to prevent cross-contamination between sequential semantic patterns;

5. **Episodic memory clearance:** Successfully consolidated episodes are removed from the hippocampal store, freeing capacity for new episodic encoding.

5.6.4. Phase 3: Synaptic Downscaling

Referring to FIG. P4-5, and implementing Tononi's Synaptic Homeostasis Hypothesis (Tononi & Cirelli, 2006), all synaptic weight parameters are multiplicatively reduced:

$$\theta_i \leftarrow \theta_i \times s \tag{14}$$

where $s = 1 - \alpha \cdot (1 - \text{ds}_{\text{config}})$, α is the downscaling strength (1.0 for full sleep, < 1.0 for micro-downscaling), and $\text{ds}_{\text{config}}$ is the configured synaptic downscale factor (default: 0.99, corresponding to 1% reduction per sleep phase).

The following parameter categories are *excluded* from downscaling, as they represent intrinsic neuronal properties rather than synaptic weights:

- Gate set points (`gate_raw`);
- Bias parameters;
- Normalization gains (`norm.weight`);
- Morpheme route preferences (`freq_logit`);
- Memory decay constants;
- Oscillatory period parameters;
- Phase-amplitude coupling strengths (`kappa`);
- Receptor thresholds and slopes (`ec50`, `slope`);
- Synaptic facilitation and depression parameters;
- Phonological rehearsal strength;
- Global workspace competition temperature;
- Grid cell spatial phase and head scale parameters;
- Phase boundary detection thresholds.

After synaptic downscaling, sleep pressure is reset to 0.0 and the system transitions back to the wake state.

5.7. Curiosity-Driven Active Inference

Referring to FIG. P4-7, the curiosity drive 160 implements the Active Inference principle (Friston, 2010), wherein the system actively seeks inputs that would maximally reduce its uncertainty.

5.7.1. Input Scoring

For each candidate input, the curiosity score is computed as:

$$c = w_{\text{curiosity}} \cdot \sigma^2_{\text{discourse}} + w_{\text{exploit}} \cdot \text{PE} \tag{15}$$

where:

- $w_{\text{curiosity}}$ (default: 1.0) weights the information gain component (discourse-level precision, reflecting the model's uncertainty at the highest-level processing scale);
- w_{exploit} (default: 0.5) weights the exploitation component (prediction error, reflecting immediate learning signal);

- $\sigma^2_{\text{discourse}}$ is the precision estimate from cortical columns processing at the discourse frequency band.

When selecting from a batch of candidates, inputs are ranked by curiosity score and the top-k most informative inputs are selected for processing.

5.7.2. Active Prompt Generation via Cross-Episode Memory Recombination

The curiosity drive further implements active prompt generation through a novel cross-episode memory recombination strategy:

1. N memories (default $N = 3$) are retrieved from the hippocampal store;
2. Token-level representations from all retrieved memories are concatenated along the temporal dimension: $H = [h_1; h_2; ...; h_N]$;
3. Temporal positions are randomly permuted: $H_{\text{shuffled}} = H[:, \pi, :]$ where π is a random permutation;
4. The first L_{prompt} positions (default $L_{\text{prompt}} = 16$) are extracted and forwarded through cortical columns;
5. The LM head projects column output to logits, and greedy argmax decoding produces token IDs that form the curiosity-generated prompt;
6. This prompt is processed as a regular training input with always-on learning.

This recombination strategy ensures that curiosity-generated prompts:

(a) Target the model's areas of maximum uncertainty (memories were stored because they were surprising);

(b) Create novel combinations not present in the training data (cross-episode shuffling);

(c) Produce coherent token sequences (model's own generation, not random tokens).

5.8. Self-Play Learning

Referring to FIG. P4-9, the self-play module 170 enables the model to learn from its own output, analogous to inner speech monitoring in biological neural systems.

The self-play pipeline comprises four phases:

1. **Generation:** The model generates a continuation of length L_{gen} (default: 32 tokens) from a seed prompt using brain-inspired decoding;
2. **Re-processing:** The generated sequence is fed back through the model as input, with the target being the shifted generated sequence;

3. **Surprise evaluation:** The prediction loss on self-generated text measures self-consistency—high loss indicates the model surprised itself;

4. **Local learning:** All local learning rules are applied to reduce self-inconsistency.

The module 170 incorporates adaptive frequency control:

- Self-play is suppressed during the first 200 training steps (warm-up period);
- If the three most recent self-play losses all exceed a gibberish threshold (default: 15.0), the self-play interval is increased by a factor of 5, preventing wasteful computation on incoherent self-evaluation;
- Once generation quality improves (any of the three most recent losses falls below the threshold), normal frequency resumes.

5.9. Activation-Based Elastic Weight Consolidation

Referring to FIG. P4-11, the continual learning guard 180 prevents catastrophic forgetting through an activation-based approximation of Elastic Weight Consolidation that eliminates the need for backpropagation.

5.9.1. Importance Estimation Without Backpropagation

Conventional EWC (Kirkpatrick et al., 2017) estimates parameter importance via the diagonal Fisher Information Matrix:

$$F_i = \mathbb{E}\left[\left(\frac{\partial \mathcal{L}}{\partial \theta_i}\right)^2\right] \tag{16}$$

Computing this requires full backpropagation passes, with memory cost $\mathcal{O}(|\theta|)$ for gradient storage.

The present invention replaces this with an activation-based approximation:

$$\hat{F}_i = \mathbb{E}[|\theta_i|^2 \cdot |\text{PE}_l|^2] \tag{17}$$

where $|\theta_i|^2$ is the squared magnitude of parameter i and $|\text{PE}_l|^2$ is the squared mean prediction error at column l that contains parameter i.

This formulation provides a biologically-grounded analog of synaptic tagging (Frey & Morris, 1997): synapses with high activation magnitude that are active during high-error predictions are tagged as important. The computational advantages are:

- Zero autograd computation required;
- Memory cost: only the importance buffer $\hat{F}$ (same size as parameters) plus prediction errors (scalar per column);

- Compatible with memory-constrained hardware (6 GB GPU);
- Naturally incorporates both parameter magnitude (synaptic strength) and prediction error (functional importance).

5.9.2. Importance Decay and Anchor Parameters

Importance estimates are subject to exponential decay:

$$\hat{F}_i(t) = \lambda_{\text{imp}} \cdot \hat{F}_i(t-1) + \hat{F}_i^{\text{new}} \tag{18}$$

where $\lambda_{\text{imp}} = 0.999$ is the importance decay factor. This allows gradual relearning of old patterns while providing short-term protection against catastrophic overwriting.

After each importance estimation, current parameter values are stored as anchor parameters θ^*. During subsequent learning, the EWC penalty is:

$$\mathcal{L}_{\text{EWC}} = \frac{\lambda_{\text{EWC}}}{2} \sum_i \hat{F}_i \cdot (\theta_i - \theta_i^*)^2 \tag{19}$$

where λ_{EWC} (default: 100.0) controls the penalty strength.

5.10. Prenatal Self-Organization

Referring to FIG. P4-15, the prenatal development module 195 executes structured self-organization before any external data is presented, establishing functional connectivity through four phases.

5.10.1. Phase 1: Spontaneous Retinal Waves (30% of prenatal steps)

Temporally-correlated activity patterns are generated via random walks with momentum in the model's embedding space. For each batch, the generation process is:

$$v_0 = \mathcal{N}(0, 0.02^2 \cdot I_d) \tag{20}$$

$$v_t = 0.9 \cdot v_{t-1} + 0.1 \cdot \mathcal{N}(0, 0.05^2 \cdot I_d) \tag{21}$$

$$h_t = h_{t-1} + v_t \tag{22}$$

$$x = \text{LayerNorm}([h_0, h_1, \ldots, h_{T-1}]) \tag{23}$$

The momentum term (0.9) ensures temporal autocorrelation, teaching the association layers (L2/3) about local sequential structure before any real linguistic input. The resulting continuous activations bypass the embedding layer and are processed directly through cortical columns with position encoding.

5.10.2. Phase 2: Motor Babbling (25% of prenatal steps)

The model generates sequences autoregressively from random token seeds (length 4), producing up to 16 tokens of self-generated output. This generated sequence is then treated as training input with shifted targets, and local learning is applied. This phase bootstraps the model's ability to produce coherent sequential output, analogous to infant babbling that calibrates sensorimotor coordination.

5.10.3. Phase 3: Competitive Differentiation (25% of prenatal steps)

Five distinct pattern types are generated with controlled statistical properties:

(a) *Repeated motifs:* Short token sequences (3–12 tokens) tiled to fill the sequence length, training pattern completion;

(b) *Ascending sequences:* Monotonically increasing token indices with noise, training sequential prediction;

(c) *Alternating patterns:* Two-token alternation with 10% random substitution, training regularity detection;

(d) *Segmented frequency bands:* 3–6 segments with different frequency characteristics, training boundary detection;

(e) *Pure random:* Uniform random token sequences, providing a null contrast.

The diversity of pattern types drives competitive Hebbian learning to differentiate cortical columns into distinct functional specializations.

5.10.4. Phase 4: Oscillatory Calibration (20% of prenatal steps)

Multi-frequency oscillatory patterns are generated with explicit phase-amplitude coupling:

- Gamma-band oscillations (first quarter of embedding dimensions), amplitude 0.3;
- Theta-band oscillations (second quarter), amplitude 0.5;
- Alpha-band oscillations (third quarter), amplitude 0.3;
- Delta-band oscillations (remaining dimensions), amplitude 0.2.

Phase-amplitude coupling is implemented as gamma amplitude modulation by theta phase:

$$x_{\text{gamma}}(t) = x_{\text{gamma}}(t) \times (0.5 + 0.5 \cdot \sin(2\pi t/\tau_\theta)) \tag{24}$$

This calibrates the network's cross-frequency coupling parameters (PAC kappa values) before real data exposure.

5.10.5. Convergence Detection

Prenatal development convergence is detected over a sliding window (default: 50 steps) when both conditions are met:

1. Weight magnitude coefficient of variation $< 5\%$: $\text{CV}(\|\theta\|) < 0.05$;
2. Predictive coding loss trend change $< 2\%$: $\frac{|\bar{\mathcal{L}}_{\text{second_half}} - \bar{\mathcal{L}}_{\text{first_half}}|}{|\bar{\mathcal{L}}_{\text{first_half}}|} < 0.02$.

5.10.6. Prenatal Sleep Cycles

Fetal sleep cycles are known to exist in biological systems (Khazipov & Luhmann, 2006). The prenatal development module 195 checks the system's sleep trigger after each step and executes sleep phases when triggered. Working memory is reset between all prenatal steps (regardless of sleep) to prevent cross-contamination between self-generated patterns.

5.10.7. Prenatal Hippocampal Encoding

When hippocampal encoding is enabled, prenatal spontaneous activity is encoded into the hippocampal store with an `is_production=False` flag, distinguishing self-organized patterns from production experiences. This models fetal hippocampal spontaneous activity (Luhmann et al., 2016) that establishes initial place and pattern cell representations.

5.11. Trend-Based Self-Evaluation and Learning Rate Adaptation

Referring to FIG. P4-21, the self-evaluation module 197 periodically tests the system on held-out hippocampal memories and adapts learning parameters based on performance trends.

Self-evaluation occurs every N_{eval} steps (default: 200) and comprises:

1. **Memory retrieval:** Up to 5 hippocampal memories are retrieved as test data;
2. **Forward evaluation:** Each test memory is processed through cortical columns, and the mean squared reconstruction error is computed;
3. **History tracking:** The average error is appended to the performance history buffer;
4. **Trend detection:** The mean of the 3 most recent evaluations is compared against the mean of the 3 evaluations preceding them:
    - If recent $< 0.95\times$ older: **improving** — advance the developmental program by 50 steps, accelerating maturation;
    - If recent $> 1.05\times$ older: **degrading** — reduce the sleep interval by 10 steps (minimum 50), multiply the Hebbian learning rate by 0.8;

- If recent $> 1.2\times$ older: **emergency** — force $p_{\text{sleep}} = 1.0$, triggering immediate sleep consolidation;
- Otherwise: **stable** — maintain current parameters.

5. **Stagnation detection:** If the last 5 evaluations have a coefficient of variation $< 2\%$:
   - Multiply Hebbian learning rate by 1.5 (capped at $10\times$ the initial value);
   - Set $p_{\text{sleep}} = 0.8$, triggering near-immediate sleep consolidation to disrupt the local minimum.

5.12. Developmental Critical Periods

Referring to FIG. P4-24, the developmental program 196 manages four critical periods that systematically alter the character of learning across the system's lifetime.

5.12.1. Stage 1: Sensory (Steps 0–1,000)

High learning rate ($2\times$), low inhibition (0.3), high Hebbian weight (0.8), high structural plasticity (1.0), infrequent sleep ($2\times$ base interval). This stage models the infant's rapid statistical absorption, where cortical circuits are maximally plastic and receptive to all input patterns.

5.12.2. Stage 2: Association (Steps 1,000–5,000)

Moderate learning rate ($1.5\times$), moderate inhibition (0.6), moderate Hebbian weight (0.6), moderate structural plasticity (0.7), moderately infrequent sleep ($1.5\times$ base interval). This stage models the transition from raw sensory processing to associative learning, where inhibitory circuits begin to mature and impose selectivity.

5.12.3. Stage 3: Integration (Steps 5,000–20,000)

Normal learning rate ($1.0\times$), strong inhibition (0.9), low Hebbian weight (0.4), low structural plasticity (0.3), normal sleep frequency ($1.0\times$ base interval). This stage models mature learning, where error-driven predictive coding dominates over correlation-based Hebbian learning and consolidation becomes the primary mechanism for knowledge acquisition.

5.12.4. Stage 4: Refinement (Steps 20,000+)

Low learning rate ($0.5\times$), full inhibition (1.0), minimal Hebbian weight (0.3), minimal structural plasticity (0.1), frequent sleep ($0.7\times$ base interval). This stage models expert refinement, where the system makes fine-grained adjustments with maximal consolidation frequency and minimal structural change.

The developmental program 196 exposes the following parameter multipliers, which are consumed by the learning manager and autonomous loop controller:

- `get_lr_multiplier()`: Scales all learning rates;
- `get_inhibitory_strength()`: Modulates interneuron gating circuits;
- `get_hebbian_weight()`: Controls the Hebbian vs. error-driven balance;
- `get_structural_plasticity_rate()`: Gates column splitting, merging, and pruning operations;
- `get_sleep_interval()`: Adjusts the step-count sleep trigger threshold.

5.13. State Persistence and Resumption

The autonomous loop controller 120 supports full state serialization for checkpoint-resume workflows. The saved state comprises:

- EWC importance estimates and anchor parameters;
- Performance history buffer;
- Global step counter and wake steps since last sleep;
- Sleep pressure value and total experience count;
- Semantic knowledge accumulated through consolidation;
- Total consolidation, dream, self-play, and curiosity session counts;
- Self-play loss history;
- Curiosity score history (last 100 entries);
- Full autonomous configuration parameters.

This enables training to be interrupted and resumed without loss of autonomous system state, ensuring continuity of the adaptive learning strategy across training sessions.

5.14. Implementation Considerations

5.14.1. Memory Safety

All sleep-phase operations execute within a `torch.no_grad()` context, preventing autograd graph construction and reducing peak VRAM usage by approximately 3 GiB for a 106M-parameter model. All learning during sleep is analytical (direct parameter manipulation via `.data` attribute), requiring zero gradient storage.

5.14.2. OOM Resilience

The autonomous loop controller 120 implements OOM-resilient execution: the primary wake forward pass retries once after garbage collection and CUDA cache clearance, while auxiliary operations (self-play, curiosity, sleep, EWC estimation, self-evaluation) are wrapped in OOM exception handlers that skip the auxiliary operation rather than crashing the training run.

5.14.3. GPU Memory Shield Compatibility

The autonomous loop controller 120 avoids calling `torch.cuda.empty_cache()` during normal wake processing to maintain compatibility with external VRAM shield mechanisms that hold free GPU memory in PyTorch's caching allocator to prevent co-located processes on shared GPUs from stealing VRAM between steps. Cache clearance is only performed before memory-intensive auxiliary operations.

CLAIMS

We claim:

Claim 1. A computer-implemented system for autonomous self-directed learning in a neural network, comprising:

(a) a neural network model comprising a plurality of processing units configured to produce, for each input, a prediction output and a precision estimate representing the model's uncertainty;

(b) an always-on learning module configured to execute local synaptic updates after every forward pass of the neural network model, wherein the effective learning rate is modulated by at least a neuromodulatory attention signal, a neuromodulatory arousal signal, and a wake efficiency factor that decreases as a sleep pressure value increases;

(c) a sleep pressure accumulator configured to maintain a sleep pressure value in the range $[0, 1]$ that accumulates after each forward pass according to:

$$p(t) = \min\big(1.0,\ p(t{-}1) + \beta_{\text{base}} \cdot (1 + \min(\text{PE}, K))\big)$$

where β_{base} is a base increment, PE is the mean prediction error from the forward pass, and K is a saturation cap, such that high-surprise experiences accumulate sleep pressure faster than low-surprise experiences;

(d) a dual sleep trigger mechanism configured to initiate a sleep consolidation cycle when either the sleep pressure value exceeds a pressure threshold or the number of wake processing steps since the last sleep cycle exceeds a step-count threshold;

(e) a multi-phase sleep consolidation module configured to execute, upon triggering of the sleep cycle, at least:

(i) a generative replay phase wherein stored episodic memories are replayed through the processing units with local learning updates;

(ii) a dream learning phase wherein stored memories are retrieved, injected with noise, and processed through the processing units with a plurality of concurrent learning rules applied to dream-induced prediction errors; and

(iii) a synaptic downscaling phase wherein synaptic weight parameters are multiplicatively reduced by a configurable factor while excluding designated non-synaptic parameters from downscaling; and

(f) a lifecycle controller configured to orchestrate the always-on learning module, the sleep pressure accumulator, the dual sleep trigger mechanism, and the multi-phase sleep consoli-

dation module in a continuous autonomous loop, and to reset the sleep pressure value to zero after each sleep consolidation cycle.

Claim 2. The system of Claim 1, wherein the sleep pressure accumulator is further configured such that the base increment β_{base} is computed as $1 - \lambda_{\text{decay}}$ where λ_{decay} is a configurable decay factor, and the surprise multiplier $(1 + \min(\text{PE}, K))$ ranges from $1\times$ for zero prediction error to $(1 + K)\times$ for maximum prediction error, the saturation cap K being set to 5.0.

Claim 3. The system of Claim 1, wherein the dual sleep trigger mechanism is configured with a pressure threshold of 0.7, corresponding to approximately 70% of the sleep pressure saturation level, and the step-count threshold is a configurable sleep interval that serves as a failsafe against pathological insomnia in cases of sustained low-surprise input.

Claim 4. The system of Claim 1, wherein the dream learning phase of the multi-phase sleep consolidation module is configured to apply the following learning rules concurrently on each dream episode:

(i) a predictive coding update on prediction weights and precision parameters;

(ii) a feedforward network update on dendritic compartment weights using precision-weighted error signals;

(iii) a local parameter update on thalamic input, top-down modulation, thalamic feedback, and association output pathways;

(iv) a BCM Hebbian update with anti-Hebbian decorrelation on association weights; and

(v) a metaplasticity tracking update that records dream-induced weight change magnitude.

Claim 5. The system of Claim 1, wherein the non-synaptic parameters excluded from synaptic downscaling comprise gate set points, bias parameters, normalization gains, oscillatory period parameters, phase-amplitude coupling strengths, receptor thresholds and slopes, memory decay constants, and spatial phase parameters.

Claim 6. The system of Claim 1, wherein the always-on learning module is further configured to modulate both Hebbian learning rates and predictive coding learning rates by a wake efficiency factor computed as $\max(0.1,\ 1.0 - p_{\text{sleep}})$, such that all forms of synaptic plasticity diminish as sleep pressure accumulates.

Claim 7. The system of Claim 1, further comprising a curiosity drive module configured to:

(i) compute a curiosity score for candidate inputs as $c = w_{\text{curiosity}} \cdot \sigma^2_{\text{discourse}} + w_{\text{exploit}} \cdot \text{PE}$, where $\sigma^2_{\text{discourse}}$ is the precision estimate at a discourse processing scale and PE is the prediction error; and

(ii) generate novel training prompts by retrieving N episodic memories, concatenating their token-level representations, randomly permuting the temporal positions, extracting a prefix of configurable length, and forwarding through the neural network to produce token identifiers via greedy argmax decoding.

Claim 8. The system of Claim 7, wherein the curiosity drive module is further configured to rank a plurality of candidate inputs by their respective curiosity scores and to select the top-k most informative candidates for processing, implementing an active inference strategy that preferentially seeks inputs that would maximally reduce the system's uncertainty.

Claim 9. The system of Claim 1, further comprising a self-play module configured to:

(i) generate a continuation of configurable length from a seed prompt using the neural network model;

(ii) re-process the generated continuation through the neural network model as input;

(iii) compute a self-play loss measuring the prediction error on self-generated text; and

(iv) apply local learning rules to reduce self-inconsistency,

wherein the self-play module further comprises an adaptive frequency controller that reduces the self-play frequency by a multiplicative factor when a configurable number of consecutive self-play losses exceed a gibberish threshold.

Claim 10. The system of Claim 9, wherein the adaptive frequency controller is configured such that if three consecutive self-play losses exceed a threshold of 15.0, the self-play interval is multiplied by 5, and normal frequency resumes when any subsequent self-play loss falls below the threshold.

Claim 11. The system of Claim 1, further comprising an activation-based continual learning guard configured to:

(i) estimate parameter importance without backpropagation according to $\hat{F}_i = \mathbb{E}[|\theta_i|^2 \cdot |\text{PE}_l|^2]$, where $|\theta_i|^2$ is the squared parameter magnitude and $|\text{PE}_l|^2$ is the squared mean prediction error at the processing unit containing parameter i;

(ii) store current parameter values as anchor parameters after each importance estimation;

(iii) apply an EWC penalty during learning computed as $\mathcal{L}_{\text{EWC}} = \frac{\lambda}{2} \sum_i \hat{F}_i \cdot (\theta_i - \theta_i^*)^2$; and

(iv) apply exponential decay to importance estimates over time to allow gradual relearning.

Claim 12. The system of Claim 11, wherein the importance estimation is performed on hippocampal episodic memories retrieved during wake processing, requiring only forward-pass activations and prediction errors with zero autograd computation.

Claim 13. The system of Claim 1, wherein the multi-phase sleep consolidation module further comprises a semantic consolidation phase configured to:

(i) cluster similar episodic memories using cosine similarity with a minimum cluster size;

(ii) extract centroid patterns as semantic summaries for each cluster;

(iii) compute a priority score for each centroid as priority $=$ confidence $\times$ schema_boost;

(iv) execute $n_{\text{passes}} = \text{clamp}(5 \times \text{priority}, 3, 8)$ training passes on each centroid at a learning rate of $\text{lr}_{\text{base}} \times$ priority; and

(v) remove consolidated episodes from the episodic store.

Claim 14. The system of Claim 13, wherein working memory subsystems are reset before semantic consolidation to prevent cross-contamination between sequential semantic patterns, and wherein each training pass applies BCM Hebbian learning with anti-Hebbian decorrelation, predictive coding local updates, feedforward network updates, and local parameter updates.

Claim 15. A computer-implemented method for prenatal self-organization of a neural network prior to external data exposure, comprising the steps of:

(a) executing a first developmental phase comprising generating temporally-correlated continuous activation patterns through random walks with momentum in an embedding space, the momentum factor ensuring temporal autocorrelation, and processing the generated patterns through a plurality of neural processing units with local learning updates;

(b) executing a second developmental phase comprising generating token sequences autoregressively from random seeds using the neural network, and processing the self-generated sequences as training input with shifted targets and local learning;

(c) executing a third developmental phase comprising generating a plurality of structurally distinct pattern types including repeated motifs, ascending sequences, alternating patterns, segmented frequency bands, and random sequences, and processing these diverse patterns through the neural processing units to induce competitive functional differentiation;

(d) executing a fourth developmental phase comprising generating multi-frequency oscillatory patterns with phase-amplitude coupling, wherein a low-frequency oscillation modulates the amplitude of a high-frequency oscillation, and processing the oscillatory patterns through the neural processing units to calibrate cross-frequency coupling parameters; and

(e) monitoring convergence by computing weight statistics over a sliding window and terminating prenatal development when weight magnitude coefficient of variation falls below a first threshold and predictive coding loss trend change falls below a second threshold.

Claim 16. The method of Claim 15, wherein the first developmental phase allocates 30% of total prenatal steps, the second developmental phase allocates 25%, the third developmental phase allocates 25%, and the fourth developmental phase allocates 20%.

Claim 17. The method of Claim 15, wherein the temporally-correlated activation patterns of the first developmental phase are generated according to:

$$v_t = \mu \cdot v_{t-1} + (1 - \mu) \cdot \mathcal{N}(0, \sigma^2 \cdot I_d)$$
$$h_t = h_{t-1} + v_t$$

where μ is a momentum coefficient, σ is a noise scale, d is the embedding dimension, and the resulting sequence is normalized via layer normalization.

Claim 18. The method of Claim 15, further comprising executing sleep phases during prenatal development when a sleep trigger condition is met, modeling fetal sleep cycles, and resetting working memory subsystems between all prenatal steps to prevent cross-contamination between self-generated patterns.

Claim 19. The method of Claim 15, further comprising encoding prenatal spontaneous activity into a hippocampal episodic memory store with a flag distinguishing self-organized patterns from production experiences, establishing initial pattern cell representations prior to external data exposure.

Claim 20. The method of Claim 15, wherein the convergence monitoring of step (e) uses a sliding window of at least 50 steps, the first threshold is 5% coefficient of variation, and the second threshold is 2% loss trend change.

Claim 21. A computer-implemented method for adaptive self-directed learning in a neural network, comprising the steps of:

(a) processing input data through the neural network during a wake phase, and executing local learning updates after each forward pass with a learning rate modulated by a sleep pressure value that accumulates proportionally to prediction error;

(b) periodically evaluating the neural network on held-out episodic memories by computing a reconstruction error metric, and storing the metric in a performance history buffer;

(c) comparing a mean of the most recent evaluation metrics against a mean of earlier evaluation metrics to determine a performance trend, and executing corrective actions based on the trend, wherein:

  (i) if the recent mean is below a first ratio of the earlier mean, advancing a developmental program that controls learning parameter scheduling;

  (ii) if the recent mean exceeds a second ratio of the earlier mean, reducing a sleep interval parameter and reducing a learning rate by a multiplicative factor;

(iii) if the recent mean exceeds a third ratio of the earlier mean, forcing the sleep pressure value to a maximum to trigger immediate sleep consolidation; and

(iv) if the coefficient of variation of recent evaluations falls below a stagnation threshold, increasing the learning rate by a multiplicative factor and elevating sleep pressure to trigger consolidation;

(d) managing a developmental staging system comprising a plurality of critical periods, each critical period defining a distinct set of parameter multipliers for learning rate, inhibitory strength, Hebbian learning weight, structural plasticity rate, and sleep frequency, the developmental staging system progressing through stages as a function of cumulative training steps.

Claim 22. The method of Claim 21, wherein the first ratio is 0.95, the second ratio is 1.05, and the third ratio is 1.2, the corrective actions for the degrading condition comprise reducing the sleep interval by a fixed decrement with a minimum floor and multiplying the learning rate by 0.8, and the corrective action for the emergency condition comprises setting the sleep pressure value to 1.0.

Claim 23. The method of Claim 21, wherein the stagnation threshold is a coefficient of variation of 2% computed over the most recent 5 evaluations, and the corrective action comprises increasing the learning rate by a factor of 1.5 subject to a cap of 10 times the initial learning rate, and setting sleep pressure to 0.8.

Claim 24. The method of Claim 21, wherein the developmental staging system comprises four critical periods:

(i) a sensory period with a learning rate multiplier of 2.0, an inhibitory strength of 0.3, a Hebbian weight of 0.8, and a sleep interval multiplier of 2.0;

(ii) an association period with a learning rate multiplier of 1.5, an inhibitory strength of 0.6, a Hebbian weight of 0.6, and a sleep interval multiplier of 1.5;

(iii) an integration period with a learning rate multiplier of 1.0, an inhibitory strength of 0.9, a Hebbian weight of 0.4, and a sleep interval multiplier of 1.0; and

(iv) a refinement period with a learning rate multiplier of 0.5, an inhibitory strength of 1.0, a Hebbian weight of 0.3, and a sleep interval multiplier of 0.7.

Claim 25. The method of Claim 21, wherein the periodic evaluation comprises retrieving up to 5 episodic memories from a hippocampal memory store, processing each memory through a plurality of cortical column processing units, computing mean squared reconstruction error between the input and the output prediction, and the performance trend comparison uses a window of 3 recent evaluations compared against 3 earlier evaluations.

ABSTRACT

A computer-implemented system and method for autonomous self-directed learning in neural networks, comprising an always-on learning module that executes local synaptic updates after every forward pass with neuromodulator-gated learning rates, a surprise-weighted sleep pressure accumulator that models adenosine build-up proportionally to prediction error, a dual sleep trigger mechanism combining physiological pressure threshold with step-count failsafe, and a multi-phase sleep consolidation cycle comprising generative replay, dream learning with five concurrent learning rules, confidence-weighted semantic consolidation transferring knowledge from episodic to cortical stores, and synaptic downscaling. The system further comprises curiosity-driven active inference for self-directed data acquisition through cross-episode memory recombination, self-play learning with adaptive frequency control, activation-based elastic weight consolidation requiring zero backpropagation, four-phase prenatal self-organization establishing functional connectivity before external data exposure, trend-based self-evaluation with emergency sleep triggers, and developmental critical periods modulating learning characteristics across the system's lifetime.

DRAWING SHEETS

FIG. 11

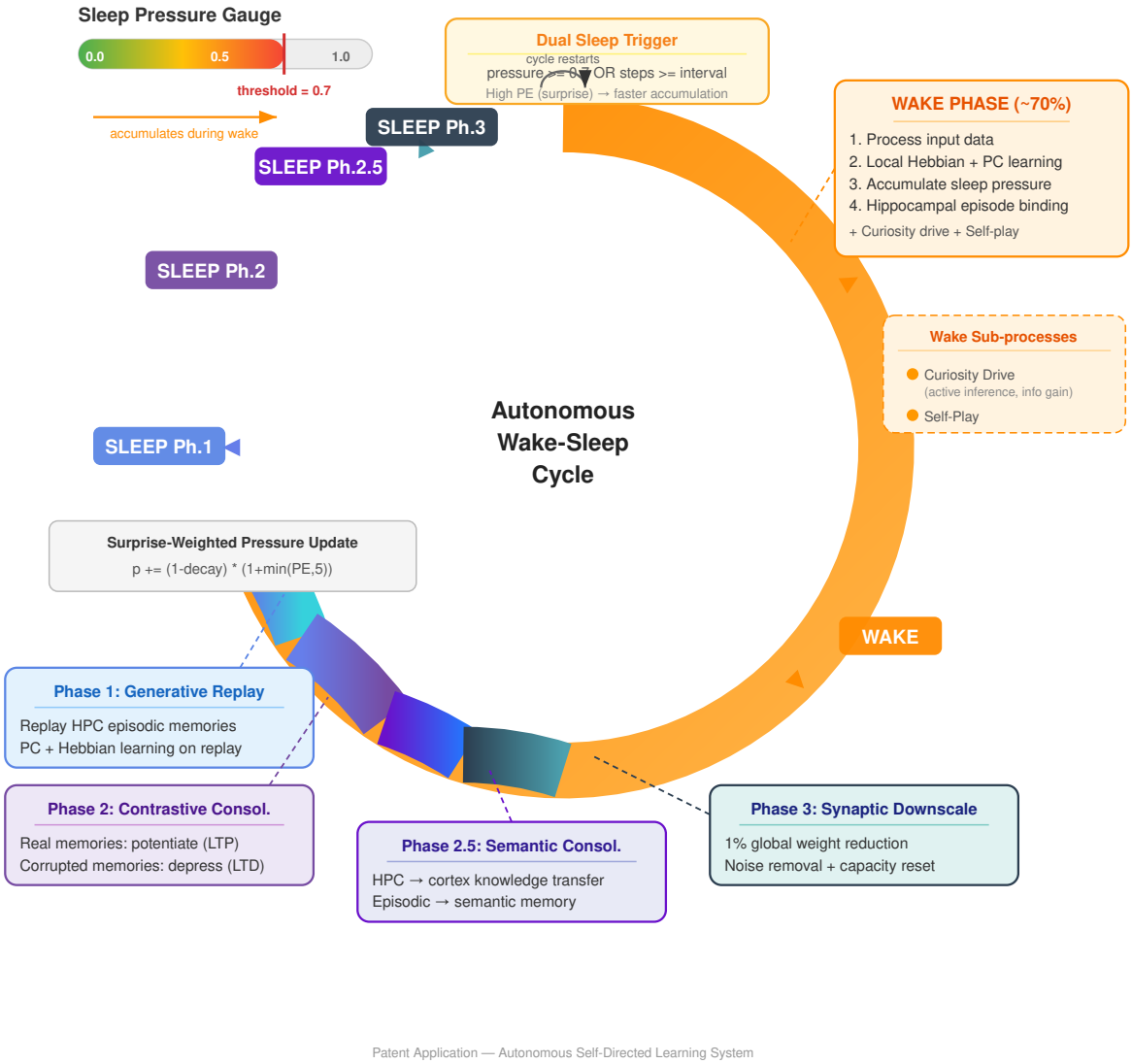

Figure 1: Wake-Sleep Autonomous Learning Lifecycle

FIG. 12

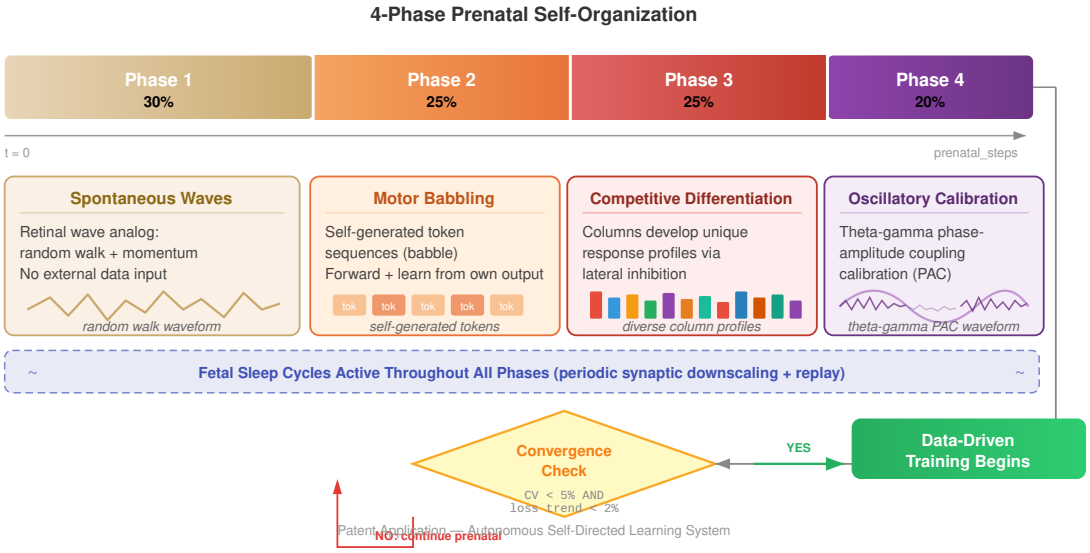

Figure 2: Four-Phase Prenatal Self-Organization

FIG. 13

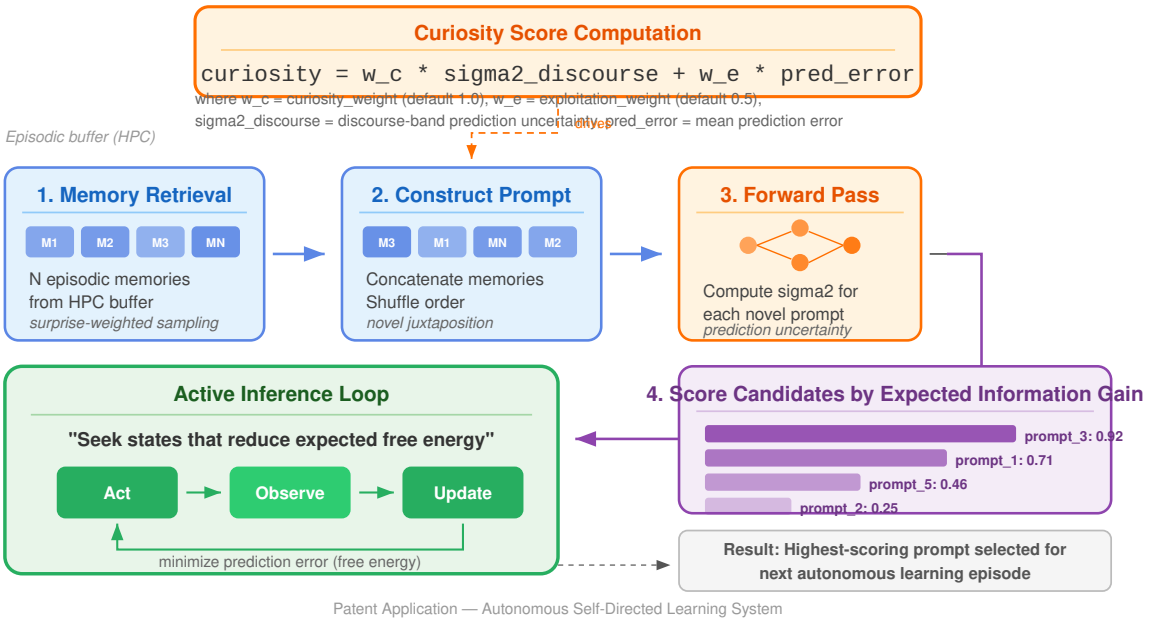

Figure 3: Curiosity-Driven Active Inference Module

CLAIM ILLUSTRATION SHEETS

FIG. P4-1

Autonomous Self-Directed Learning System (Claim 1)

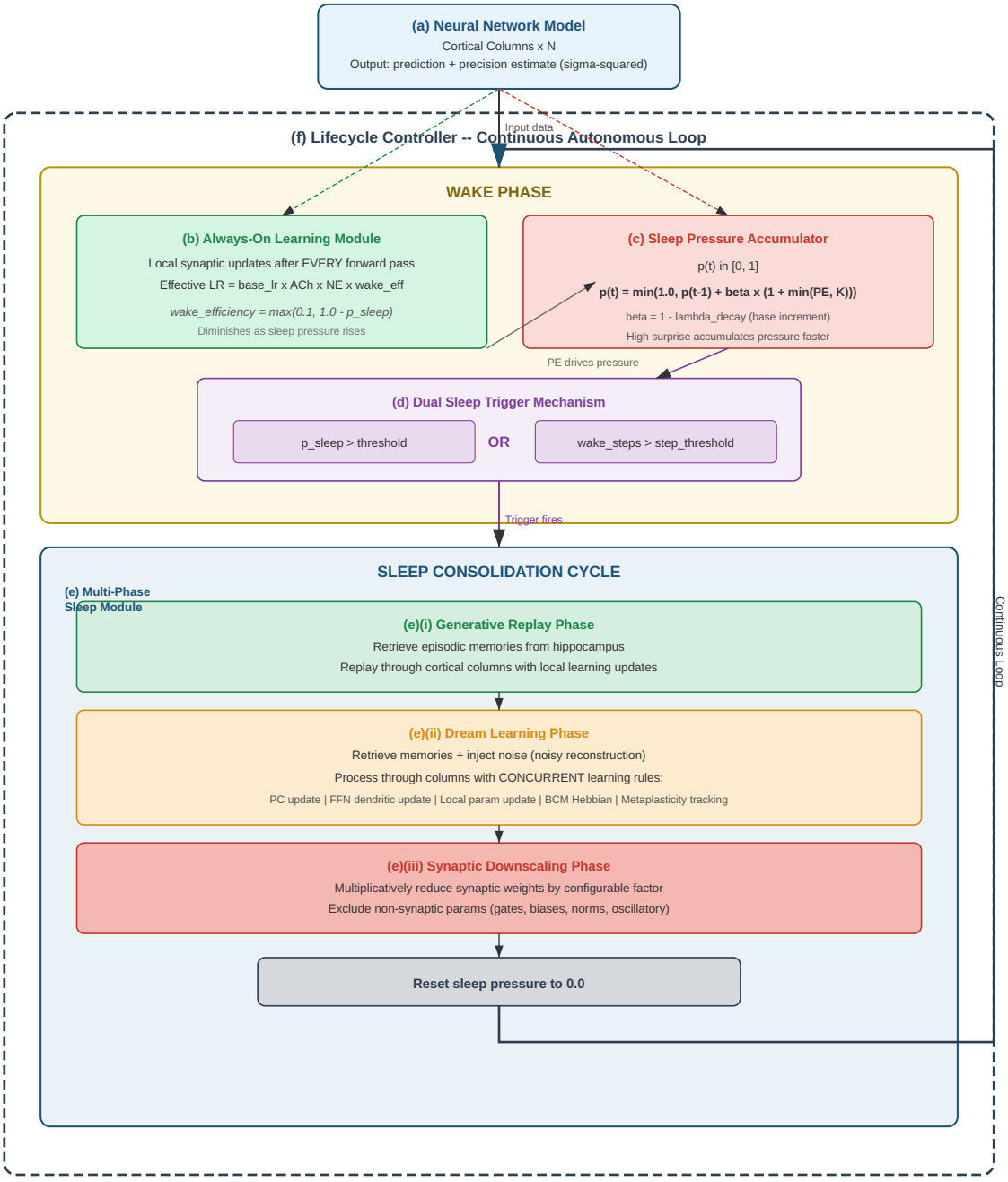

Figure 4: FIG. P4-1 — Autonomous Self-Directed Learning System (Claim 1)



Figure 5: FIG. P4-2 — Sleep Pressure Accumulation Formula (Claim 2)

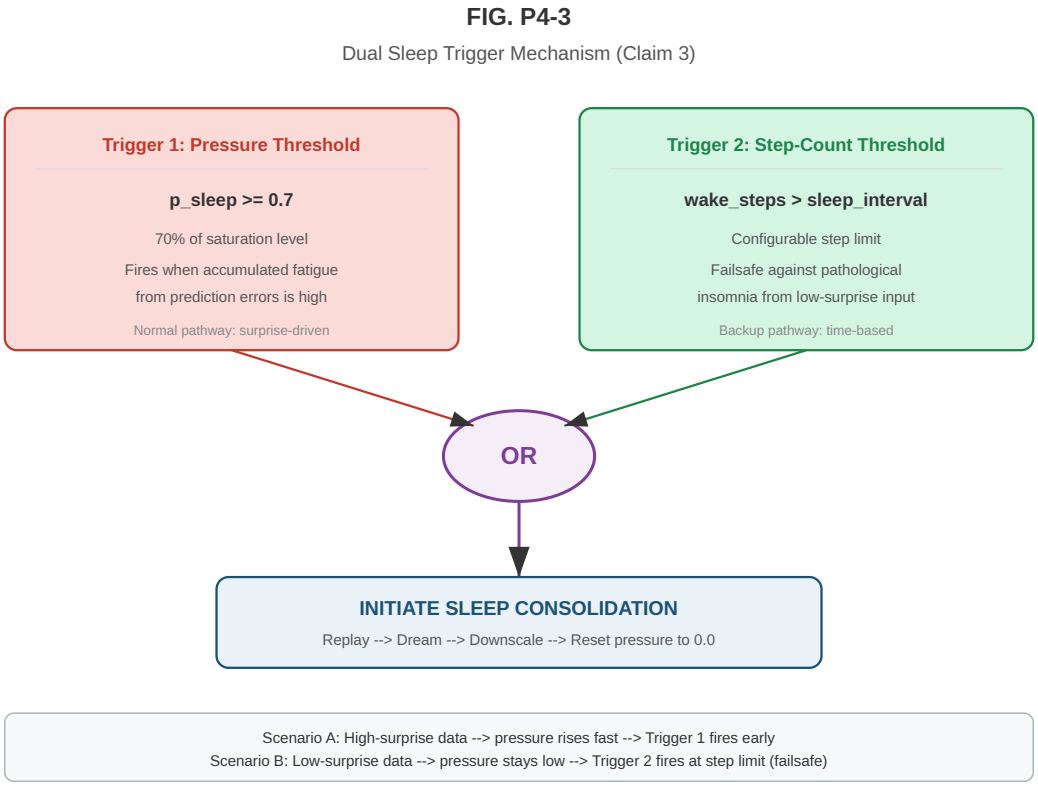

Figure 6: FIG. P4-3 — Dual Sleep Trigger Mechanism (Claim 3)

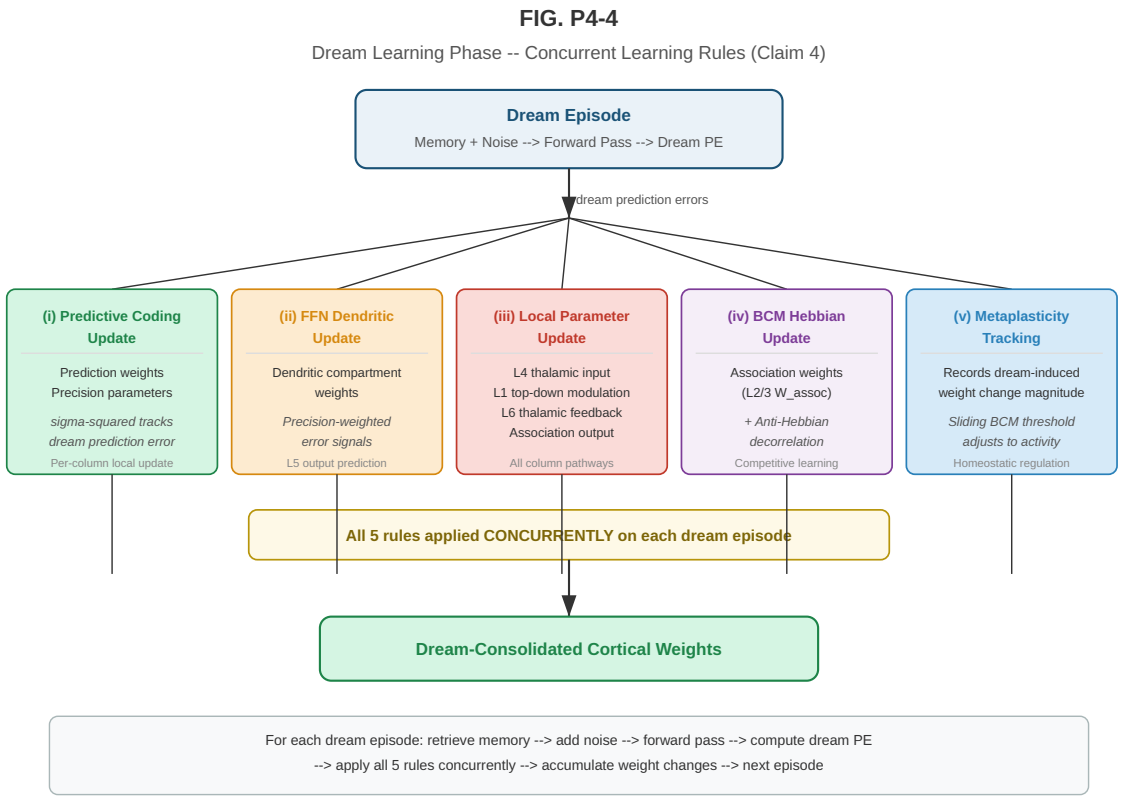

Figure 7: FIG. P4-4 — Five Concurrent Dream Learning Rules (Claim 4)

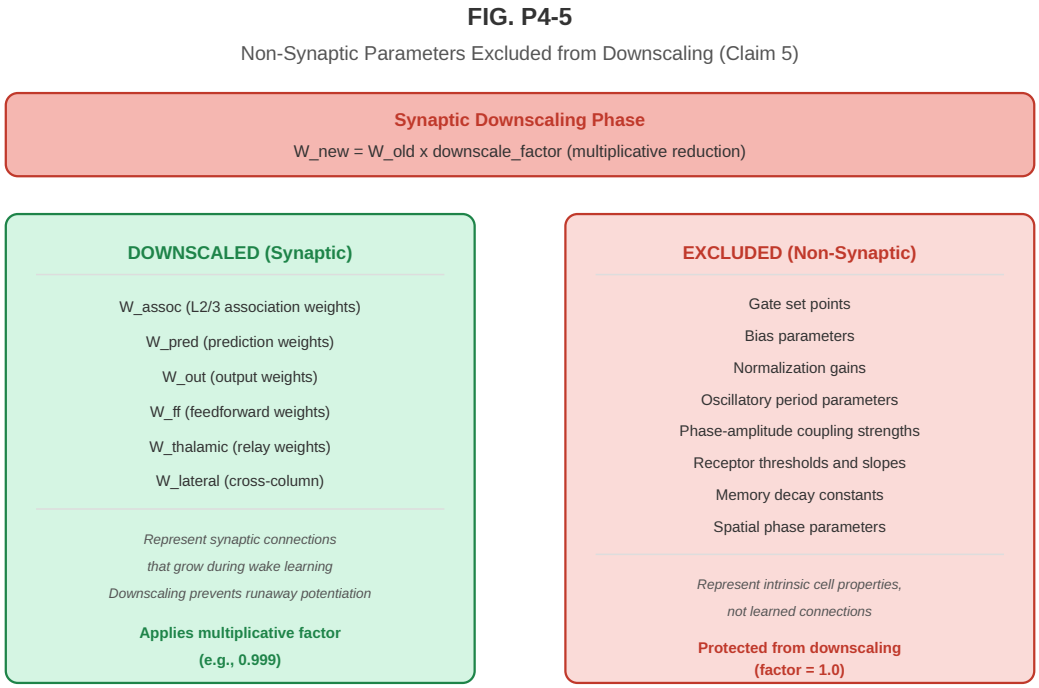

Figure 8: FIG. P4-5 — Non-Synaptic Parameters Excluded from Downscaling (Claim 5)

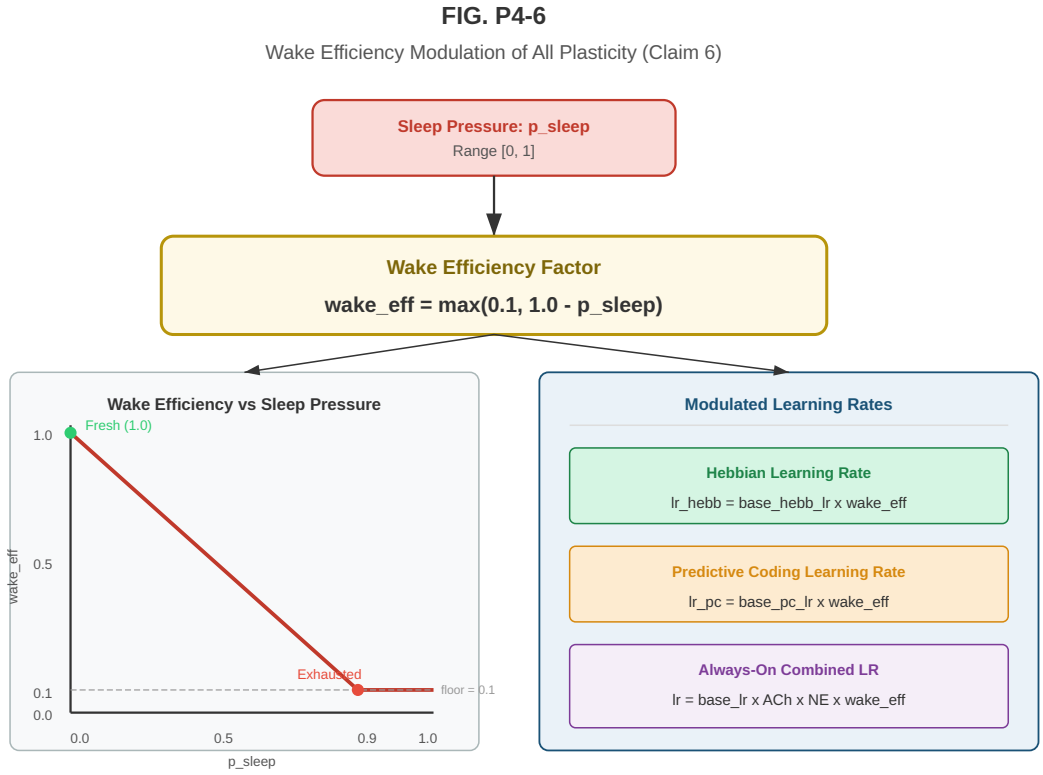

Figure 9: FIG. P4-6 — Wake Efficiency Modulation (Claim 6)

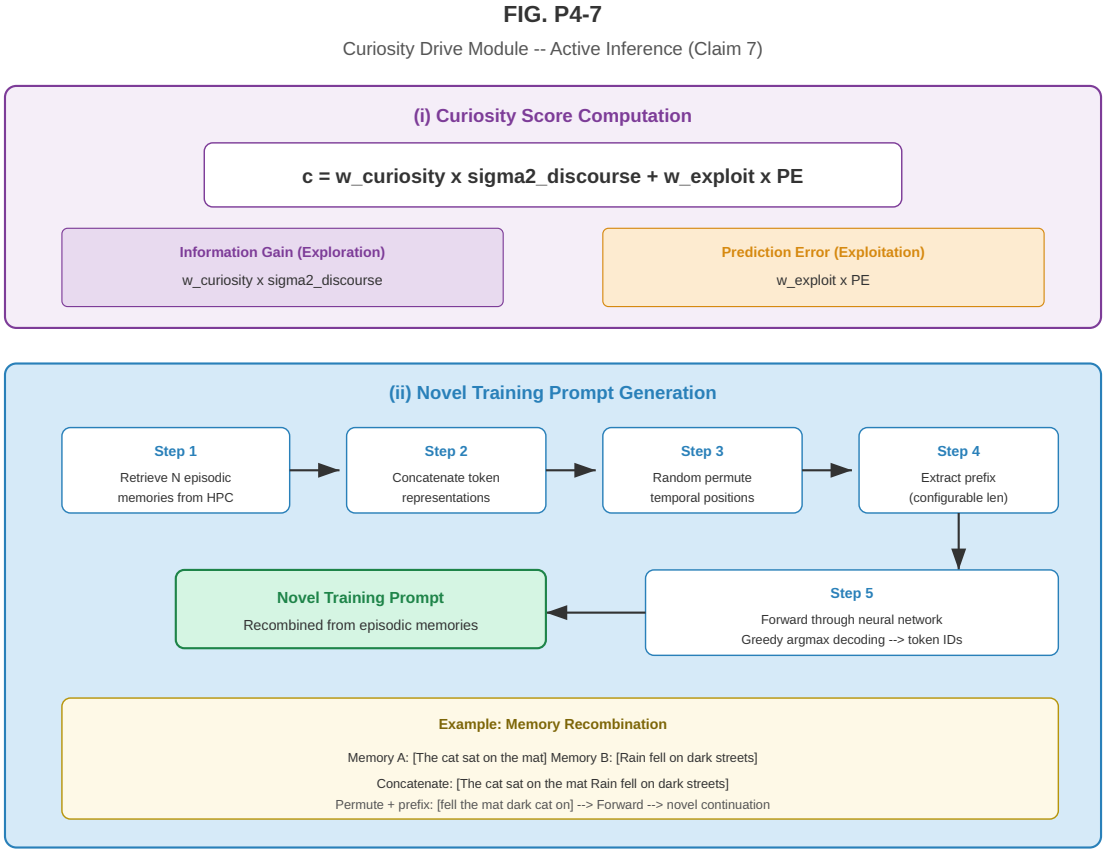

Figure 10: FIG. P4-7 — Curiosity Drive Module (Claim 7)

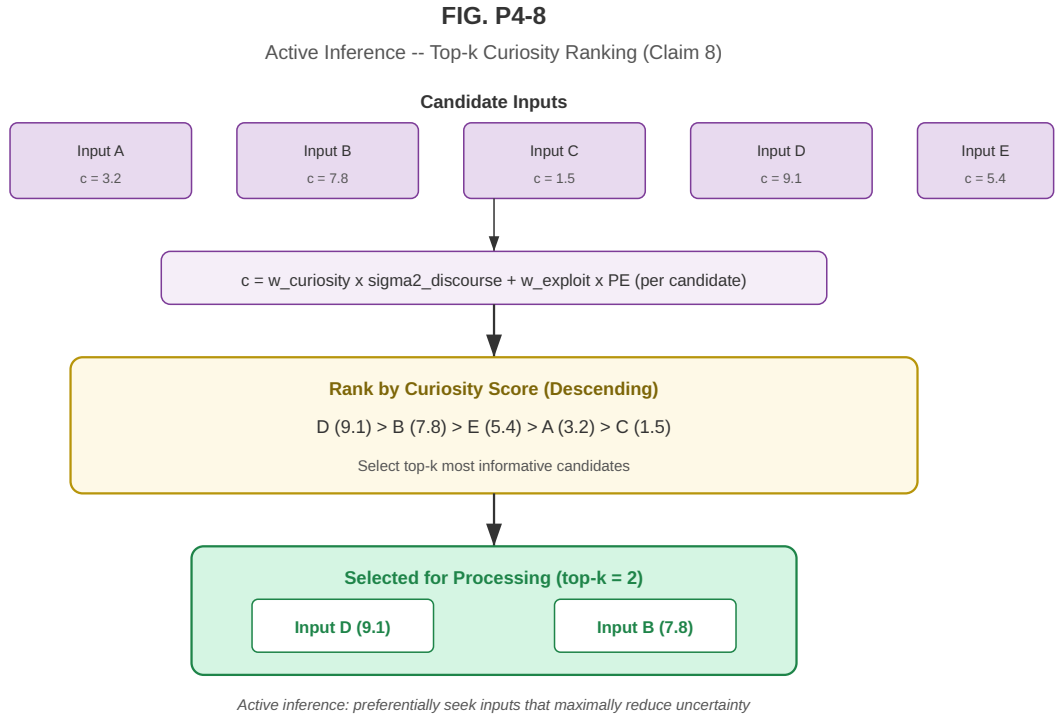

Figure 11: FIG. P4-8 — Top-k Curiosity Ranking (Claim 8)

FIG. P4-9

Self-Play Module -- Generate-Perceive-Evaluate-Learn (Claim 9)

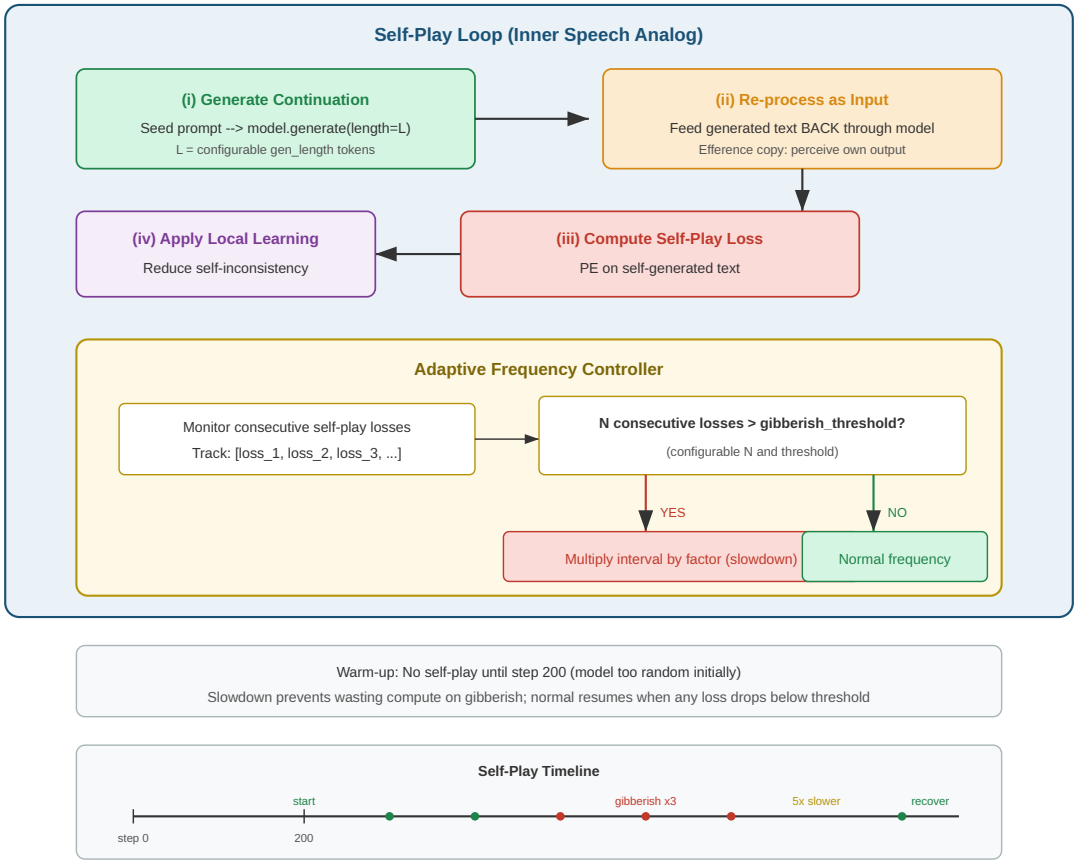

Figure 12: FIG. P4-9 — Self-Play Module (Claim 9)

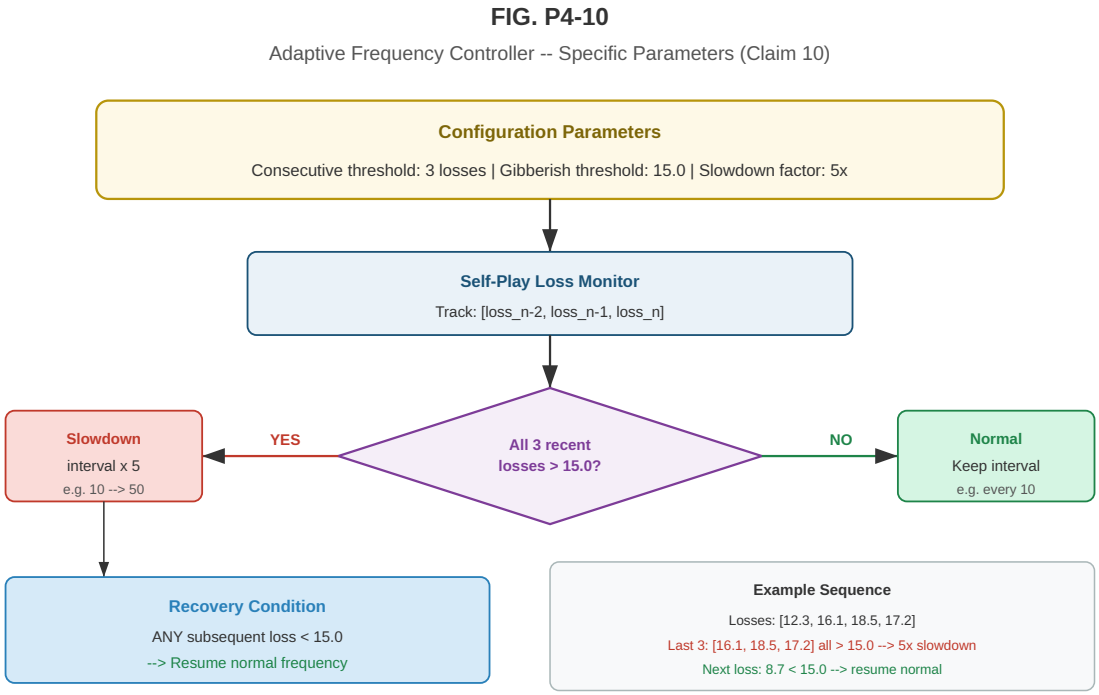

Figure 13: FIG. P4-10 — Adaptive Frequency Controller (Claim 10)

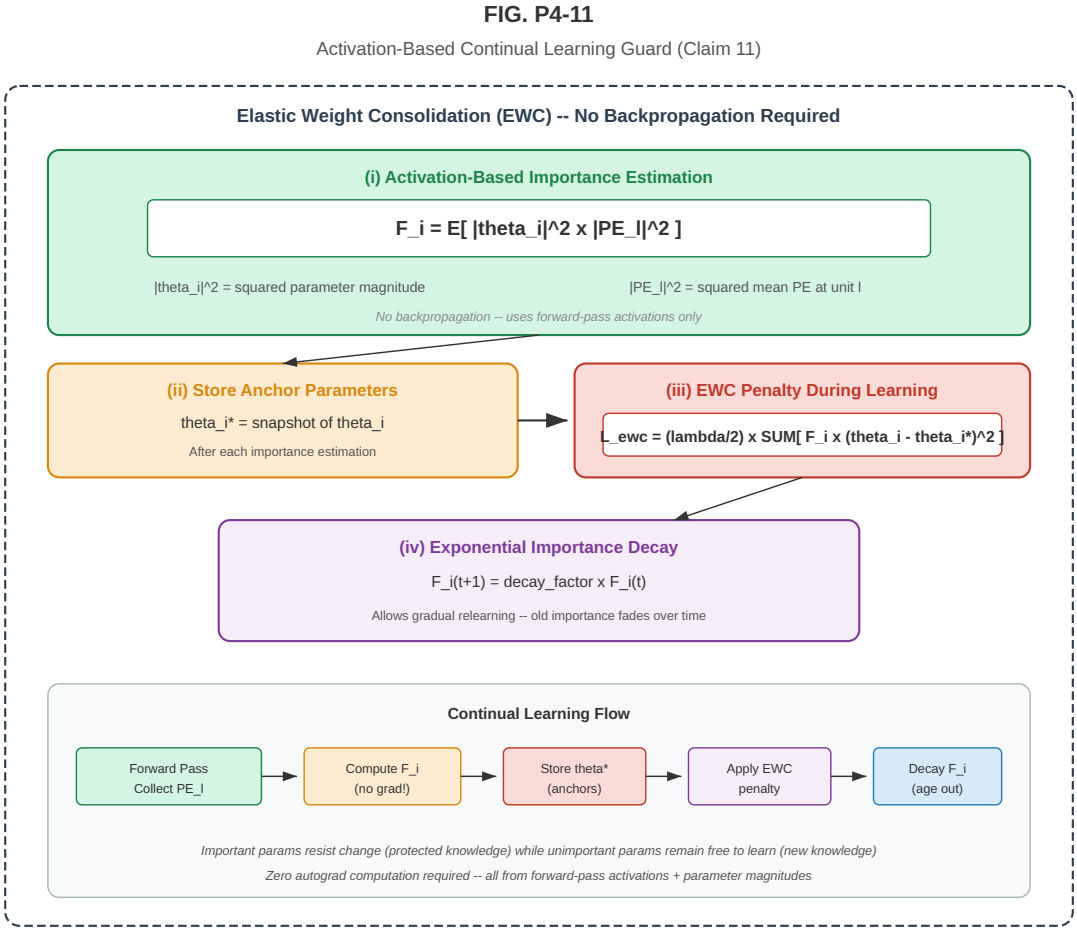

Figure 14: FIG. P4-11 — Activation-Based Continual Learning Guard (Claim 11)

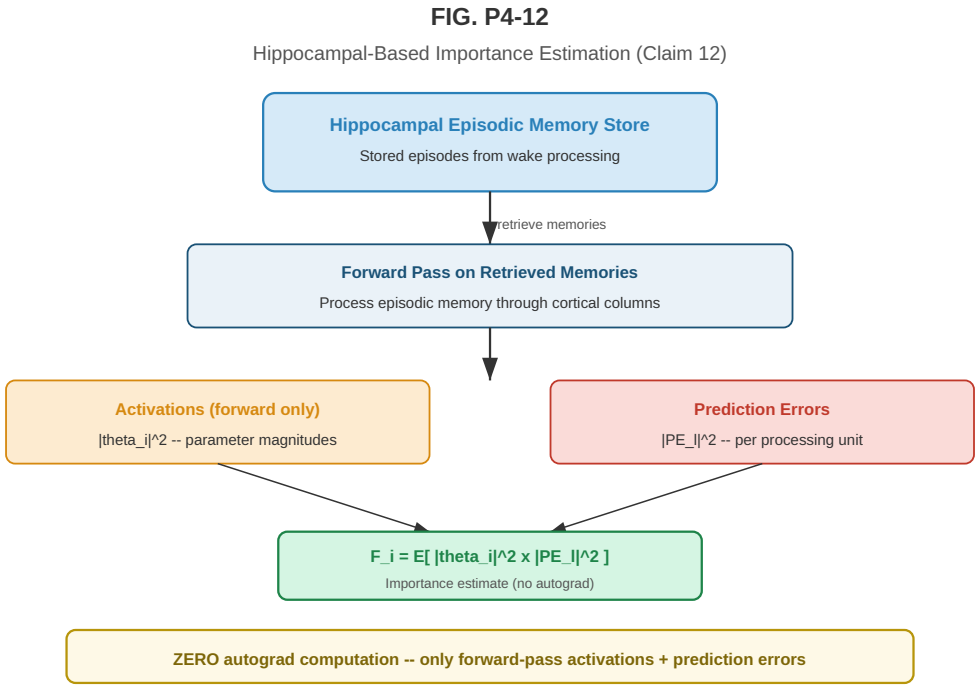

Figure 15: FIG. P4-12 — Hippocampal-Based Importance Estimation (Claim 12)

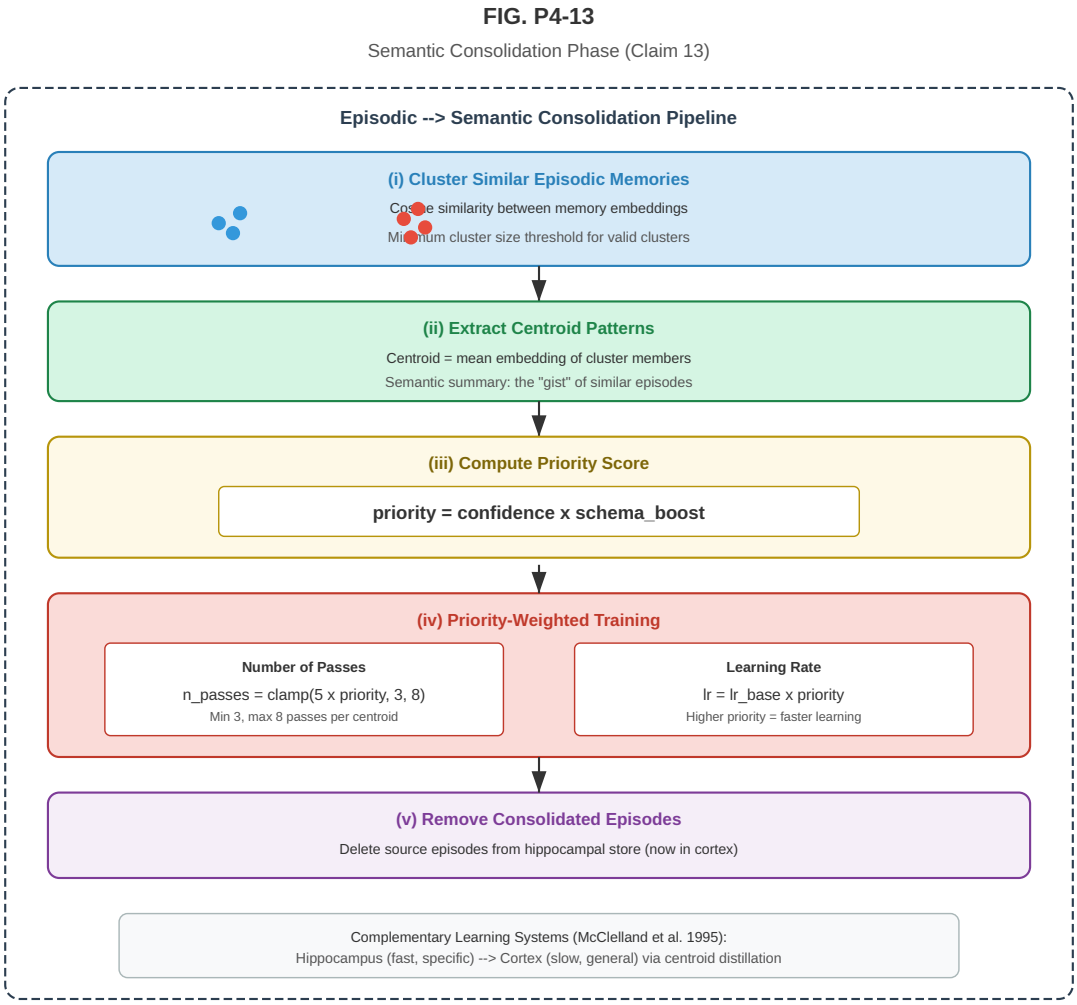

Figure 16: FIG. P4-13 — Semantic Consolidation Phase (Claim 13)

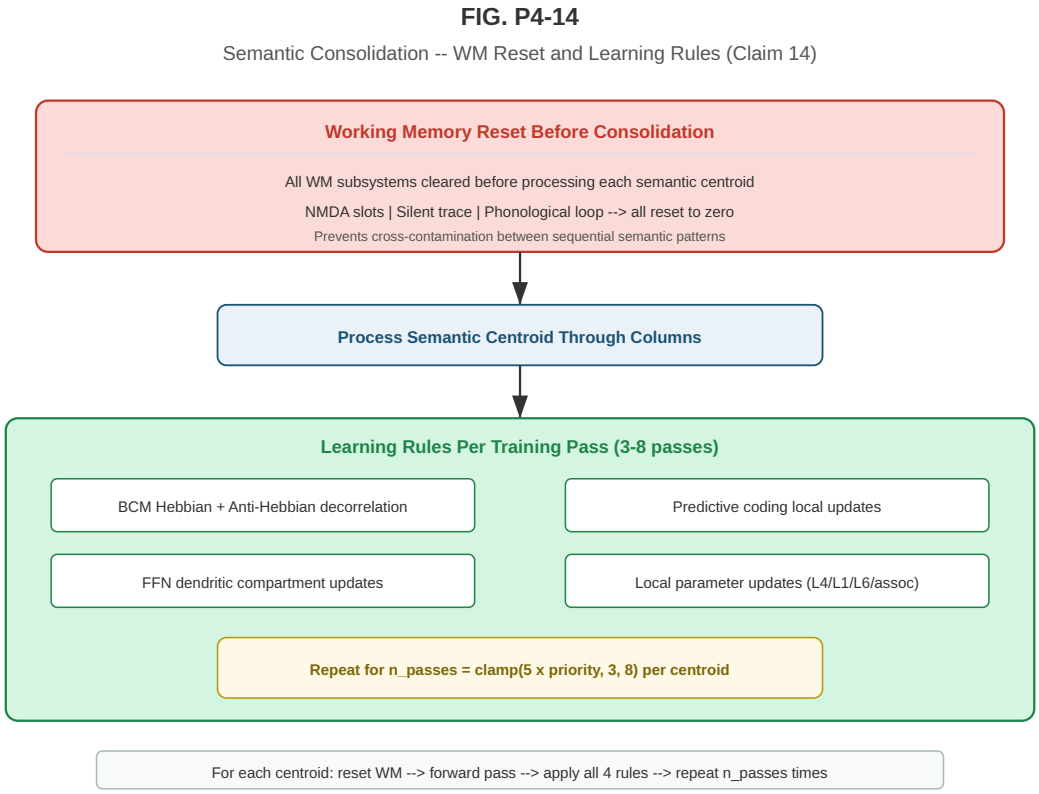

Figure 17: FIG. P4-14 — Semantic Consolidation Learning Rules (Claim 14)

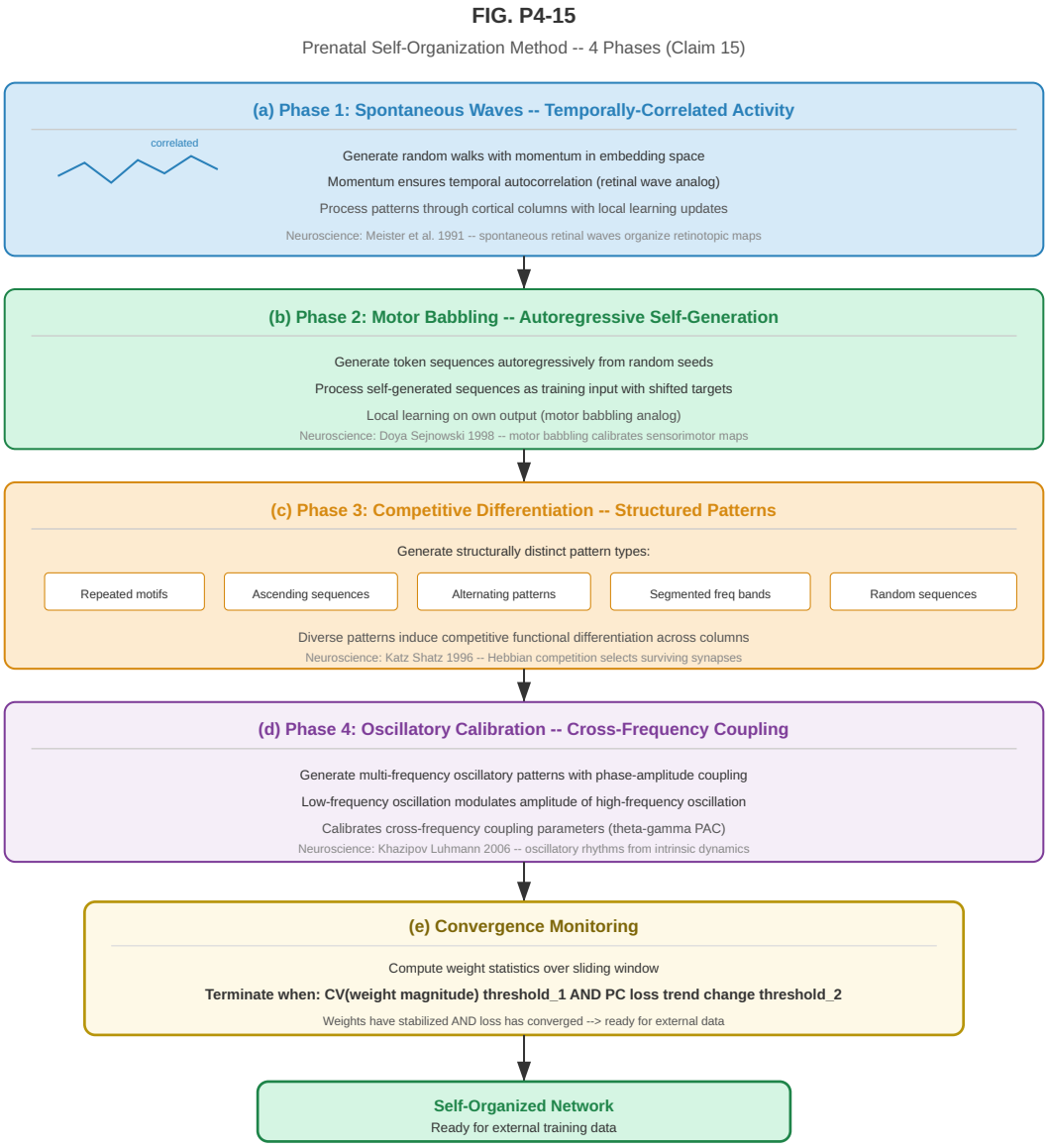

Figure 18: FIG. P4-15 — Prenatal Self-Organization Method (Claim 15)

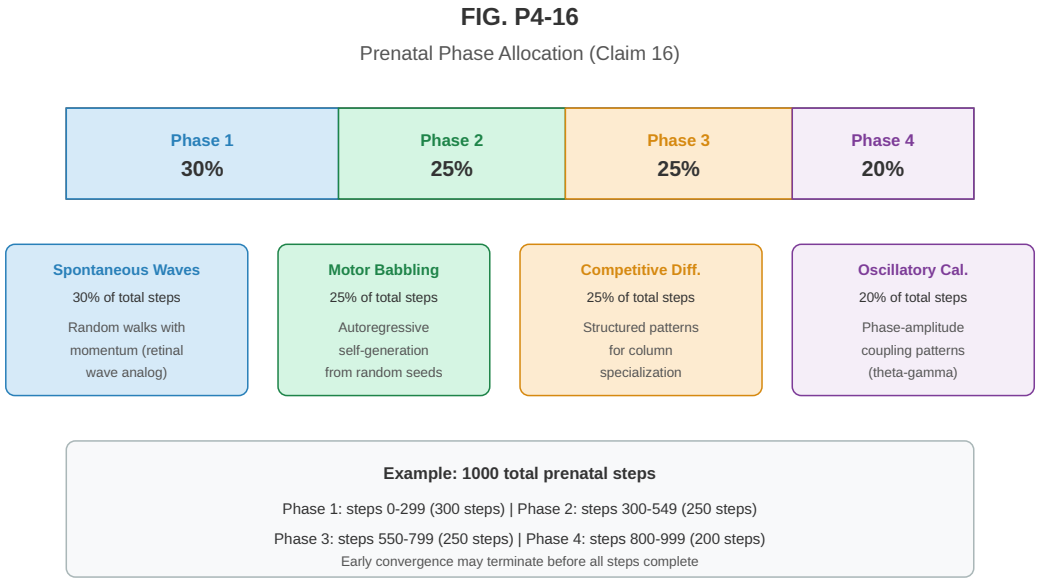

Figure 19: FIG. P4-16 — Prenatal Phase Allocation (Claim 16)

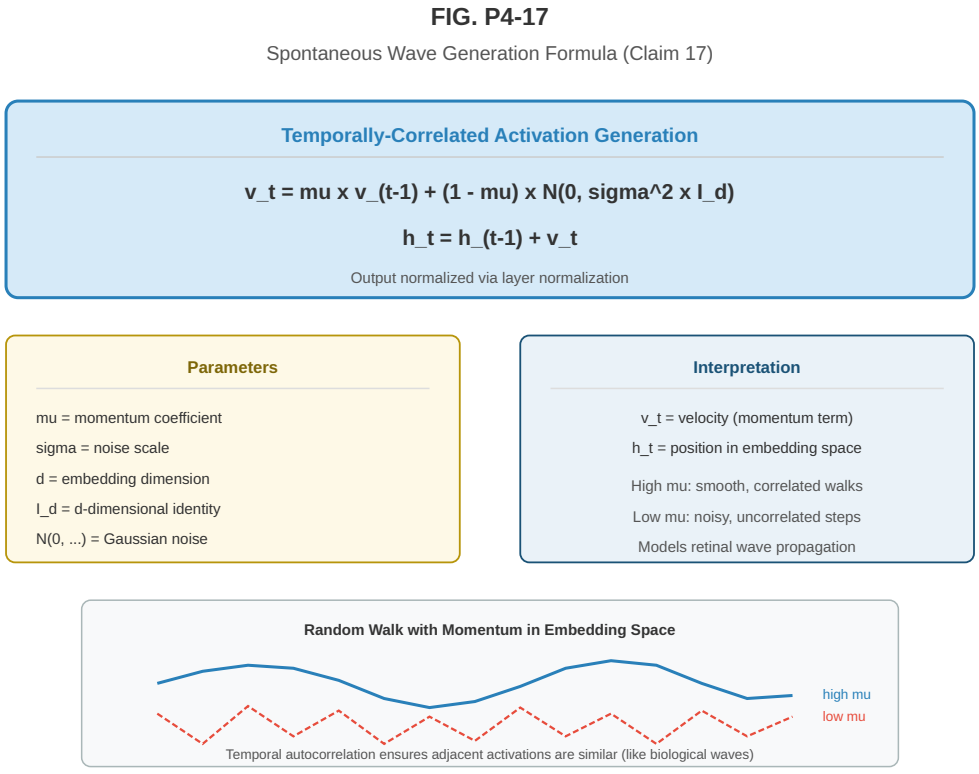

Figure 20: FIG. P4-17 — Spontaneous Wave Generation Formula (Claim 17)

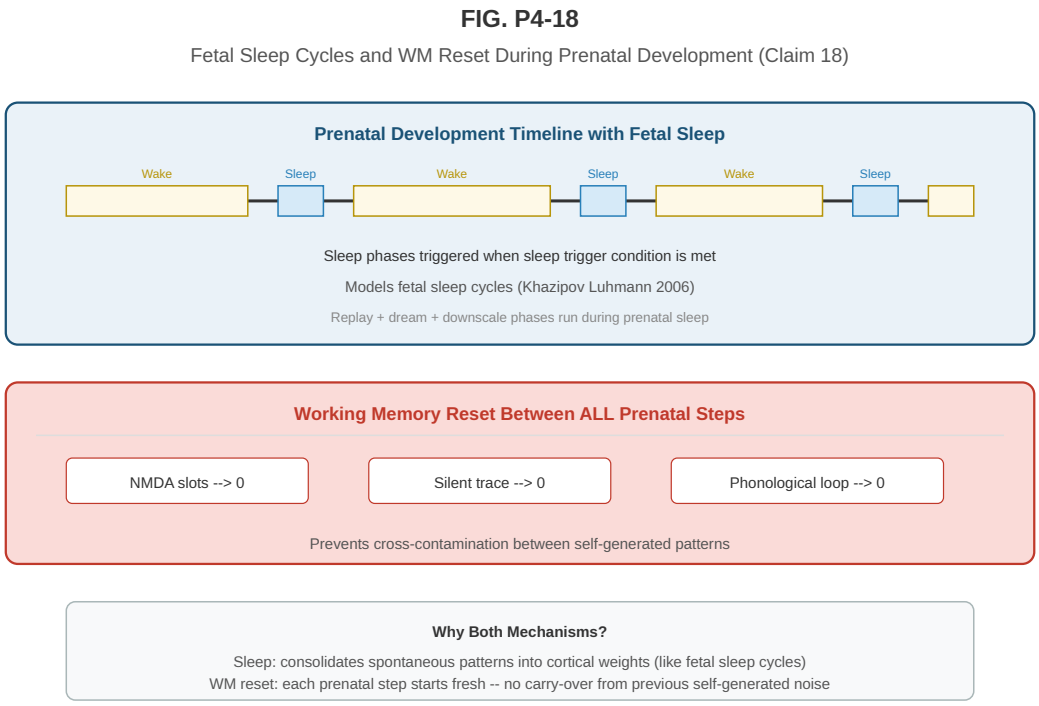

Figure 21: FIG. P4-18 — Fetal Sleep Cycles During Prenatal Development (Claim 18)

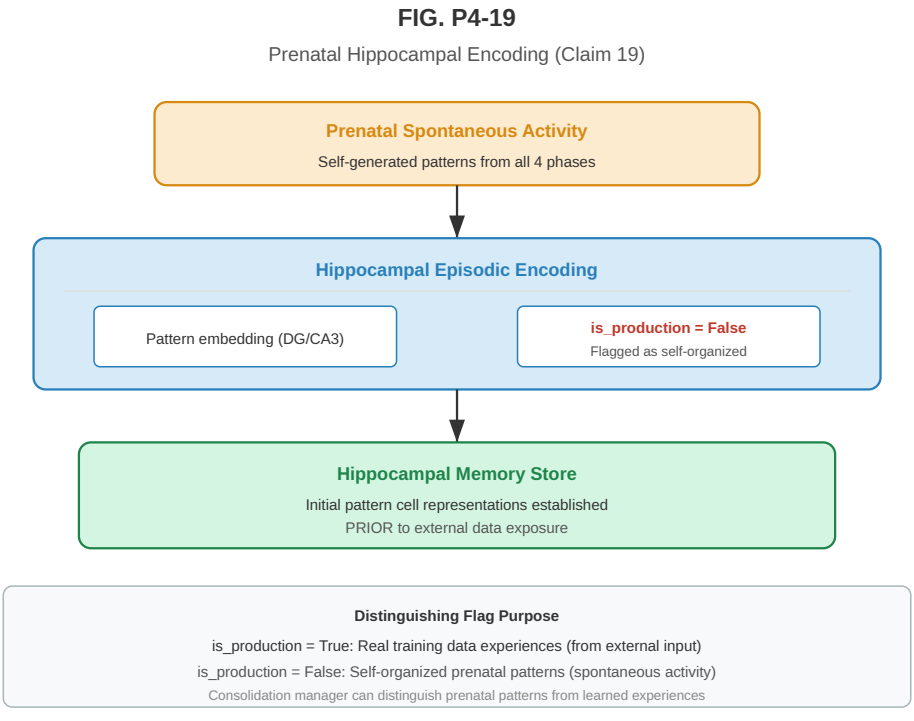

Figure 22: FIG. P4-19 — Prenatal Hippocampal Encoding (Claim 19)

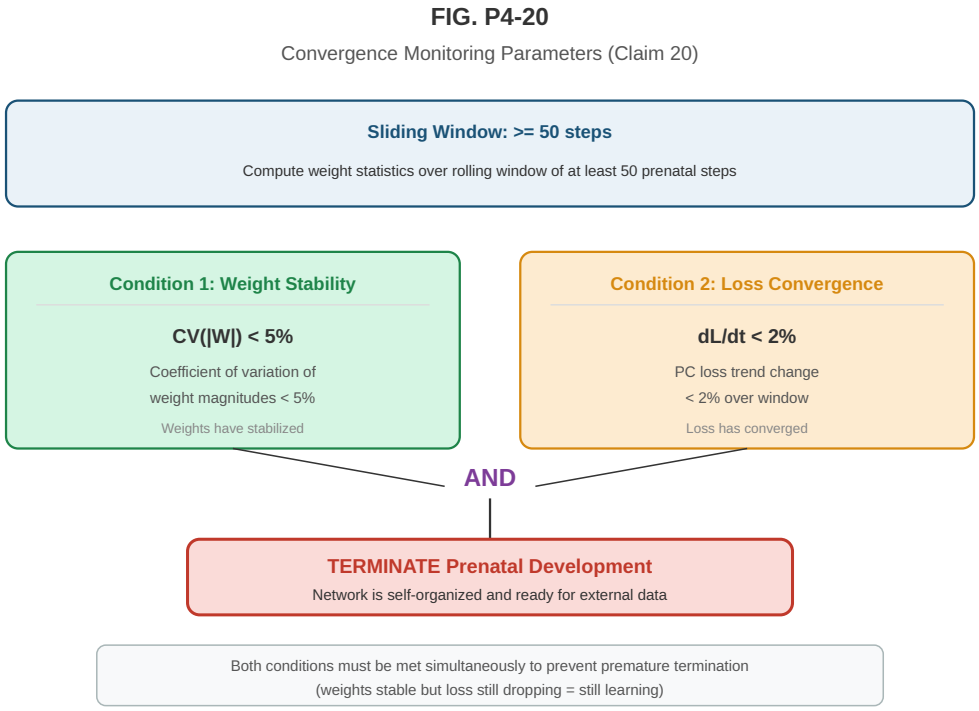

Figure 23: FIG. P4-20 — Convergence Monitoring Parameters (Claim 20)

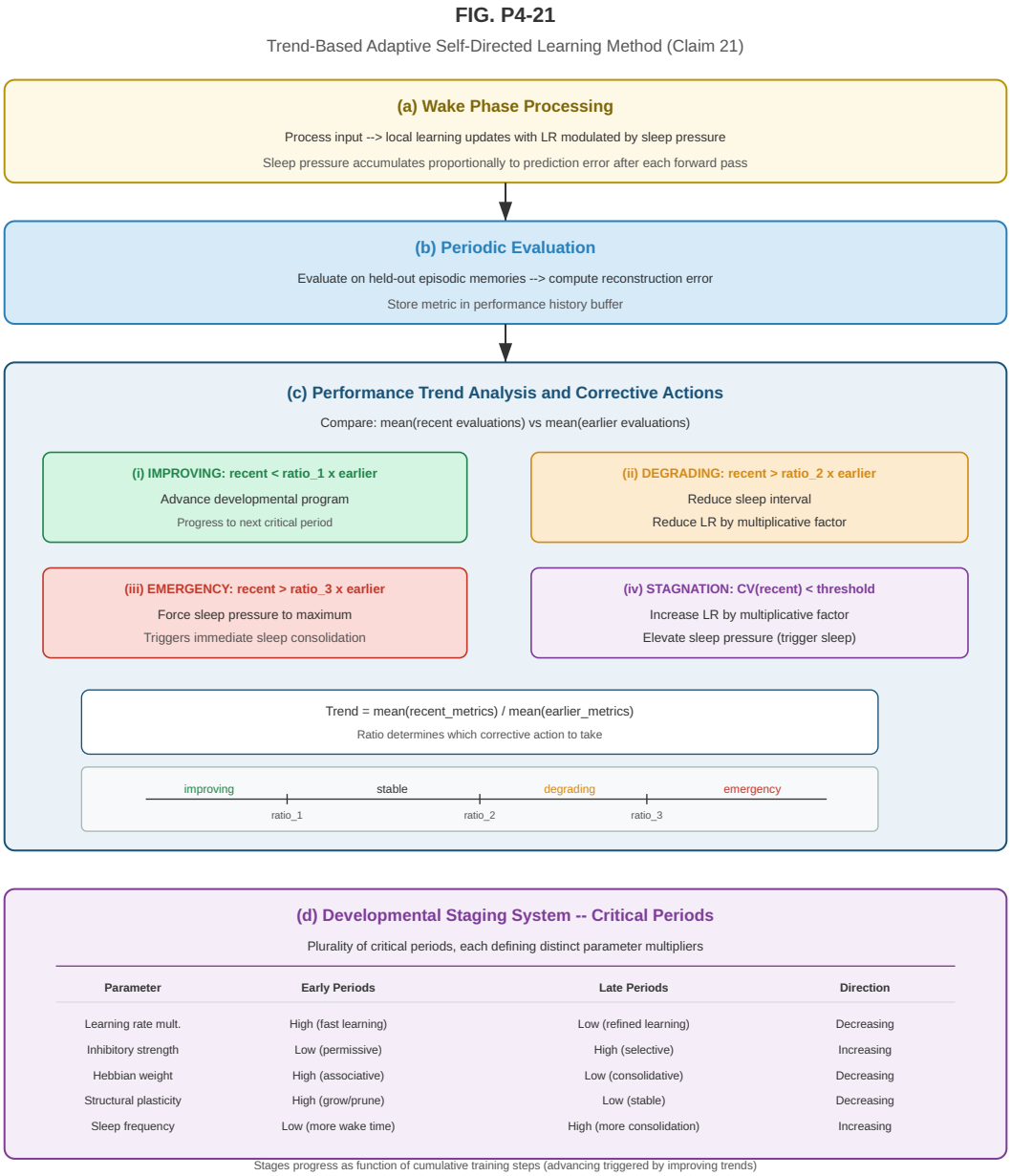

Figure 24: FIG. P4-21 — Trend-Based Adaptive Self-Directed Learning Method (Claim 21)

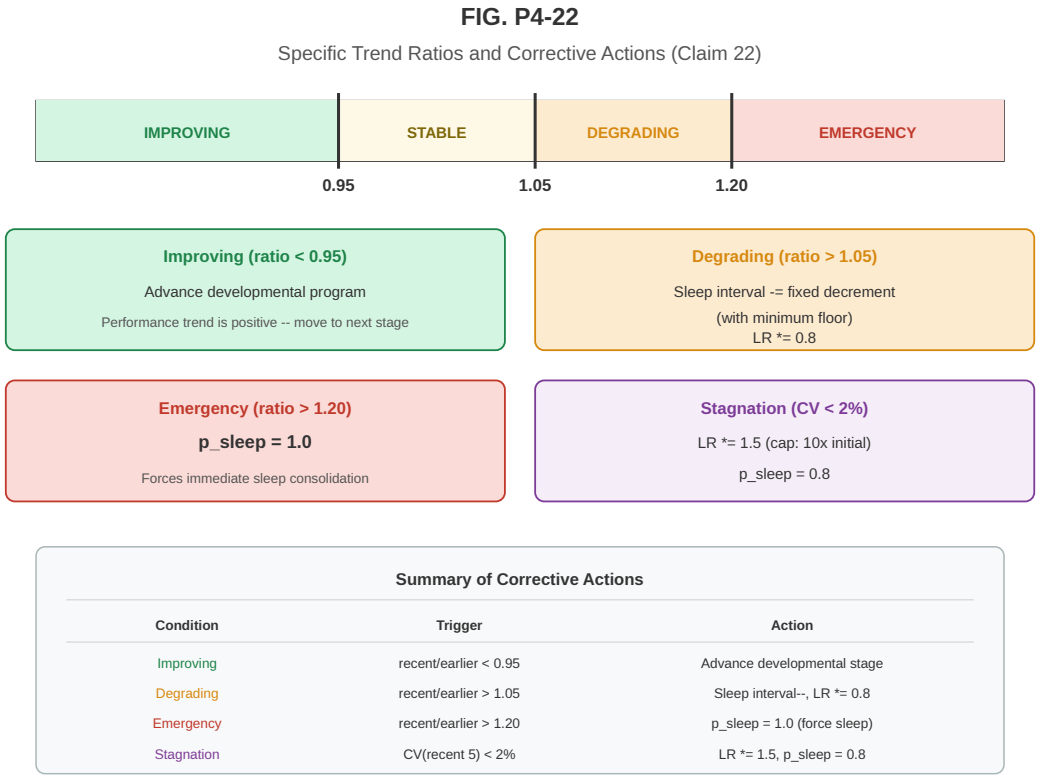

Figure 25: FIG. P4-22 — Trend Ratios and Corrective Actions (Claim 22)

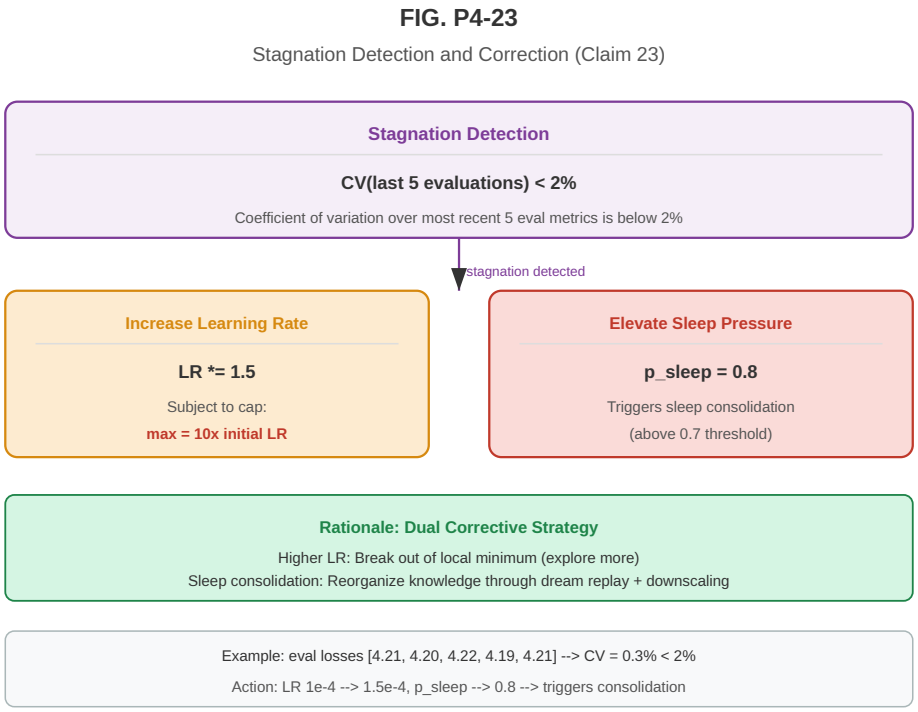

Figure 26: FIG. P4-23 — Stagnation Detection and Correction (Claim 23)

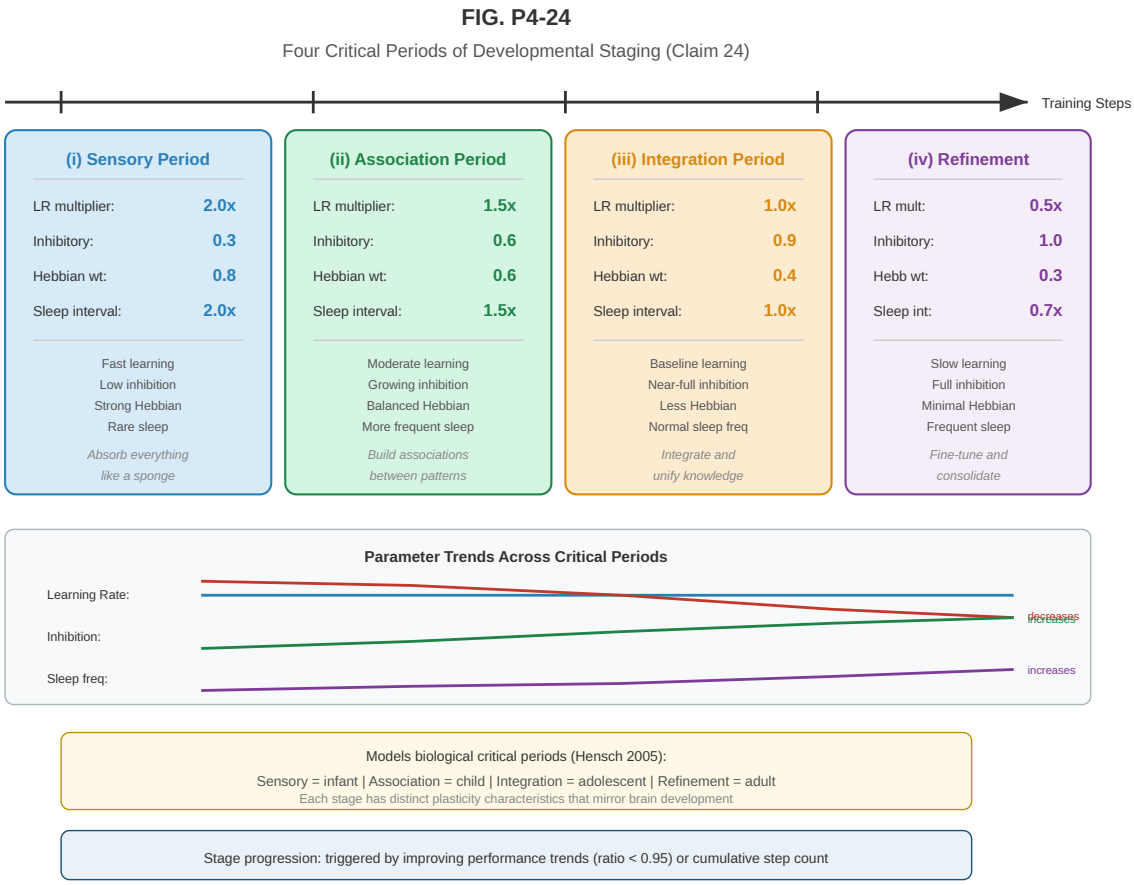

Figure 27: FIG. P4-24 — Four Critical Periods of Developmental Staging (Claim 24)

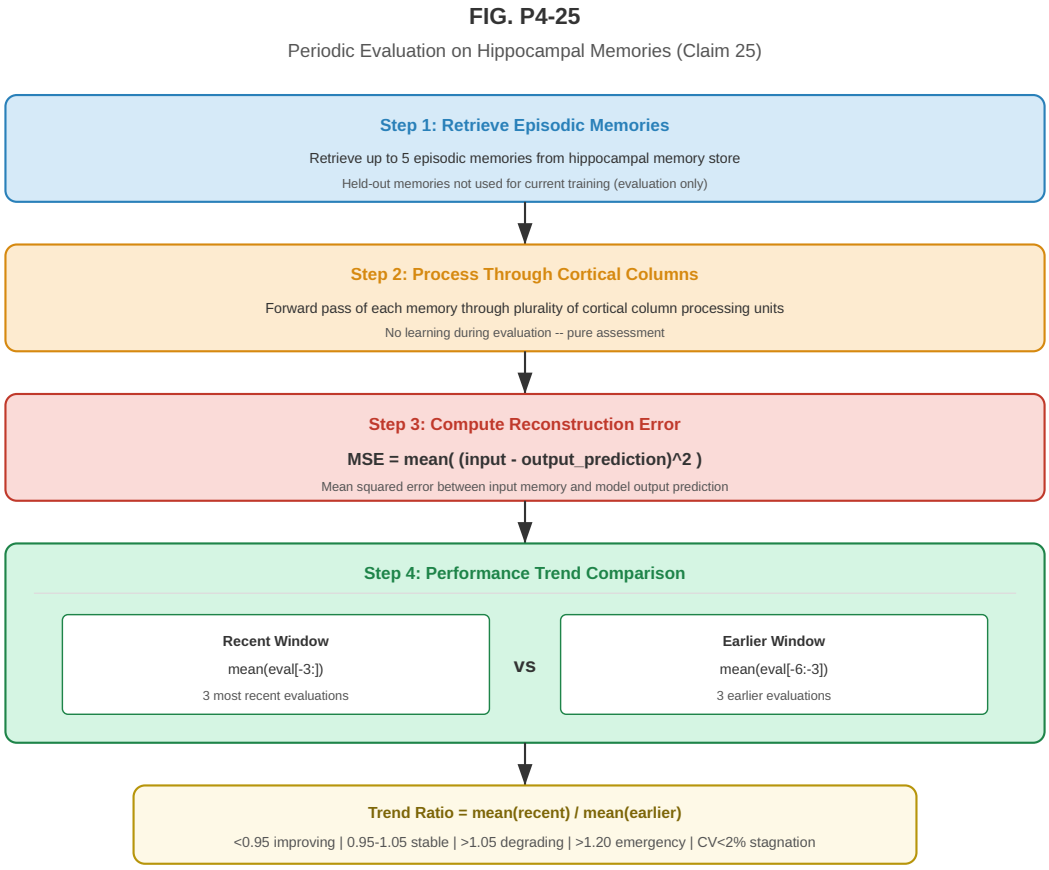

Figure 28: FIG. P4-25 — Periodic Evaluation on Hippocampal Memories (Claim 25)