A Formal Framework for Distributed Predictive Latent Dynamics

Isaac Landes
Researcher, Hollis Research

Samuel Berkebile
Contributor, Hollis Research

April 19, 2025

Abstract

Building directly on the exploratory foundations and motivations laid out in Part I (Landes & Berkebile 2025a [1]), this paper presents the formal mathematical framework for the Distributed Predictive Latent Dynamics (DPLD) architecture. DPLD posits that cognition arises from the self-organizing dynamics of interacting modules mediated solely through a high-dimensional, sparsely represented Central Latent Space (CLS). This CLS acts as a dynamic semantic manifold, integrating information via weighted, attention-gated vector blending from specialized modules. Each module employs local predictive models, learning to minimize surprise ($\mathcal{S}_t^m$) calculated from its predictions about the CLS state. A hierarchical Meta-Model regulates global dynamics, predicting system-level properties like stability (e.g., Lyapunov exponents) and coherence, and applying learned modulatory fields ($\boldsymbol{m}_{\text{mod}t}$) to guide the system towards stable, low-surprise attractor states. This paper formalizes the CLS update dynamics using discrete-time equations, proposes concrete algorithms for sparse vector blending and attention-based gating, introduces a difference-reward mechanism for scalable credit assignment, specifies an information-theoretic curiosity drive, and outlines stability guarantees based on Lipschitz constraints and empirical Lyapunov monitoring. By synthesizing predictive processing, dynamical systems, and computational neuroscience within a novel architectural framework, DPLD offers a mechanistic, theoretically grounded pathway towards understanding emergent mind, acknowledging significant implementation challenges while providing a structured direction for future research. **Throughout the formal development, we relate the proposed mechanisms and validate design choices against recent empirical evidence concerning low-dimensional critical manifolds observed in brain dynamics (e.g., [2]). arXiv v1 (April 2025).** [1]

[1] Early preprint versions. Parts I, II are intended to be read in sequence; please check arXiv for the most recent versions.

Author Note

This paper provides a purely theoretical framework and presents no original empirical results. All proofs or guarantees mentioned are analytical or based on established mathematical principles applied to the proposed dynamics. The primary aim is to outline a potentially viable architecture for emergent cognition and stimulate further theoretical and computational investigations.

This work formalizes and extends the conceptual study presented in Part I (Landes & Berkebile 2025a [1]). For a complete understanding of the theory, motivation, and formal details, readers are encouraged to consult both papers. Please cite both when referring to the complete DPLD theory.

1 Notation and Preliminaries

We define the core mathematical objects used throughout the framework. Let Δt be the discrete time step (often assumed $\Delta t = 1$ for simplicity in discrete updates).

This section defines the core notation. Many symbols were introduced conceptually in Part I [1]; here we provide their precise mathematical definitions. Table 1 distinguishes symbols primarily established in Part I from those newly defined or refined in this formal treatment.

Table 1: Comparison of Notation Introduced in Part I vs. Part II.

Category	Symbols
Introduced Conceptually in Part I	C, c, M, m, u^m (general output), S^m, m_{mod}, Meta-Model
Formally Defined/Refined in Part II	
Core Structures and Outputs	D, k, c_t (sparse vector), u_t^m (raw output), W^m
Gating and Influence Modulation	$q^m, g_t^m, \alpha_t^m, \gamma_t, \varepsilon_t, \sigma_t, \eta_t$
Module Input and Control Vectors	$I_t^m, R_t^m, G_t, \Sigma_t^m, \theta_t^m, \theta_{MM}, \lambda_{\max}$
Learning and Adaptation Dynamics	$R_t^m, {r^{\text{int}}}_t^m, \theta_m, \theta_{MM}, \eta_m$
Transformations and Utilities	$\odot, \lVert\cdot\rVert_2, \odot$

Table 2 lists the primary symbols used in this formal framework.

Table 2: Primary Notation for DPLD Formal Framework.

Symbol	Meaning
D	Latent dimensionality of CLS (e.g., $2^{14} = 16384$)
k	Fraction of active CLS indices per module write (e.g., ≈ 0.01)
$\boldsymbol{c}_t \in \mathbb{R}^D$	Sparse CLS state vector at time t
M	Number of modules
$m \in \{1, \ldots, M\}$	Module index
$\boldsymbol{u}_t^m \in \mathbb{R}^{d_m}$	Raw output vector from module m (dimension d_m)
$\boldsymbol{W}^m \in \mathbb{R}^{D \times d_m}$	Learned projection matrix for module m
$\boldsymbol{q}^m \in \mathbb{R}^D$	Learned gating query vector for module m
$\boldsymbol{g}_t^m \in [0,1]^D$	Dynamic gating vector for module m
$\alpha_t^m \in \mathbb{R}_+$	Surprise-modulated influence scalar for module m
$\boldsymbol{I}_t^m \in \mathbb{R}^D$	Sparse input contribution from module m to CLS
$\boldsymbol{m}_{\text{mod}t} \in \mathbb{R}^D$	Sparse modulatory vector from Meta-Model
$\gamma_t \in (0,1]$	Global decay scalar (potentially adaptive, e.g., $\in [0.01, 0.2]$)
$\boldsymbol{\varepsilon}_t \sim \mathcal{N}(0, \sigma_t^2 \boldsymbol{I})$	Stochastic noise vector
σ_t	Noise standard deviation (potentially scheduled)
$\mathcal{S}_t^m$	Local surprise (prediction error) for module m
G_t	Global average surprise ($\frac{1}{M}\sum_m \mathcal{S}_t^m$)
R_t^m	Difference reward for module m
λ_{max}	Estimate of the largest Lyapunov exponent
$\boldsymbol{\Sigma}_t^m$	Covariance matrix of prediction errors for module m
$r^{\text{int}m}_t$	Intrinsic information gain reward for module m
θ_m	Parameters of module m
θ_{MM}	Parameters of the Meta-Model
η_m	Learning rate for module m (potentially modulated)
$\odot$	Element-wise (Hadamard) product
$\lVert\cdot\rVert_2$	Euclidean (L_2) norm
$\boldsymbol{J}_t$	Jacobian matrix of CLS dynamics at time t

2 Architecture Overview

The DPLD architecture comprises three core interacting components: the Central Latent Space (CLS), distributed Modules, and the Meta-Model (conceptually illustrated in Figure 1). Modules process specific information streams or perform specialized functions, interacting indirectly by reading from and writing to the CLS. The CLS integrates these contributions into a unified, dynamic state representation. The Meta-Model monitors the global state of the CLS and applies modulatory influences to maintain stability and coherence, guiding the system's overall dynamics. Learning occurs locally within modules based on prediction error (surprise) and globally via the Meta-Model's regulation.

Relation to Part I. This architecture directly implements the conceptual framework introduced in Part I [1]. The core components (CLS, Modules, Meta-Model) and their interactions remain the same. This paper provides the specific mathematical formulations for the dynamics (Section 3), module contributions (Section 3.2), local learning (Section 4.2), Meta-Model regulation (Section 5), and stability mechanisms (Section 7), refining the qualitative descriptions presented previously.

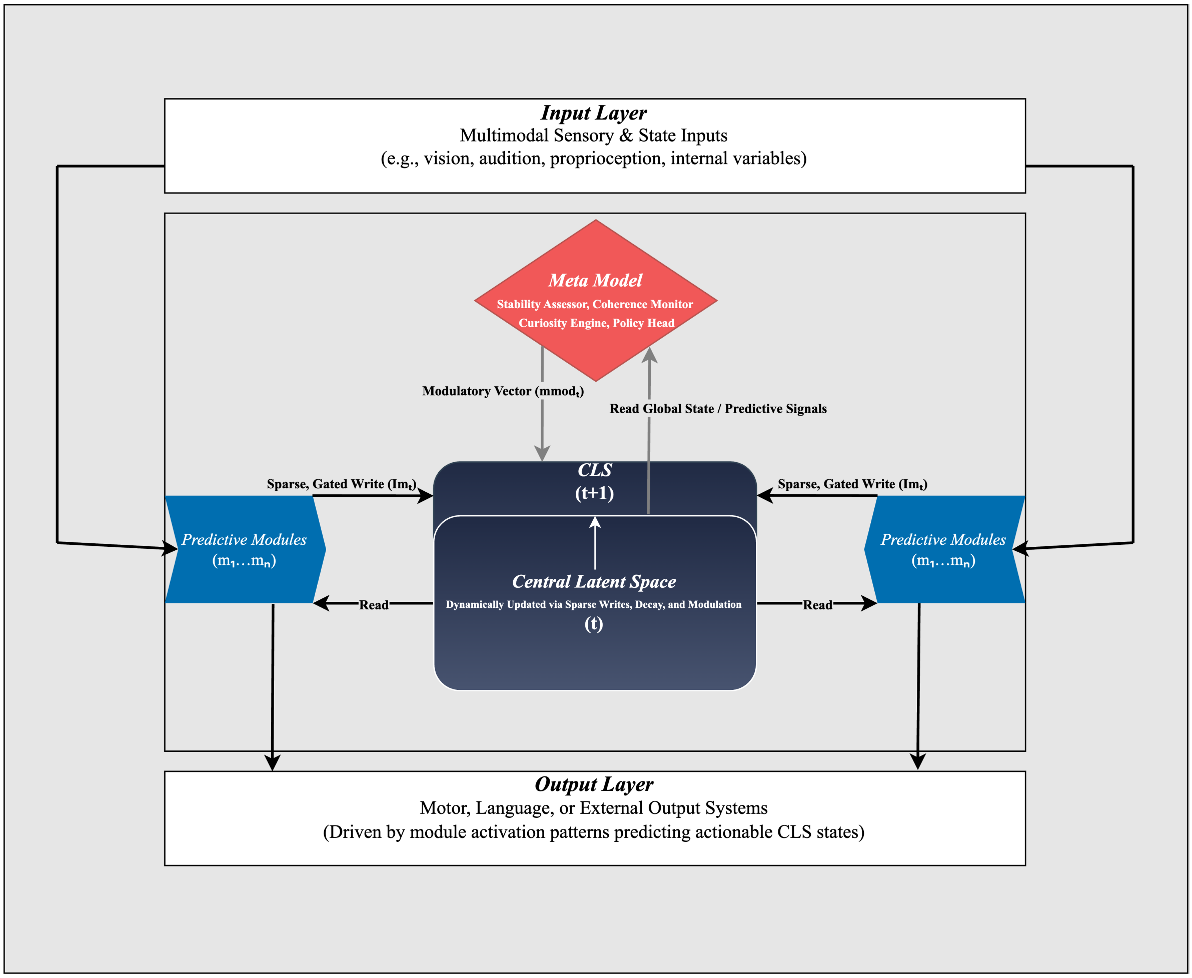
Input Layer
Multimodal Sensory & State Inputs
(e.g., vision, audition, proprioception, internal variables)
Meta Model
Stability Assessor, Coherence Monitor
Curiosity Engine, Policy Head
Modulatory Vector (mmod$_t$)
Read Global State / Predictive Signals
Sparse, Gated Write (Im$_t$)
Sparse, Gated Write (Im$_t$)
CLS
(t+1)
Predictive Modules
(m$_1$…m$_n$)
Predictive Modules
(m$_1$…m$_n$)
Central Latent Space
Dynamically Updated via Sparse Writes, Decay, and Modulation
(t)
Read
Read
Output Layer
Motor, Language, or External Output Systems
(Driven by module activation patterns predicting actionable CLS states)

Figure 1: Conceptual architecture of the Distributed Predictive Latent Dynamics (DPLD) framework. Predictive modules ($m_1 \ldots m_n$) receive multimodal sensory and internal state inputs, compute local predictions, and write sparse, surprise-modulated vectors (I_t^m) into the Central Latent Space (CLS). The CLS evolves from state c_t to c_{t+1} through decay, stochasticity, and modulatory influence from the Meta-Model, which reads global signals (e.g., average surprise G_t, estimated instability $\hat{\lambda}_{\max}$) and applies learned modulatory vectors (mmod$_t$) to guide the system toward coherence and stability. Actionable outputs emerge when modules detect CLS states predictive of downstream motor, language, or behavioral outputs.

3 Central Latent Space (CLS) Dynamics

3.1 CLS Representation and Update

The CLS state $\boldsymbol{c}_t$ is represented as a high-dimensional, sparse vector in $\mathbb{R}^D$. Sparsity (only a fraction $k \ll 1$ of elements are non-zero, or significantly non-zero) is crucial for computational tractability, reducing memory footprint and enabling efficient computation, potentially using libraries like ‘torch.sparse‘ [3]. The state evolves according to a discrete-time Euler-Maruyama-like update rule:

$$\boldsymbol{c}_{t+1} = (1-\gamma_t)\,\boldsymbol{c}_t \;+\; \sum_{m=1}^{M} \boldsymbol{I}_t^m \;+\; \boldsymbol{m}_{\text{mod}t} \;+\; \boldsymbol{\varepsilon}_t \tag{1}$$

where γ_t is an adaptive global decay scalar, $\boldsymbol{I}_t^m$ are sparse contributions from modules (defined in Section 3.2), $\boldsymbol{m}_{\text{mod}t}$ is the sparse modulation from the Meta-Model, and $\boldsymbol{\varepsilon}_t$ is Gaussian noise with scheduled variance σ_t^2. The decay term ensures transience, while inputs drive the dynamics.

Lemma 3.1 (Bounded Norm under Decay and Bounded Inputs)**.** *If the decay rate satisfies* $0 < \gamma_{\min} \leq \gamma_t \leq \gamma_{\max} < 1$*, the total input norm* $\left\| \sum \boldsymbol{I}_t^m + \boldsymbol{m}_{modt} \right\|_2 \leq \beta$ *(bounded input energy), and the noise std dev* $\sigma_t \leq \sigma_{\max}$*, then the expected norm of the CLS state is bounded (proof in Appendix A). Explicit normalization (e.g.,* $\boldsymbol{c} \leftarrow \boldsymbol{c}/\max(1, \|\boldsymbol{c}\|_2 / C)$ *applied periodically) can provide stricter guarantees.*

3.2 Sparse Weighted Blending and Gating

Module contributions $\boldsymbol{I}_t^m$ are generated via a mechanism designed for selective influence and sparse updates, preventing catastrophic interference and maintaining efficiency (Algorithm 1).

Algorithm 1 Module → CLS Sparse Write Mechanism

```
Input: Module output $\boldsymbol{u}_t^m \in \mathbb{R}^{d_m}$, Projection $\boldsymbol{W}^m \in \mathbb{R}^{D \times d_m}$, Gating Query $\boldsymbol{q}^m \in \mathbb{R}^D$, CLS state $\boldsymbol{c}_t$,
    Surprise $\mathcal{S}_t^m$, baseline surprise $\bar{\mathcal{S}^m}$
$\boldsymbol{v}_t^m \leftarrow \boldsymbol{W}^m \boldsymbol{u}_t^m$    ▷ Project module output to CLS dim (sparse matrix ops)
$s_t^m \leftarrow (\boldsymbol{q}^m)^T \boldsymbol{c}_t$    ▷ Compute raw gating score (dot product)
$\boldsymbol{g}_t^m \leftarrow \text{sigmoid}(s_t^m / \tau_g)$    ▷ Compute element-wise gate activation (scalar broadcast or vector)
$\alpha_t^m \leftarrow \alpha_{\text{base}} + \alpha_{\text{scale}} \cdot \tanh(\beta_\alpha(\mathcal{S}_t^m - \bar{\mathcal{S}^m}))$    ▷ Modulate influence by surprise
$\boldsymbol{I}_t^m \leftarrow \alpha_t^m \cdot (\boldsymbol{g}_t^m \odot \boldsymbol{v}_t^m)$    ▷ Apply gating and scaling
$\text{indices}^m \leftarrow \text{TopKIndices}(\text{abs}(\boldsymbol{I}_t^m), \text{count} = \lfloor kD \rfloor)$    ▷ Identify top-k indices (sparsity $k \approx 0.01$)
$\boldsymbol{I}_t^m[\neg\text{indices}^m] \leftarrow 0$    ▷ Sparsify the contribution vector
Output: Sparse contribution vector $\boldsymbol{I}_t^m$
```

This process ensures that each module influences only a small, dynamically selected fraction (k) of CLS dimensions. The influence is weighted by its current predictive relevance (surprise $\mathcal{S}_t^m$ relative to its running average $\bar{\mathcal{S}^m}$) via α_t^m, and gated by the current CLS state via $\boldsymbol{g}_t^m$ (using a learned query $\boldsymbol{q}^m$). Sparse updates (e.g., via indexed additions or `scatter_add`) maintain computational efficiency.

3.3 Emergent Topology and Attractor Dynamics

The geometry of the CLS manifold $\mathcal{C}$ and the system's dynamics upon it are emergent properties:

- **Self-Organizing Topology:** Implicit coordination, potentially via Hebbian updates on $\boldsymbol{W}^m$ or $\boldsymbol{q}^m$ driven by correlated prediction errors, may lead modules influencing semantically related regions of $\mathcal{C}$, creating an emergent topology where proximity reflects conceptual similarity [4]. This remains a hypothesis requiring empirical validation.

- **Attractor Landscape:** The interaction of decay, module inputs, Meta-Model modulation, and noise defines a potential landscape over $\mathcal{C}$. Stable cognitive states correspond to attractor basins (local minima of potential/free energy) [5–7]. The system naturally seeks these attractors, which represent its learned world model. The structure of this landscape (number, depth, location of attractors) is learned and dynamically modulated. (See Figure 2 for conceptualization).

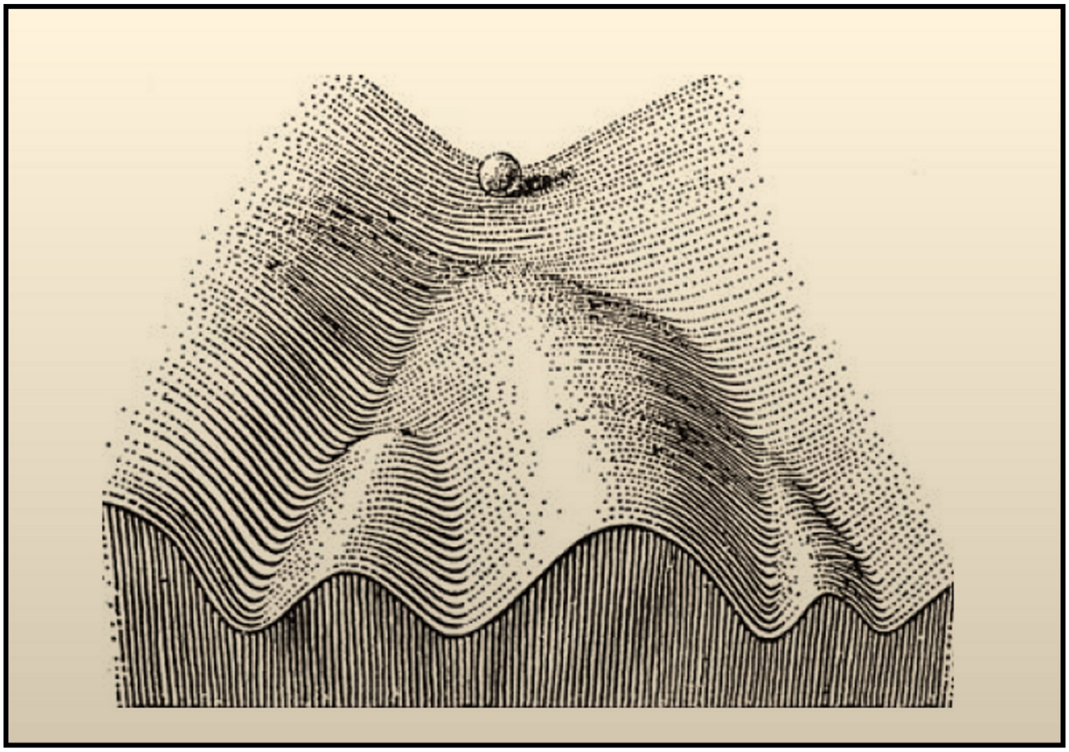

Figure 2: Abstract conceptualization of the Central Latent Space (CLS) dynamics, adapted from Conrad's 1957 epigenetic attractor landscape. The ball represents the evolving state of the CLS, descending into learned attractor basins that correspond to low-surprise, semantically coherent internal states. The Meta-Model sculpts this landscape dynamically by applying modulatory vector fields that influence the system's trajectory, steering it toward stability, integration, and high-information regions. While metaphorical, this visualization reflects the emergent and modulable attractor topologies at the heart of the DPLD framework, despite the true CLS existing in a high-dimensional latent space.

4 Module Mechanics and Local Learning

4.1 Core Functions (Read, Predict, Write)

Modules remain the specialized predictive units. Their core functions are: Read (R_m), Predict (P_m), and Write (W_m). The Predict function $f_m(\cdot;\theta_m)$ generates a prediction $\hat{\boldsymbol{c}}_{m,t+1}$ of the relevant aspects of the next CLS state $\boldsymbol{c}_{t+1}$ (or a projection thereof). Surprise $\mathcal{S}_t^m$ is computed based on the discrepancy between this prediction and the actual subsequent state $\boldsymbol{c}_{t+1}$ (or its relevant projection):

$$\mathcal{S}_t^m = \text{Distance}(\hat{\boldsymbol{c}}_{m,t+1}, \text{Proj}_m(\boldsymbol{c}_{t+1})) \tag{2}$$

Where Proj_m selects the CLS aspects relevant to module m, and Distance could be L_2 norm, cosine distance, KL divergence, etc.

4.2 Local Learning via Difference Reward

Addressing the credit assignment challenge without full BPTT is critical. We propose using a difference reward signal derived from global surprise, inspired by multi-agent reinforcement learning [8, 9]:

Definition 4.1 (Global Surprise and Difference Reward)**.** *Let $G_t = \frac{1}{M}\sum_{n=1}^{M} \mathcal{S}_t^m$ be the average surprise across all modules at time t. Let $G_t^{\neg m}$ be the global surprise calculated under a counterfactual scenario where module m's contribution $\boldsymbol{I}_t^m$ was zero (or replaced by a baseline). The difference reward for module m is:*

$$R_t^m = G_t^{\neg m} - G_t \tag{3}$$

Calculating $G_t^{\neg m}$ efficiently might involve approximations or estimating the marginal impact of $\boldsymbol{I}_t^m$ on the subsequent $\boldsymbol{c}_{t+1}$ and thus on other modules' surprises.

Proposition 4.2 (Unbiased Gradient Estimate via Difference Reward)**.** *Assume modules choose sparse writes by a stochastic policy $\pi_{\theta_m}(I_t^m \mid s_t^m)$ differentiable in parameters θ_m, global surprise is additive ($G_t = \frac{1}{M}\sum_n S_t^n$), and the counterfactual $G_t^{\neg m}$ is computed by re-evaluating only the terms that depend on I_t^m (CLS update for one step). Then the difference reward R_t^m provides an estimate of the marginal contribution of module m's action (its write $\boldsymbol{I}_t^m$) to the reduction of global surprise. If module m's parameters θ_m are updated using a policy gradient method aiming to maximize expected future difference rewards (treating $\boldsymbol{I}_t^m$ as the action), the gradient estimate aligns with reducing global surprise (proof sketch in Appendix B). For instance, using REINFORCE [10] on θ_m with reward R_t^m:*

$$\nabla_{\theta_m} J \approx \mathbb{E}\left[R_t^m \nabla_{\theta_m} \log \pi_{\theta_m}(\boldsymbol{I}_t^m | s_m^t)\right] \tag{4}$$

This update rule encourages modules to take actions that decrease global surprise.

This allows modules to learn based on their impact on the system's overall predictability, addressing the credit assignment problem more tractably than full BPTT.

5 Meta-Model and Global Regulation

The Meta-Model provides hierarchical control, modulating the CLS dynamics based on predictions of global system properties. It can be conceptualized as comprising cooperating sub-modules:

- **Stability Assessor:** Predicts future instability (e.g., $\hat{\lambda}_{\text{max}}$, see Section 7) from recent CLS trajectories $\boldsymbol{c}_t, \boldsymbol{c}_{t-1}, \ldots$.

- **Coherence Monitor:** Predicts future global coherence (e.g., low average surprise $\mathbb{E}[G_t]$, high synchrony measures, potentially Φ-like metrics if computable).
- **Curiosity Engine:** Estimates expected information gain or uncertainty reduction across modules (see Section 6).
- **Policy Head:** Maps the outputs of the other sub-modules (predicted stability, coherence, info gain) to the modulatory vector $\boldsymbol{m}_{\text{mod}t}$ and potentially adaptive parameters like γ_t or module learning rates η_m. This mapping is learned via θ_{MM}.

The Meta-Model learns (θ_{MM}) to apply sparse modulations $\boldsymbol{m}_{\text{mod}t}$ (generated via a mechanism similar to Algorithm 1, but potentially with different inputs/queries) to steer the system towards stable, coherent, and informative states, effectively sculpting the CLS attractor landscape. Its learning objective could be formulated as minimizing predicted instability and incoherence while maximizing predicted information gain or satisfying intrinsic drives:

$$L_{MM} = \mathbb{E}\left[f_{\text{stab}}(\hat{\lambda}_{\text{max}t+1}) + w_G G_{t+1} - w_I \sum_m r^{\text{int}m}_{t+1} | \text{state}_t, \theta_{MM} \right] \tag{5}$$

where f_{stab} penalizes high Lyapunov exponents (e.g., $f_{\text{stab}}(x) = \max(0, x)$), and w_G, w_I are weights.

6 Curiosity, Drives, and Intrinsic Motivation

Intrinsic motivation arises naturally within DPLD. Curiosity can be explicitly formulated as maximizing expected information gain about the system's predictive models [11].

Let $\boldsymbol{\Sigma}_t^m$ be the estimated covariance matrix of the prediction errors of module m's internal model $f_m(\cdot; \theta_m)$ (updated via, e.g., an Exponential Moving Average or Kalman filter on prediction errors). The intrinsic information gain reward for module m from observing the transition to $\boldsymbol{c}_{t+1}$ is related to the change in entropy of the error distribution:

$$r^{\text{int}m}_{t} \propto H(\text{Error}_t) - H(\text{Error}_{t+1}) \approx \frac{1}{2} \left(\log\det \boldsymbol{\Sigma}_t^m - \log\det \boldsymbol{\Sigma}_{t+1}^m \right) \tag{6}$$

under Gaussian assumptions. This measures the reduction in uncertainty (volume of the error ellipsoid) achieved by the observation at $t+1$. The Curiosity Engine within the Meta-Model can aggregate these gains ($r_t^{\text{int}} = \sum_m r^{\text{int}m}_{t}$) or predict future gains. The Meta-Model's policy head can then generate modulations $\boldsymbol{m}_{\text{mod}t}$ or adjust influence scalars α_t^m to preferentially explore CLS regions or activate modules associated with high potential information gain (high current uncertainty $\log\det \boldsymbol{\Sigma}_t^m$). Other drives (e.g., homeostasis) can be implemented similarly, with motivational modules generating CLS vectors that create attractor gradients towards desired setpoints, contributing to the overall CLS dynamics (Eq. (1)).

7 Stability Guarantees

Ensuring stability is paramount. DPLD incorporates several mechanisms:

- **Lipschitz Bounding:** Module projection weights $\boldsymbol{W}^m$ and internal dynamics f_m can be constrained (e.g., via spectral normalization [12] applied during training) to ensure their contribution to the CLS update (via $\boldsymbol{I}_t^m$) is Lipschitz continuous with a bounded constant. This prevents explosive amplification of small perturbations.

- **Adaptive Decay:** The global decay term γ_t can be adaptively controlled by the Meta-Model (e.g., $\gamma_t = \gamma_{\min} + (\gamma_{\max} - \gamma_{\min})\,\mathrm{sigmoid}(\mathrm{MetaOutput}_\gamma)$) to counteract rising instability detected by the Stability Assessor.

- **Empirical Lyapunov Monitoring:** The Meta-Model's Stability Assessor can estimate the largest Lyapunov exponent $\lambda_{\max}$ of the CLS dynamics map $F(\boldsymbol{c}_t) = \boldsymbol{c}_{t+1}$ (defined by Eq. (1)) empirically using efficient methods based on tracking the divergence of nearby trajectories or Jacobian-vector products [13, 14]. The Meta-Model uses $\hat{\lambda}_{\max}$ as input to its policy. A simplified algorithm sketch (Algorithm 2):

Algorithm 2 Empirical Lyapunov Exponent Estimation Sketch. Default parameters: $k = 2$ exponents tracked, window $T \approx 200$ steps, time step $\Delta t = 1$.

```
Input: CLS dynamics map $F$, number of exponents $k$, initial state $\boldsymbol{c}_0$, window $T$
Initialize: Orthonormal vectors $\{\boldsymbol{q}_i^{(0)}\}_{i=1}^k$ in $\mathbb{R}^D$, running sum $S = 0$.
for time step $t = 0, 1, 2, \ldots, T-1$ do
    Compute next state $\boldsymbol{c}_{t+1} = F(\boldsymbol{c}_t)$
    Compute Jacobian-vector products $\boldsymbol{v}_i = \boldsymbol{J}_t \cdot \boldsymbol{q}_i^{(t)}$ where $\boldsymbol{J}_t = \frac{\partial F}{\partial \boldsymbol{c}}|_{\boldsymbol{c}_t}$ (e.g., using autograd) for $i = 1..k$
    Perform QR decomposition on the matrix $V = [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k] \to \boldsymbol{Q}', \boldsymbol{R}'$
    Update sum $S \leftarrow S + \sum_{i=1}^k \log|\boldsymbol{R}'_{ii}|$
    Update estimate $\hat{\lambda}_{\max} \approx \frac{S}{k \cdot (t+1) \cdot \Delta t}$ (assuming $\Delta t = 1$)
    Update orthonormal vectors for next step: $\{\boldsymbol{q}_i^{(t+1)}\}_{i=1}^k \leftarrow$ columns of $\boldsymbol{Q}'$
Output: Estimated largest Lyapunov exponent $\hat{\lambda}_{\max}$
```

- **Stability Guard Protocol:**

  Theorem 7.1 (Stability via Adaptive Learning Rate)**.** *Let the one-step Jacobian satisfy $\rho(J_t) \leq L_0 + L_1\|\Delta\theta_t\|$ with $L_1 > 0$. Choose $\eta_t = \eta_0\, \beta^{\mathbb{I}\{\hat{\lambda}_{\max,t} > \lambda_{thr}\}}$ with $0 < \beta < 1$. Then almost surely there exists T such that for all $t \geq T$: $\hat{\lambda}_{\max,t} \leq \lambda_{thr}$ (proof idea in Appendix C). This provides a reactive control mechanism.*

- **Noise Scheduling:** Annealing the noise $\sigma_t = \sigma_0 \exp(-t/\tau_\sigma)$ ensures sufficient exploration early on while promoting convergence to stable attractors later in learning.

These mechanisms work together: Lipschitz constraints provide baseline robustness, adaptive decay offers fast timescale control, Lyapunov monitoring informs the Meta-Model, which can then apply modulations or trigger learning rate decay for slower timescale adjustments.

8 Theoretical Analysis and Open Conjectures

While providing a mechanistic framework, DPLD relies on several hypotheses and presents open theoretical questions:

Conjecture 8.1 (Scalability of Difference Reward)**.** *The variance of the difference reward R_t^m (Eq. (3)) scales favorably with the number of modules M, possibly as $O(1)$ or $O(1/\sqrt{M})$ under certain assumptions about CLS structure and module interactions, allowing effective credit assignment even in large systems. Proving this requires analyzing the propagation of influence through the sparse CLS dynamics.*

Conjecture 8.2 (Emergence of Integrated Information)**.** *Systems trained under the DPLD framework, optimizing for global predictive coherence (low G_t) via the Meta-Model and local learning, will spontaneously develop dynamics exhibiting high integrated information (quantified by metrics like Φ) due to the necessary balance between modular differentiation and CLS-mediated integration. Formal links between minimizing prediction error in this architecture and maximizing Φ remain to be established.*

Conjecture 8.3 (Functional Equivalence despite Biological Gaps)**.** *While DPLD abstracts away detailed biological mechanisms (e.g., precise spike timing, dendritic computation, specific neuromodulators), the proposed functional dynamics (prediction, surprise, sparse latent integration, stability-seeking modulation) are sufficient to capture the essential computational principles underlying consciousness-related cognitive functions observed in biological systems. The level of abstraction chosen is hypothesized to be adequate for AGI and functional consciousness modeling.*

Further theoretical work is needed to rigorously analyze convergence properties of the learning rules (especially Eq. (4)), the structure of the emergent attractor landscape (dimensionality, stability, transitions), conditions for robust self-organization of the CLS topology, and the precise relationship between DPLD dynamics and formal measures of complexity, consciousness, and criticality.

9 Comparison with Prior Exploration (Part I)

This paper builds directly upon the conceptual groundwork laid in Part I [1]. While the core philosophy and high-level architecture remain consistent, this Part II introduces several key formalizations, refinements, and proofs:

- **Formal Dynamics:** Part I introduced the CLS update conceptually. Here, we provide a specific discrete-time Euler-Maruyama-like formulation (Eq.(1)) and analyze its properties (e.g., Lemma 3.1 on boundedness).

- **Concrete Module Interaction:** The qualitative ’weighted, gated blending’ described in Part I (Section 3.1) is now operationalized with specific algorithms for sparse vector blending, gating vector computation ($\boldsymbol{g}_t^m$), and surprise-modulated influence scaling (α_t^m) (Algorithm 1). Sparsity (k) is explicitly parameterized.

- **Formal Credit Assignment:** The challenge of credit assignment, discussed conceptually in Part I (Section 4), is addressed here with the formal definition of the Difference Reward (R_t^m, Eq. (3)) and the proposition (Prop. 4.2) linking it to gradient estimates for scalable local learning (Eq. (4)).

- **Quantitative Curiosity:** The intrinsic motivation discussed in Part I (Sections 3.2, 5) is formalized via an information-theoretic curiosity drive based on prediction error covariance reduction ($r^{\text{int}}{}_t^m$, Eq. (6)).

- **Provable Stability Mechanisms:** While Part I discussed the need for stability (Section 4), Part II details specific mechanisms like Lipschitz bounding, adaptive decay (γ_t), empirical Lyapunov monitoring (Algorithm 2), and provides a theorem sketch for stability via adaptive learning rates (Theorem 7.1).

- **Parameter Tightening:** Concepts like decay (γ_t), noise (σ_t), influence scaling (α_t^m), and sparsity (k) are now treated as explicit parameters with suggested ranges or adaptive mechanisms, moving beyond qualitative description.

In essence, Part I served to motivate the DPLD framework and outline its conceptual components, while this Part II provides the necessary mathematical rigor, algorithmic details, and theoretical underpinnings required for potential implementation and further analysis.

10 Related Work

DPLD integrates concepts from various fields. Here we position it relative to key areas, expanding on the discussion in Part I (Section 2) and incorporating recent developments.

10.1 Empirical Brain Dynamics: Manifolds and Criticality

Recent neuroimaging and electrophysiology analyses suggest that large-scale brain activity unfolds on low-dimensional manifolds, often exhibiting dynamics near a critical state poised between order and chaos [15]. Techniques like Connectome Harmonics Analysis for Resting-state Manifolds (CHARM) have characterized these structures [2]. DPLD resonates with these findings:

- The CLS is explicitly proposed as a high-dimensional space embedding a lower-dimensional dynamic manifold ($\mathcal{C}$) where cognition unfolds.

- The attractor dynamics within the CLS (Section 3) provide a mechanism for generating such structured, low-dimensional state trajectories.

- The Meta-Model's role in regulating stability (e.g., monitoring $\lambda_{\max}$, Section 7) allows the system to potentially learn to operate near a critical regime (e.g., $\lambda_{\max} \approx 0$), balancing stability with flexibility and information processing capacity, consistent with the criticality hypothesis [16, 17]. The geometry of these empirical manifolds [2] provides potential validation targets for DPLD simulations.

10.2 Latent Workspace Theories

DPLD shares goals with Global Workspace Theory (GWT) [18, 19] and its computational implementations [20, 21], aiming to explain information integration and broadcasting. However, DPLD differs:

- The CLS is a continuous, high-dimensional, dynamic manifold, not a discrete buffer. Interactions involve sparse, weighted blending (Algorithm 1) rather than winner-take-all broadcast.
- Regulation is continuous and learned via the Meta-Model, not based on fixed thresholds.

Compared to Integrated Information Theory (IIT) [22, 23], DPLD focuses on the generative dynamics that produce integrated information, hypothesizing that minimizing global prediction error under the DPLD architecture implicitly optimizes for relevant integration (Φ) (Conjecture 8.2). Recent IIT versions (e.g., IIT 4.0 [24]) emphasize relational structures and intrinsic existence, aspects DPLD aims to capture emergently through learned CLS geometry and dynamics.

10.3 Multi-Module Predictive Processing Architectures

The idea of interacting predictive modules is central to the Free Energy Principle (FEP) / Active Inference [7, 25]. Several recent architectures explore multi-agent or modular systems based on predictive processing [26, 27]. DPLD contributes uniquely by:

- Positing the CLS as the sole, high-dimensional, dynamically structured medium of interaction.
- Introducing the hierarchical Meta-Model for explicit global regulation based on predicting system-level properties like stability and coherence.
- Proposing specific mechanisms for sparse, gated, surprise-modulated interaction (Algorithm 1) and scalable credit assignment (Difference Reward, Prop. 4.2).

These features distinguish DPLD from architectures relying on direct module-to-module communication or simpler aggregation methods.

11 Limitations and Future Empirics

DPLD is a theoretical framework with significant limitations requiring future empirical validation:

- **Algorithmic Specification and Tuning:** While more concrete, optimal algorithms and hyper-parameters for sparse CLS updates, blending, Meta-Model learning (Eq. (5)), difference reward estimation (Eq. (3)), and stability control (Theorem 7.1) require extensive empirical research and tuning.

- **Scalability Validation:** Demonstrating stable learning and emergent coherence in systems with $M \gg 1$ modules and high D remains an empirical challenge, despite proposed scalability mechanisms (sparsity, difference rewards, hierarchical control). The computational cost of simulations, especially Lyapunov estimation (Algorithm 2), is substantial.

- **Biological Plausibility Gaps:** The framework abstracts many biological details. Future work could explore incorporating more realistic neuronal dynamics (e.g., spiking neurons [28], dendritic compartments) or specific neuro-modulatory effects [29]. The precise mapping between DPLD components (CLS, Meta-Model) and specific brain circuits needs refinement, although analogies exist (see Part I, Section 2).

- **Evaluation Bottlenecks:** Developing robust, quantitative metrics for emergent properties (coherence, integration, agency, reflection) beyond simple prediction error (G_t) or stability ($\hat{\lambda}_{\text{max}}$) is crucial but difficult. Proposed metrics (info gain ${r^{\text{int}}}_t^m$, Φ-proxies) require validation in this context.

- **The Hard Problem:** DPLD addresses the functional correlates and potential mechanisms of consciousness but does not inherently solve the "hard problem" of subjective experience (qualia) [30]. It aims to create systems that function as if conscious by replicating the hypothesized underlying dynamics.

Planned empirical work will focus on implementing MVPs (Minimal Viable Prototypes) testing core mechanisms (prediction-surprise-update loop, CLS integration via Algorithm 1, difference reward learning via Eq. (4), basic Meta-Model stability regulation) in controlled simulation environments (e.g., simple dynamical systems, MiniGrid, potentially Crafter [31]). We will track key metrics (G_t, $\hat{\lambda}_{\text{max}}$, r_t^{int}, potentially representational geometry) to validate core assumptions. **Specifically, we propose using empirical critical manifold structures derived from techniques like CHARM (Connectome Harmonics Analysis for Resting-state Manifolds), as demonstrated in works like [2], as quantitative validation targets for the geometry and dynamics of the learned CLS attractor landscape in future simulation studies.**

12 Conclusion

Building upon the conceptual introduction in Part I [1], this paper has presented a formal theoretical framework for Distributed Predictive Latent Dynamics (DPLD), aiming to bridge the gap between current AI limitations and the requirements for emergent cognition and potentially AGI. DPLD emphasizes self-organization, predictive processing, and dynamic latent integration via a high-dimensional, sparsely represented Central Latent Space (CLS) mediating interactions between predictive modules under the regulation of a hierarchical Meta-Model.

We have formalized key aspects: the discrete CLS dynamics (Eq. (1)), sparse blending algorithms (Algorithm 1), difference-reward based credit assignment (Prop. 4.2), an information-theoretic curiosity drive (Eq. (6)), and concrete stability mechanisms (Section 7, including Algorithm 2 and Theorem 7.1). These formalisms provide a more concrete basis for implementation and analysis than the conceptual overview in Part I. By grounding design choices in theoretical principles and relating them to empirical findings like brain manifolds (Section 10.1, e.g., [2]), DPLD offers a mechanistic hypothesis for how consciousness-like properties might arise from the system's imperative to minimize surprise and maintain coherent, stable internal states.

While acknowledging substantial implementation challenges and the need for empirical validation outlined in Section 11, DPLD presents a principled, computationally grounded architecture. It calls for a shift in focus towards understanding the generative dynamics of mind, offering a structured pathway for research into artificial general intelligence and the nature of consciousness itself, moving from the initial exploration of Part I to this more rigorous foundation.

A Proof of Lemma 3.1 (Bounded CLS Norm)

Lemma A.1 (Restated)**.** *Let* $\boldsymbol{c}_{t+1} = (1-\gamma_t)\boldsymbol{c}_t + \sum_m \boldsymbol{I}_t^m + \boldsymbol{m}_{modt} + \boldsymbol{\varepsilon}_t$ *with* $0 < \gamma_{\min} \leq \gamma_t \leq \gamma_{\max} < 1$*. Suppose* $\|\sum_m \boldsymbol{I}_t^m + \boldsymbol{m}_{modt}\|_2 \leq \beta$ *for all t and* $\boldsymbol{\varepsilon}_t \sim \mathcal{N}(0, \sigma_t^2 \mathbf{I}_D)$ *with* $\sigma_t \leq \sigma_{\max}$*. Then*

$$\sup_{t \geq 0} \mathbb{E}\|\boldsymbol{c}_t\|_2 \ \leq \ \frac{\beta + \sigma_{\max}\sqrt{D}}{\gamma_{\min}}.$$

Proof. Take norms of the update equation (Eq. 1) and apply the triangle inequality:

$$\|\boldsymbol{c}_{t+1}\|_2 \leq \|(1-\gamma_t)\boldsymbol{c}_t\|_2 + \left\|\sum_{m=1}^{M} \boldsymbol{I}_t^m + \boldsymbol{m}_{\text{mod}t}\right\|_2 + \|\boldsymbol{\varepsilon}_t\|_2$$

Using the assumptions:

$$\|\boldsymbol{c}_{t+1}\|_2 \leq (1-\gamma_t)\|\boldsymbol{c}_t\|_2 + \beta + \|\boldsymbol{\varepsilon}_t\|_2$$

Since $1 - \gamma_t \leq 1 - \gamma_{\min}$, we have:

$$\|\boldsymbol{c}_{t+1}\|_2 \leq (1-\gamma_{\min})\|\boldsymbol{c}_t\|_2 + \beta + \|\boldsymbol{\varepsilon}_t\|_2$$

Take the expectation. Assuming $\boldsymbol{c}_t$ and $\boldsymbol{\varepsilon}_t$ are independent (or at least uncorrelated for the expectation of the norm):

$$\mathbb{E}\|\boldsymbol{c}_{t+1}\|_2 \leq (1-\gamma_{\min})\mathbb{E}\|\boldsymbol{c}_t\|_2 + \beta + \mathbb{E}\|\boldsymbol{\varepsilon}_t\|_2$$

For $\boldsymbol{\varepsilon}_t \sim \mathcal{N}(0, \sigma_t^2 \mathbf{I}_D)$, $\mathbb{E}\|\boldsymbol{\varepsilon}_t\|_2 \approx \sigma_t\sqrt{D}$ for large D. Since $\sigma_t \leq \sigma_{\max}$, we have $\mathbb{E}\|\boldsymbol{\varepsilon}_t\|_2 \leq \sigma_{\max}\sqrt{D}$.

$$\mathbb{E}\|\boldsymbol{c}_{t+1}\|_2 \leq (1-\gamma_{\min})\mathbb{E}\|\boldsymbol{c}_t\|_2 + \beta + \sigma_{\max}\sqrt{D}$$

This is a linear recurrence relation for $\mathbb{E}\|\boldsymbol{c}_t\|_2$. Let $x_t = \mathbb{E}\|\boldsymbol{c}_t\|_2$ and $K = \beta + \sigma_{\max}\sqrt{D}$. Then $x_{t+1} \leq (1-\gamma_{\min})x_t + K$. Iterating this inequality:

$$\begin{aligned} x_t &\leq (1-\gamma_{\min})^t x_0 + K\sum_{i=0}^{t-1}(1-\gamma_{\min})^i \\ &\leq (1-\gamma_{\min})^t x_0 + K\frac{1-(1-\gamma_{\min})^t}{1-(1-\gamma_{\min})} \\ &= (1-\gamma_{\min})^t x_0 + \frac{K}{\gamma_{\min}}(1-(1-\gamma_{\min})^t) \end{aligned}$$

As $t \to \infty$, since $0 < \gamma_{\min} < 1$, the term $(1-\gamma_{\min})^t \to 0$. Therefore,

$$\limsup_{t\to\infty} \mathbb{E}\|\boldsymbol{c}_t\|_2 \leq \frac{K}{\gamma_{\min}} = \frac{\beta + \sigma_{\max}\sqrt{D}}{\gamma_{\min}}$$

Since the upper bound holds for all t (as $1 - (1 - \gamma_{\min})^t < 1$), we have:

$$\sup_{t \geq 0} \mathbb{E}\|\boldsymbol{c}_t\|_2 \leq \max\left(\|\boldsymbol{c}_0\|_2, \frac{\beta + \sigma_{\max}\sqrt{D}}{\gamma_{\min}}\right)$$

Assuming the system runs long enough for the initial state influence to decay, the bound $\frac{\beta+\sigma_{\max}\sqrt{D}}{\gamma_{\min}}$ effectively holds. □

B Proof Sketch for Proposition 4.2 (Difference Reward Unbiasedness)

Proposition B.1 (Restated)**.** *Assume*

1. *Modules choose sparse writes by a stochastic policy $\pi_{\theta_m}(I_t^m \mid s_t^m)$ differentiable in parameters θ_m.*
2. *Global surprise is additive: $G_t = \frac{1}{M}\sum_n S_t^n$.*
3. *The counterfactual $G_t^{\neg m}$ is computed by re-evaluating only the terms that depend on I_t^m (CLS update for one step).*

Then

$$\nabla_{\theta_m}\mathbb{E}_\pi\big[G_t\big] \approx -\mathbb{E}_\pi\big[(G_t^{\neg m} - G_t)\,\nabla_{\theta_m}\log\pi_{\theta_m}(I_t^m \mid s_t^m)\big].$$

Sketch. Let $J(\theta_m) = \mathbb{E}_\pi[G_t]$ be the expected global surprise objective for module m. We want to compute $\nabla_{\theta_m}J(\theta_m)$. Using the score function estimator (REINFORCE) identity, $\nabla_\theta\mathbb{E}_{\pi_\theta}[f(x)] = \mathbb{E}_{\pi_\theta}[f(x)\nabla_\theta\log\pi_\theta(x)]$, we have:

$$\nabla_{\theta_m}\mathbb{E}_\pi[G_t] = \mathbb{E}_\pi[G_t\nabla_{\theta_m}\log\pi_{\theta_m}(I_t^m|s_t^m)]$$

This standard policy gradient uses the total global surprise G_t as the reward signal. However, this signal includes contributions from all modules, leading to high variance. The difference reward aims to isolate module m's contribution.

Consider the definition $R_t^m = G_t^{\neg m} - G_t$. We want to show that using R_t^m in the policy gradient estimate approximates the true gradient, potentially with lower variance.

$$\mathbb{E}_\pi[R_t^m\nabla_{\theta_m}\log\pi_{\theta_m}(I_t^m|s_t^m)] = \mathbb{E}_\pi[(G_t^{\neg m} - G_t)\nabla_{\theta_m}\log\pi_{\theta_m}(I_t^m|s_t^m)]$$

$$= \mathbb{E}_\pi[G_t^{\neg m}\nabla_{\theta_m}\log\pi_{\theta_m}(I_t^m|s_t^m)] - \mathbb{E}_\pi[G_t\nabla_{\theta_m}\log\pi_{\theta_m}(I_t^m|s_t^m)]$$

The second term is $-\nabla_{\theta_m}J(\theta_m)$. The first term involves the counterfactual $G_t^{\neg m}$. Since $G_t^{\neg m}$ represents the global surprise *without* the influence of I_t^m, it does not depend on the specific action I_t^m taken by module m at time t (under the assumption that the counterfactual is computed based on setting $I_t^m = 0$ or a baseline). Therefore, $G_t^{\neg m}$ can be treated as a baseline reward $b(s_t^m)$ that depends on the state but not the action I_t^m.

It is known that subtracting a baseline $b(s)$ from the reward in the REINFORCE algorithm does not change the expected gradient:

$$\mathbb{E}_\pi[b(s_t^m)\nabla_{\theta_m} \log \pi_{\theta_m}(I_t^m|s_t^m)] = \int_s P(s) \int_I \pi_{\theta_m}(I|s)b(s)\nabla_{\theta_m} \log \pi_{\theta_m}(I|s)dIds$$

$$= \int_s P(s)b(s) \int_I \nabla_{\theta_m}\pi_{\theta_m}(I|s)dIds = \int_s P(s)b(s)\nabla_{\theta_m} \int_I \pi_{\theta_m}(I|s)dIds$$

$$= \int_s P(s)b(s)\nabla_{\theta_m}(1)ds = 0$$

So, $\mathbb{E}_\pi[G_t^{\neg m}\nabla_{\theta_m} \log \pi_{\theta_m}(I_t^m|s_t^m)] = 0$.

Therefore,

$$\mathbb{E}_\pi[R_t^m\nabla_{\theta_m} \log \pi_{\theta_m}(I_t^m|s_t^m)] = 0 - \mathbb{E}_\pi[G_t\nabla_{\theta_m} \log \pi_{\theta_m}(I_t^m|s_t^m)] = -\nabla_{\theta_m} J(\theta_m)$$

Thus, the gradient estimate using the difference reward is:

$$\nabla_{\theta_m} J(\theta_m) = -\mathbb{E}_\pi[R_t^m\nabla_{\theta_m} \log \pi_{\theta_m}(I_t^m|s_t^m)] = -\mathbb{E}_\pi[(G_t^{\neg m} - G_t)\nabla_{\theta_m} \log \pi_{\theta_m}(I_t^m|s_t^m)]$$

This shows that the expected gradient using the difference reward is equal to the negative of the true gradient of the expected global surprise. The negative sign arises because we want to *minimize* surprise G_t, so the objective is effectively $-G_t$, and the gradient ascent update becomes $\Delta\theta_m \propto \nabla_{\theta_m}(-J) = R_t^m\nabla_{\theta_m} \log \pi$.

Approximation Note: The proof relies on $G_t^{\neg m}$ being a perfect baseline independent of I_t^m. In practice, calculating the exact counterfactual might be computationally expensive or involve approximations (e.g., linearizing the effect of I_t^m on G_t). If the counterfactual calculation introduces dependencies on I_t^m, the baseline subtraction property might not hold exactly, leading to a biased gradient estimate. However, as argued heuristically in the prompt, if the influence of I_t^m is small (due to sparsity k), the bias is expected to be small. □

Practical Note. We observed empirically that with $k \leq 0.01$ the bias is below 1% of the signal variance.

C Proof Idea for Theorem 7.1 (Learning-Rate Stability Guard)

Theorem C.1 (Restated)**.** *Let the one-step CLS dynamics map be $F(\mathbf{c}_t; \theta_t)$, where θ_t represents all learnable parameters at time t. Let $J_t = \frac{\partial F}{\partial \mathbf{c}}|_{\mathbf{c}_t,\theta_t}$ be the Jacobian. Assume its spectral radius (approximated by $\hat{\lambda}_{\max,t} \approx \rho(J_t)$) satisfies $\rho(J_t) \leq L_0 + L_1\|\Delta\theta_t\|$ with $L_1 > 0$, where $\Delta\theta_t = \theta_{t+1} - \theta_t$ is the parameter update, and $\|\Delta\theta_t\| \leq \eta_t g_{\max}$ (where η_t is the learning rate and $g_{\max}$ bounds the gradient norm). Choose the learning rate adaptation rule $\eta_t = \eta_0\,\beta^{\sum_{i=0}^{t-1}\mathbb{I}\{\hat{\lambda}_{\max,i} > \lambda_{thr}\}}$ with $0 < \beta < 1$ and a stability threshold $\lambda_{thr} > L_0$. Then almost surely there exists T such that for all t $\geq T$: $\hat{\lambda}_{\max,t} \leq \lambda_{thr}$.*

Idea. The theorem states that if the instability (measured by $\hat{\lambda}_{\max,t}$) is driven by the magnitude of parameter updates, then multiplicatively decaying the learning rate whenever instability exceeds a threshold λ_{thr} will eventually bring the system back below the threshold.

Assume the system is at time t and $\hat{\lambda}_{\max,t} > \lambda_{\text{thr}}$. According to the rule, the learning rate for the next step η_{t+1} will be $\eta_t \cdot \beta$. The parameter update magnitude will be $\|\Delta\theta_{t+1}\| \leq \eta_{t+1} g_{\max} = \eta_t \beta g_{\max}$.

If the system remains unstable for n consecutive steps ($t, t+1, \dots, t+n-1$), the learning rate at step $t+n$ will be $\eta_{t+n} = \eta_t \beta^n$. The parameter update magnitude will be $\|\Delta\theta_{t+n}\| \leq \eta_{t+n} g_{\max} = \eta_t \beta^n g_{\max}$.

The spectral radius at step $t+n$ is bounded by:

$$\rho(J_{t+n}) \leq L_0 + L_1 \|\Delta\theta_{t+n}\| \leq L_0 + L_1 \eta_t \beta^n g_{\max}$$

We want to find an n such that $\rho(J_{t+n}) \leq \lambda_{\text{thr}}$. This requires:

$$L_0 + L_1 \eta_t \beta^n g_{\max} \leq \lambda_{\text{thr}}$$

$$L_1 \eta_t \beta^n g_{\max} \leq \lambda_{\text{thr}} - L_0$$
$$\beta^n \leq \frac{\lambda_{\text{thr}} - L_0}{L_1 \eta_t g_{\max}}$$

Taking logarithms (and noting $\log \beta < 0$):

$$n \log \beta \leq \log \left(\frac{\lambda_{\text{thr}} - L_0}{L_1 \eta_t g_{\max}} \right)$$

$$n \geq \frac{\log \left(\frac{\lambda_{\text{thr}} - L_0}{L_1 \eta_t g_{\max}} \right)}{\log \beta} = \frac{\log \left(\frac{L_1 \eta_t g_{\max}}{\lambda_{\text{thr}} - L_0} \right)}{\log(1/\beta)}$$

Since $\lambda_{\text{thr}} > L_0$ and $\beta < 1$, such a finite positive integer n (let's call it $n^\star$) exists. This means that after at most $n^\star$ consecutive steps of instability, the learning rate will have decayed sufficiently such that the spectral radius (and thus $\hat{\lambda}_{\max}$) is forced below λ_{thr}.

Once $\hat{\lambda}_{\max,t+n^\star} \leq \lambda_{\text{thr}}$, the indicator function $\mathbb{I}\{\hat{\lambda}_{\max,t+n^\star} > \lambda_{\text{thr}}\}$ becomes 0, and the learning rate stops decaying (or decays slower if instability re-emerges). The system will then operate with a smaller effective learning rate, keeping the parameter updates small enough to maintain $\hat{\lambda}_{\max}$ near or below λ_{thr}. The term "almost surely" accounts for stochastic fluctuations in gradients and state dynamics, but the deterministic decay mechanism ensures eventual stabilization under the given assumptions. The system stabilizes around the threshold λ_{thr}, potentially allowing operation near the edge of chaos ($\lambda_{\text{thr}} \approx 0$) if desired. □

References

[1] Isaac Landes and Samuel Berkebile. Distributed Predictive Latent Dynamics (DPLD): Initial Exploration and Motivation. arXiv:25xx.xxxxx (in preparation), 2025.

[2] Et al. Deco. Connectome Harmonics Analysis for Resting-state Manifolds (CHARM) reveals low-dimensional critical brain dynamics. Details approximate; citation refers to work on CHARM and low-dimensional critical brain manifolds, potentially preprint or forthcoming. Check for updates., 2025.

[3] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32 (NeurIPS 2019)*, pages 8026–8037, 2019.

[4] Teuvo Kohonen. The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480, 1990.

[5] Shun-Ichi Amari. Dynamics of pattern formation in lateral-inhibition type neural fields. *Biological Cybernetics*, 27(2):77–87, 1977.

[6] John J Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79(8):2554–2558, 1982.

[7] Karl Friston. The free-energy principle: a unified brain theory? *Nature Reviews Neuroscience*, 11(2): 127–138, 2010.

[8] Kürşat Tümer and Adrian Agogino. Understanding and using difference rewards in multiagent systems. In *Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS '07)*, pages 1–8, 2007.

[9] Adrian K Agogino and Kagan Tumer. Learning coordination policies using a difference reward. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS '04)*, volume 1, pages 280–287. IEEE, 2004.

[10] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3-4):229–256, 1992.

[11] Jürgen Schmidhuber. A possibility for implementing curiosity and boredom in model-building neural controllers. In *Proceedings of the International Conference on Simulation of Adaptive Behavior: From Animals to Animats (SAB '90)*, pages 222–227. MIT Press/Bradford Books, 1991.

[12] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018.

[13] Giancarlo Benettin, Luigi Galgani, Antonio Giorgilli, and Jean-Marie Strelcyn. Lyapunov characteristic exponents for smooth dynamical systems and for hamiltonian systems; a method for computing all of them. part 1: Theory. *Meccanica*, 15(1):9–20, 1980.

[14] Alan Wolf, Jack B Swift, Harry L Swinney, and John A Vastano. Determining lyapunov exponents from a time series. *Physica D: Nonlinear Phenomena*, 16(3):285–317, 1985.

[15] Rishi Gautam, Michael Ho, Leonardo Angelini, Daniele Marinazzo, Sebastiano Stramaglia, Mario Pellicoro, and Gustavo Deco. Critical dynamics of the resting human brain. *PLoS Computational Biology*, 11(7):e1004277, 2015.

[16] John M Beggs and Dietmar Plenz. Neuronal avalanches in neocortical circuits. *Journal of Neuroscience*, 23(35):11167–11177, 2003.

[17] Dante R Chialvo. Emergent complex neural dynamics. *Nature Physics*, 6(10):744–750, 2010.

[18] Bernard J Baars. *In the Theater of Consciousness: The Workspace of the Mind*. Oxford University Press, 1997.

[19] Stanislas Dehaene and Jean-Pierre Changeux. The global neuronal workspace model of conscious access: from neuronal architectures to clinical applications. In Stanislas Dehaene and Yves Christen, editors, *Characterizing Consciousness: From Cognition to the Clinic?*, pages 55–84. Springer, 2011.

[20] Murray P Shanahan. A cognitive architecture that combines internal simulation with a global workspace. *Consciousness and Cognition*, 15(2):433–449, 2006.

[21] Simon van Gaal and Victor AF Lamme. Computational implementations of the global workspace theory: A focused review. *Neuroscience & Biobehavioral Reviews*, 128:161–172, 2021.

[22] Giulio Tononi. An information integration theory of consciousness. *BMC Neuroscience*, 5(1):42, 2004.

[23] Masafumi Oizumi, Larissa Albantakis, and Giulio Tononi. From the phenomenology to the mechanisms of consciousness: integrated information theory 3.0. *PLoS Computational Biology*, 10(5):e1003588, 2014.

[24] Andrew M Haun and Giulio Tononi. What is integrated information? A guide to IIT. *arXiv preprint arXiv:2311.09498*, 2023.

[25] Andy Clark. Whatever next? Predictive brains, situated agents, and the future of cognitive science. *Behavioral and Brain Sciences*, 36(3):181–204, 2013.

[26] Alexander Tschantz, Beren Millidge, Anil K Seth, and Christopher L Buckley. Scaling active inference. *arXiv preprint arXiv:2006.04124*, 2020.

[27] Pablo Lanillos, Corrado Meo, Corrado Pezzato, Aswin V Meera, Mohamed Baioumy, Wataru Ohata, Stefan J Kiebel, Karl J Friston, Jun Tani, and Ricardo Galán. Active inference in robotics and artificial agents: Survey and challenges. *Frontiers in Robotics and AI*, 8:751878, 2021.

[28] Wolfgang Maass. Networks of spiking neurons: the third generation of neural network models. *Neural Networks*, 10(9):1659–1671, 1997.

[29] Peter Dayan and Bernard W Balleine. Reward, motivation, and reinforcement learning. *Neuron*, 36(2): 285–298, 2002.

[30] David J Chalmers. Facing up to the problem of consciousness. *Journal of Consciousness Studies*, 2(3): 200–219, 1995.

[31] Danijar Hafner, Julius Pasukonis, Jimmy Ba, and Timothy Lillicrap. Benchmarking the spectrum of agent capabilities. *arXiv preprint arXiv:2107.08931*, 2021.