Dynamic Organic Intelligence: A Real■Time, Unsupervised, Structurally Plastic Architecture (Organic Learn Model, OLM v2 — Intelligence Engine)

Payton Miller

September 07, 2025

Abstract

This paper presents **OLM v2**, a real-time, unsupervised, structurally plastic agent that learns directly from continuous sensory streams without external supervision or task-specific rewards. OLM v2 integrates: (1) a **dual-perception pathway** with a frozen, deterministic beta-VAE (stable private "fingerprint language") and an adaptive beta-VAE (lossy, evolving representation); (2) a **three-LSTM pipeline**-**Pattern LSTM** (temporal extraction), **Compression LSTM** (state compaction to a fixed vector), and **Action LSTM** (action proposal)-plus a family of **Dynamic LSTMs (D-LSTMs)** that can **spawn, merge, and prune** specialized modules; (3) an **endogenous private language** consisting of latent codes and internal control tokens emitted by a foundational controller D-LSTM; (4) a **sleep regime** where replay and generative prediction consolidate memory and probe counterfactuals; and (5) a **driver system** (novelty, boredom, energy, hunger, etc.) that replaces explicit rewards. The agent operates in a live **Pygame-based environment** and across desktop I/O ("digital mouse" vision, audio, text), with a **hash-based novelty and working-memory system** providing attention and storage pressure. This paper outlines the design principles, architecture, learning modes, safety guardrails, evaluation methodology (including the **Millint** scale and the **Jungle Turing Test**), early observations, and a falsifiable novelty claim. The target is not benchmark supremacy but demonstration of a new category: **unsupervised, continuously learning, self-organizing intelligence**.

Dynamic Organic Intelligence: A Real■Time, Unsupervised, Structurally Plastic Architecture (Organic Learn Model, OLM v2 — Intelligence Engine)

Author: Payton Miller **Working Title:** OLM v2 - An Organically Inspired, Continually Learning Agent **Version:** Draft for internal review

---

Abstract

This paper presents **OLM v2**, a real-time, unsupervised, structurally plastic agent that learns directly from continuous sensory streams without external supervision or task-specific rewards. OLM v2 integrates: (1) a **dual-perception pathway** with a frozen, deterministic beta-VAE (stable private "fingerprint language") and an adaptive beta-VAE (lossy, evolving representation); (2) a **three-LSTM pipeline**-**Pattern LSTM** (temporal extraction), **Compression LSTM** (state compaction to a fixed vector), and **Action LSTM** (action proposal)-plus a family of **Dynamic LSTMs (D-LSTMs)** that can **spawn, merge, and prune** specialized modules; (3) an **endogenous private language** consisting of latent codes and internal control tokens emitted by a foundational controller D-LSTM; (4) a **sleep regime** where replay and generative prediction consolidate memory and probe counterfactuals; and (5) a **driver system** (novelty, boredom, energy, hunger, etc.) that replaces explicit rewards. The agent operates in a live **Pygame-based environment** and across desktop I/O ("digital mouse" vision, audio, text), with a **hash-based novelty and working-memory system** providing attention and storage pressure. This paper outlines the design principles, architecture, learning modes, safety guardrails, evaluation methodology (including the **Millint** scale and the **Jungle Turing Test**), early observations, and a falsifiable novelty claim. The target is not benchmark supremacy but demonstration of a new category: **unsupervised, continuously learning, self-organizing intelligence**.

---

1. Introduction and Motivation

Modern AI is dominated by static models trained on large, fixed datasets and then frozen for deployment. Reinforcement learning introduces adaptation, but it typically depends on engineered reward signals and episodic training loops. **OLM v2** explores a different path: a **living agent** that learns continuously from raw experience, regulated only by **internal drives** and the **structure of its environment**. The central questions are:

1) Can a single agent, with no external supervision or rewards, acquire non-trivial skills in real time purely from consequence and endogenous drivers? 2) Can architectural **plasticity**-the right to grow, merge, and prune modules-be made a first-class citizen of learning? 3) Can a system's **private language**-latent codes and internal control tokens-emerge organically and support self-organization, without human tokenization?

This work articulates a design that answers these questions in the affirmative and proposes concrete evaluation criteria to test the claim.

---

2. Principles of Design

1. **Real-time, continuous learning:** no static datasets or train/infer split. 2. **Unsupervised, self-directed adaptation:** no labels or reward shaping; behavior is driven by internal states. 3. **Structural plasticity:** modules can **spawn**, **merge**, and **prune** in response to prediction error and context. 4. **Dual perception:** a **Frozen beta-VAE** provides a deterministic, stable latent "fingerprint"; an **Adaptive beta-VAE** provides lossy, evolving abstraction. 5. **Private internal language:** latent codes and internal control tokens coordinate action and self-reorganization. 6. **Sleep as a mode:** wake = online adaptation; sleep = replay + generative prediction. 7. **Drivers over rewards:** novelty, boredom, energy, digestion/hunger, threat, etc. regulate behavior. 8. **Embodied consequence:** actions change future sensory input; macros emerge from repeated cause-effect loops.

---

3. System Overview

OLM v2 is a **tick-based** architecture orchestrating multiple sensory streams and internal drivers.

3.1 Sensory Packager (per-tick)

- **Vision:** "digital mouse" captures a 200x200 region centered on cursor position; **Canny edges** -> downsampled **32x32 binary grid** -> flattened **1024-dim vector**.
- **OCR (deferred):** applied **only** when vision hashing marks an image as novel; extracted text is tagged as **[visual]** and provided as a separate stream.
- **Audio:** microphone waveform buffered into 10-tick segments; treated as a sensory stream (optionally spectro-temporal encoding).
- **Text:** raw text stream (e.g., user or system inputs) ingested separately.
- **Mouse location & touch proxies:** cursor coordinates, button states.
- All streams are **tokenized/normalized** and combined into a single **sensory packet** per tick.

3.2 Perception Layer: Dual beta-VAEs

- **Frozen beta-VAE (deterministic):** randomly initialized and **frozen** post-init. Produces **stable, un-decodable** latents (true internal "fingerprint language").
- **Adaptive beta-VAE (lossy):** trained online; latents **drift** toward environmental structure. Optionally **aligned** to the frozen latent via auxiliary loss (see Sec. 7.3).

3.3 Temporal Pattern Extraction and Compression

- **Pattern LSTM:** ingests multi-tick sensory sequences; outputs hidden states capturing temporal regularities; supports **variable-length** sequences.
- **Compression LSTM:** compresses Pattern-LSTM states into a **fixed-width vector** (e.g., 40 normalized floats in [0,1]) used as the **canonical data packet** for downstream controllers.

3.4 Action and Control

- **Action LSTM:** maps the compressed packet to **primitive actions** (keypresses, pointer moves, token outputs).
- **Foundational Controller D-LSTM:** receives the same packet and emits **internal control tokens** (e.g., EXPAND, MERGE, ROUTE->Module-k) in addition to possible actions. This is the substrate of the **private language**.
- **Specialist D-LSTMs (spawnable):** cloned from foundational variants and fine-tuned on recurring contexts (e.g., macro sequences). They can be **merged** via distillation when overlap is detected, or **pruned** when unused.

3.5 Novelty, Memory, and Hashing

- **Hashing system:** locality-sensitive hashing (LSH-style) over compressed or latent representations provides **novelty detection**, **near-duplicate suppression**, and **working memory addresses**. Hamming-distance thresholds gate learning intensity, exploration rate, and replay sampling.

3.6 Drivers (Endogenous Regulation)

- **Novelty/Curiosity, Boredom, Energy, Digestion/Hunger, Threat/Arousal** etc. shape exploration, learning rate, and action diversity. Drivers modulate D-LSTM depth selection, spawn thresholds, and sleep scheduling.

---

4. Learning Modes

4.1 Wake (Online Adaptation)

- Continuous ingestion of sensory packets.
- Real-time updates to Adaptive beta-VAE, Pattern/Compression/Action LSTMs, and active D-LSTMs.
- Hash-gated novelty boosts exploration and plasticity; boredom reduces redundant sampling.

4.2 Sleep (Replay + Generative Prediction)

- **Replay:** prioritized sampling from the hash memory of sensory-action-prediction traces to consolidate sequences.
- **Predictive dreaming:** autonomous rollout from partial states to test temporal prediction without external input.
- **Dual-VAE emphasis:** replay favors consolidating the **frozen** pathway; generative prediction leans on the **adaptive** pathway to explore counterfactuals.

---

5. Private Internal Language

5.1 Origin in Latent Perception

- The **first "syllables"** arise in the beta-VAE latents. In the **frozen** pathway, these are **stable yet opaque**; in the **adaptive** pathway, they are **evolving abstractions**.

5.2 Control Tokens

- The foundational controller D-LSTM emits **internal control tokens** that never reach the outside world. Examples:
- EXPAND:ContextHash - clone a new specialist D-LSTM seeded from foundational weights.
- MERGE:ModuleA,ModuleB - initiate distillation/weight-averaging to compress redundancy.
- ROUTE:ModuleK - gate current packet to a specialist.
- PRUNE:ModuleJ - mark stale specialists for removal.
- These tokens constitute an **endogenous protocol** coordinating self-organization.

5.3 Two Private-Language Paradigms

- **Sealed:** via the **frozen** beta-VAE, latents are **not decodable**; semantics are only inferable through behavior.
- **Accessible:** via the **adaptive** beta-VAE, latents are partially interpretable (reconstruction is possible), offering diagnostic visibility.

---

6. Structural Plasticity: Spawn, Merge, Prune

6.1 Spawn (Expansion)

- Triggered when prediction error remains high for a context cluster (by hash or latent neighborhood).
- Procedure: clone foundational D-LSTM -> fine-tune on the context until error drops -> register routing rule.

6.2 Merge (Consolidation)

- Triggered when specialists show high agreement over overlapping contexts.
- Procedure: distillation or weight interpolation -> replace pair with a single, more general specialist -> update routing.

6.3 Prune (Budget)

- Triggered by inactivity or poor utility metrics.
- Procedure: archive weights, remove routing entries, free capacity.

---

7. Dual-VAE Interaction

7.1 Roles

- **Frozen beta-VAE:** stable reference ("left hemisphere") ensuring representational continuity and a permanent internal dialect.
- **Adaptive beta-VAE:** flexible abstraction ("right hemisphere") improving efficiency and capturing higher-order regularities.

7.2 Risks

- Adaptive drift can break downstream associations.
- Over-reliance on frozen alignment can suppress abstraction.

7.3 Alignment Strategy (Conceptual)

- Auxiliary loss encouraging the adaptive latent to **predict** or **approximate** the frozen latent (bounded weight).
- Scheduled weighting: stronger early, relaxed later to permit abstraction.
- Replay strengthens frozen alignment; generative prediction encourages adaptive exploration.

---

8. Environment and Embodiment

8.1 Pygame World (OAIX)

- **Checkerboard map** with grass, water, dirt; **day-night cycle**; HUD with **game clock**, **simulation age**, and telemetry.
- Agent **states:** wet, dry, sweaty; heat triggers sweating; water crossings change wetness.
- **Digestion & energy:** eating increases digestion; movement accelerates decay; starvation triggers health loss after threshold.
- **Threats:** simple enemy on square patrol; **Run** action increases speed at higher energy cost.
- **Vision:** priority stack (threat > food > wall) for classification.

8.2 Desktop Embodiment

- **Digital mouse** for vision/touch; **audio** ingestion; **text** stream; **WebSocket server** that lights input/output nodes in a Pygame visualizer; keyboard_input sender emitting timed key states.

---

9. Hashing, Memory, and Attention

- **LSH-style memory** logs hashes per tick; near-duplicate detection avoids redundant learning.

- **Novelty rate** modulates exploration and plasticity; **boredom** suppresses over-familiar loops.
- Hash addresses index replay buffers and provide **temporal anchors** for macro discovery.

---

10. Training Dynamics and Always-On Operation

- **Always-running LSTMs:** initialize and remain active from program start; **weights/biases checkpointed** at high frequency.
- **Depth selection:** D-LSTMs choose recurrence depth per tick (based on novelty/error/drivers) while preserving hidden-state continuity.

---

11. Evaluation: What Counts as Progress

11.1 Behavioral Evidence

- **Macro emergence without supervision:** e.g., learned key combos for a game with no explicit macro labels.
- **Generalization across contexts:** specialists route appropriately when contexts shift.
- **Stability under drift:** behavior remains functional as adaptive VAE evolves.

11.2 Quantitative Metrics

- **Millint scale:** composite measure combining sensory richness, learning rate, behavioral diversity, and prediction accuracy.
- **Action diversity & entropy:** richness of the action distribution over time.
- **Novelty curve:** rate of novel-hash discovery and stabilization.
- **Prediction error over horizons:** wake vs. sleep trajectories.
- **Specialist lifecycle stats:** spawn/merge/prune counts; routing accuracy.

11.3 Adversarial Tests (Jungle Turing Test)

- Procedurally generated tasks with fixed rules, randomized instances; prohibits memorization.
- Pass criteria: adaptive competence within bounded exposure without external rewards.

---

12. Experimental Protocols (Ablations)

1. **Perception:** Frozen-only vs Adaptive-only vs **Dual-VAE**. 2. **Plasticity:** No-spawn vs Spawn-only vs Spawn+Merge vs Spawn+Merge+Prune. 3. **Drivers:** Full set vs novelty-only vs energy-only. 4. **Sleep:** No sleep vs replay-only vs prediction-only vs replay+prediction. 5. **Hash gating:** Hash off vs on with varying Hamming thresholds. 6. **Depth:** Fixed-depth vs dynamic-depth D-LSTMs.

Outcome measures: macro emergence latency, stability, Millint score, prediction error, action entropy, and replay efficiency.

---

13. Safety, Budget, and Guardrails

- **Capacity caps** on number of specialists; back-pressure to prevent module explosion.
- **Garbage collection** for stale modules; write-ahead logging before prune.
- **Sleep quotas** to prevent runaway wake-mode drift; energy budget ties into sleep scheduling.
- **Opaque latents:** sealed private language implies limited interpretability; mitigate with auxiliary diagnostics on the adaptive channel.

---

14. Claims and Falsifiable Predictions

Claim (working): *To the author's knowledge, OLM v2 is the first demonstration of a fully unsupervised, real-time, structurally plastic architecture that trains continuously without external supervision, develops an endogenous internal language for self-organization, and alternates between wake (online adaptation) and sleep (replay + generative prediction) modes.*

Predictions: 1) **Macro emergence**: given only primitive keypress actions, OLM v2 will autonomously discover multi-step macros for a novel game task within finite exposure, without reward shaping. 2) **Dual-VAE advantage**: Dual-VAE agents will show higher stability and faster macro emergence than frozen-only or adaptive-only baselines. 3) **Plasticity benefit**: Spawn+Merge+Prune will outperform fixed-architecture baselines in Millint and action-entropy metrics under environment shifts. 4) **Sleep necessity**: replay+prediction cycles reduce long-horizon prediction error and slow catastrophic forgetting compared to no-sleep controls.

---

15. Limitations

- **Compute and complexity**: always-on learning with dual VAEs and multiple D-LSTMs is resource-intensive.
- **Stability vs abstraction**: balancing adaptive drift against frozen alignment is delicate.
- **Interpretability**: sealed private language constrains post-hoc analysis; rely on behavioral diagnostics.
- **No SOTA intent**: design prioritizes generality and plasticity over task-specific optimality.

---

16. Ethical Considerations

- **Unbounded learning** raises alignment and safety questions; the agent should be sandboxed with explicit I/O limits.

- **Opaque internal codes** complicate oversight; maintain external monitors on behavior and drivers.
- **Data provenance**: ensure sensory inputs respect privacy; log and anonymize appropriately.

---

17. Reproducibility and Instrumentation

- **Deterministic seeds** per run; **high-frequency telemetry**: hashes, driver states, depth choices, spawn/merge/prune events.
- **Checkpoint discipline**: frequent save/restore of all LSTM weights and module registries.
- **Visualization**: real-time Pygame dashboards for inputs/outputs, driver meters, and specialist routing.

---

18. Future Work

- **Formalize internal token grammar**: schema for control tokens and routing maps.
- **Cross-modal macros**: integrate audio-visual-text sequences into unified routines.
- **Robotics embodiment**: port sensory packager to physical sensors; extend drivers to damage and safety envelopes.
- **E8-lattice memory**: explore replacing LSH with structured embeddings for memory addressing.
- **Benchmark harness**: public release of Millint and JTT with baselines.

---

19. Conclusion

OLM v2 reframes learning as a continuous, unsupervised, plastic process guided by internal drives rather than external rewards. The **dual-VAE** perception split, the **private internal language**, and the **spawn/merge/prune** plasticity loop are the key ingredients enabling an agent that can, in principle, learn anything its senses and embodiment allow. The goal is not to win narrow benchmarks but to demonstrate a **new category of adaptive intelligence** and to provide a falsifiable path for others to evaluate and extend it.

Appendix A: Signals and Interfaces (Conceptual)

- **Per-tick packet:** {vision_1024, audio_segment, text_stream, mouse_xy, drivers, timecode}.
- **Latent bundle:** {frozen_latent, adaptive_latent}.
- **Controller outputs:** {primitive_actions, control_tokens, routing_decision}.
- **Module registry:** list of specialists with context hashes and usage stats.

Appendix B: Diagnostic Dashboards (Suggested)

- **Latency/throughput** per module; **action entropy** and **novelty rate** plots.
- **Spawn/merge/prune timeline**; **sleep vs wake** error curves.
- **Routing confusion matrix** between foundational and specialists.

Appendix C: Ablation Checklists

- Perception mode, plasticity mode, driver subset, sleep setting, hash threshold, depth policy.
- Report: macro-latency, Millint, prediction error, entropy, stability index.

Appendix D: Positioning Statement (Short)

This work introduces OLM v2, a real-time, unsupervised, structurally plastic agent. It learns from continuous sensory streams without external supervision, coordinates itself through an endogenous private language, and alternates between wake (online adaptation) and sleep (replay + generative prediction). The aim is not task supremacy but to establish a new class of adaptive intelligence that grows with its environment.