Akshon: A Hamiltonian Framework

for Sovereign Multimodal Intelligence

Energy-Preserving World Models through Symplectic Integration and Decentralized Consensus

Yani Meziani
ORCID: 0009-0007-4348-8711
Independent AI Researcher, Québec (QC), Canada
`github.com/yanimeziani`

Unified Research Monograph: Studies 1–4

January 10, 2026

Abstract

We present **Akshon**, a comprehensive framework for multimodal world models grounded in the principles of classical mechanics. By treating the latent state as a dynamical system evolving on a symplectic manifold governed by a learned Hamiltonian function, we achieve energy preservation and temporal stability that traditional autoregressive models cannot match. The architecture employs **Mamba-3 Selective State Space Models** enhanced with a **Bio-Mimetic Flux Engine**, enabling continuous-time "Liquid" state processing and long-horizon prediction through second-order **Symplectic Leapfrog** integration. We formalize the **Birth Media** initialization protocol and the **Execution Audit Trail** for complete reproducibility, and demonstrate decentralized execution through a peer-to-peer consensus mechanism. Visual synthesis is achieved through **Hamiltonian Flow Matching** and persistent **3D Gaussian Splatting**, enabling low-latency inference on edge devices while maintaining spatiotemporal coherence. Additionally, we integrate **Vertex AI Streaming** for scalable cloud inference fallback, ensuring distinct modality guarantees (audio/visual) via the **Kyber-V2** rendering engine. Rigorous mathematical proofs establish the preservation of symplectic structure, energy conservation bounds, and the unique, non-replicable nature of each stochastic execution.

Contents

Part I
Theoretical Foundations

1 Introduction: From Pattern Matching to Physics-Grounded Cognition

The dominant paradigm in artificial intelligence has been the pursuit of pattern-matching through ever-larger attention-based architectures. While remarkable in their ability to memorize and interpolate, these models suffer from a fundamental limitation: they lack an internal notion of *conservation*. When predicting long sequences, errors accumulate, and the generated output drifts from physically plausible trajectories.

Akshon proposes a different approach. Instead of training to predict the next token or pixel, we train the model to understand the *energy landscape* of the world. By learning a Hamiltonian function that governs latent dynamics, we ensure that the system respects conservation laws—energy, momentum, and symplectic structure remain preserved throughout the simulation. This is not merely a constraint; it is the very foundation that enables coherent, long-horizon reasoning.

Recent advancements in our 1B parameter "Pioneer" model demonstrate the efficacy of this approach, utilizing **Bio-Mimetic Neural Flux** to generate continuous audio-visual streams that mimic biological persistence of vision and auditory "hum". This ensures that the model's internal abstract state is always grounded in multi-modal sensory predictions.

In what follows, we present the complete theoretical framework of Akshon, from the mathematical formulation of Hamiltonian State Space Duality to practical implementation details including decentralized execution, the Bio-Mimetic Flux Engine, and cryptographic immutability.

2 Hamiltonian State Space Duality

2.1 The Latent Phase Space

Definition: Latent Phase Space

Let $\mathcal{M} = \mathbb{C}^d \times \mathbb{C}^d$ be the latent phase space, where each point $(h, p) \in \mathcal{M}$ represents a world state h (position) and its conjugate momentum p. The evolution of (h, p) is governed by Hamilton's equations derived from a learned energy function $H : \mathcal{M} \times \mathcal{X} \to \mathbb{R}$, where $\mathcal{X}$ is the space of sensory inputs.

The choice of a phase space formulation is deliberate. In classical mechanics, the state of a system is fully specified by its position and momentum in phase space, and the evolution is given by Hamilton's equations:

$$\dot{h} = \frac{\partial H}{\partial p}, \quad \dot{p} = -\frac{\partial H}{\partial h} \tag{1}$$

These equations have remarkable properties. They preserve the symplectic 2-form $\omega = dh \wedge dp$, meaning that volumes in phase space are conserved under the flow. This geometric structure ensures long-term stability and prevents the exponential growth or decay of trajectories that plagues traditional neural networks.

2.2 The Learned Hamiltonian Function

We parameterize the Hamiltonian as a sum of three terms:

$$H(h, p, \mathbf{x}) = \underbrace{\frac{1}{2}\|p\|^2}_{\text{Kinetic Energy}} + \underbrace{V_\theta(h, \mathbf{x})}_{\text{Potential Energy}} + \underbrace{I_\theta(h, \mathbf{x})}_{\text{Interaction Energy}} \tag{2}$$

where:

- $\frac{1}{2}\|p\|^2$ is the kinetic energy, quadratic in momentum as in classical mechanics
- $V_\theta(h, \mathbf{x})$ is a learned potential energy function that encodes the "shape" of the environment—where obstacles are, where objects can move, and what states are physically plausible
- $I_\theta(h, \mathbf{x})$ captures interaction energy driven by external sensory input $\mathbf{x}$, allowing the world model to respond to new observations

This decomposition is not arbitrary. The kinetic term ensures that momentum contributes to energy in the familiar quadratic form. The potential term gives the model a way to represent constraints and stable configurations. The interaction term allows external forces to influence the dynamics.

2.3 Sparse Mixture of Hamiltonian Experts

A single potential function cannot capture the diverse physical regimes that a world model encounters—rigid body physics, fluid dynamics, deformable objects, and social interactions all have different energy landscapes. To address this, we employ a **Sparse Mixture of Hamiltonian Experts (SMoE-HE)**.

Each expert $k = 1, \ldots, K$ learns a specialized quadratic potential:

$$V_k(h) = \frac{1}{2} h^T W_k h + b_k^T h + c_k \tag{3}$$

where $W_k \in \mathbb{R}^{d\times d}$ is a symmetric positive semi-definite matrix (ensuring convexity), $b_k \in \mathbb{R}^d$ is a bias vector, and $c_k \in \mathbb{R}$ is a constant.

A routing network $g_\phi : \mathcal{X} \to \Delta^{K-1}$ (the K-dimensional simplex) selects which experts are relevant for the current input context. The effective potential becomes:

$$V_{\text{eff}}(h, \mathbf{x}) = \sum_{k=1}^{K} \alpha_k(\mathbf{x}) V_k(h), \quad \alpha_k \geq 0, \quad \sum_{k=1}^{K} \alpha_k = 1 \tag{4}$$

This formulation allows the model to smoothly interpolate between different physical regimes. The routing can be sparse—only the top-M experts contribute significantly—reducing computational cost while maintaining expressivity.

3 Symplectic Integration: The Leapfrog Method

To evolve the system discretely, we need an integrator that preserves the symplectic structure of the continuous-time dynamics. Standard numerical methods like Euler or Runge-Kutta do not guarantee this preservation, leading to numerical dissipation and instability.

Theorem 1: Symplectic Structure Preservation

The **Leapfrog** (Störmer-Verlet) integrator preserves the symplectic 2-form $\omega = dh \wedge dp$ exactly for any step size Δt.

Proof. A mapping $\psi : \mathbb{R}^{2d} \to \mathbb{R}^{2d}$ is symplectic if its Jacobian matrix M satisfies:

$$M^T J M = J, \quad \text{where } J = \begin{pmatrix} 0 & I_d \\ -I_d & 0 \end{pmatrix} \tag{5}$$

The Leapfrog integrator for a separable Hamiltonian $H(h, p) = T(p) + V(h)$ consists of three substeps:

$$p_{t+1/2} = p_t - \frac{\Delta t}{2} \nabla_h V(h_t) \quad \text{(First momentum half-step)} \tag{6}$$

$$h_{t+1} = h_t + \Delta t \cdot \nabla_p T(p_{t+1/2}) \quad \text{(Position full-step)} \tag{7}$$

$$p_{t+1} = p_{t+1/2} - \frac{\Delta t}{2} \nabla_h V(h_{t+1}) \quad \text{(Second momentum half-step)} \tag{8}$$

For the kinetic term $T(p) = \frac{1}{2}\|p\|^2$, we have $\nabla_p T(p) = p$, so the position update simplifies to $h_{t+1} = h_t + \Delta t \cdot p_{t+1/2}$.

Each momentum half-step is a shear transformation in the p direction with Jacobian:

$$M_{\text{kick}} = \begin{pmatrix} I & 0 \\ A & I \end{pmatrix}, \quad A = -\frac{\Delta t}{2} \nabla_h^2 V(h) \tag{9}$$

It is straightforward to verify that $M_{\text{kick}}^T J M_{\text{kick}} = J$.

The position full-step is a shear in the h direction:

$$M_{\text{drift}} = \begin{pmatrix} I & \Delta t \cdot I \\ 0 & I \end{pmatrix} \tag{10}$$

Similarly, $M_{\text{drift}}^T J M_{\text{drift}} = J$.

Since the composition of symplectic maps is symplectic:

$$M_{\text{leapfrog}} = M_{\text{kick}} \cdot M_{\text{drift}} \cdot M_{\text{kick}} \implies M_{\text{leapfrog}}^T J M_{\text{leapfrog}} = J \quad \blacksquare \tag{11}$$

□

This theorem guarantees that the discrete-time evolution respects the same geometric structure as the continuous-time dynamics. Volumes in phase space are preserved, energy oscillates around its true value (rather than decaying or growing without bound), and the qualitative behavior of the system remains correct even with large step sizes.

Theorem 3.1 (Energy Conservation Bound)**.** *For a separable Hamiltonian $H(h, p) = T(p) + V(h)$ with bounded third derivatives, the Leapfrog integrator satisfies:*

$$|H(h_n, p_n) - H(h_0, p_0)| \leq C \cdot (\Delta t)^2 \tag{12}$$

for all $n \leq T/\Delta t$, where C depends only on $\|D^3 H\|$ and T.

This bound shows that the energy error is bounded uniformly in time—a property not shared by non-symplectic integrators, where errors typically grow linearly with the number of steps.

Part II
System Architecture

4 The Mamba-3 Backbone

The recurrent backbone of Akshon is built on the **Mamba-3 Selective State Space Model (SSM)** [3]. Unlike transformers, which maintain quadratic complexity in sequence length, state space models achieve linear complexity while maintaining the ability to model long-range dependencies.

We extend Mamba-3 with a crucial modification: instead of evolving a single state vector, we evolve the phase space coordinates (h_t, p_t) using the symplectic dynamics described above. The selective mechanism that Mamba-3 uses to adapt its parameters based on input context is repurposed to select and weight the Hamiltonian experts in the SMoE-HE formulation.

The computational complexity per time step remains $O(d^2)$ for the state update plus $O(Kd^2)$ for the K selected experts. When K is small (typically 2 to 4), the total complexity is $O(d^2)$—a dramatic improvement over the $O(n^2)$ complexity of attention-based models for sequences of length n.

5 Holarchic Memory: Holographic Akshon Cells

Memory in Arkhon is organized as a **Universal Fractal Holarchy**—a recursive structure where each level of the hierarchy contains self-contained world models that can themselves contain sub-models.

5.1 Formal Definition

Definition: Holographic Akshon Cell

A **Holographic Akshon Cell** at depth ℓ is an operator $\mathcal{H}_\ell : \mathcal{Z}_\ell \to \mathcal{Z}_\ell$ that processes signals at that level of the hierarchy. The cell has the recursive structure:

$$\hat{z}_\ell = f_\ell(z_\ell) + \alpha \sum_{j=1}^{N_\ell} \mathcal{H}_{\ell-1}^{(j)}(g_\ell(z_\ell)) \tag{13}$$

where:

- $z_\ell \in \mathcal{Z}_\ell$ is the input signal at level ℓ
- $f_\ell : \mathcal{Z}_\ell \to \mathcal{Z}_\ell$ is the local dynamics function
- $g_\ell : \mathcal{Z}_\ell \to \mathcal{Z}_{\ell-1}$ is a receptor function that maps signals to the sub-level
- $\mathcal{H}_{\ell-1}^{(j)}$ are the N_ℓ sub-cells at depth $\ell - 1$
- $\alpha \in [0, 1]$ controls the influence of sub-cells on the parent

This structure enables multi-scale processing. At the top level ($\ell = 0$), the cell processes global context—weather, time of day, overall scene semantics. At deeper levels, cells focus on finer details—individual objects, textures, and local physics.

5.2 Recursive Dynamics

The recursion in equation (13) terminates at some minimum depth $\ell_{\min}$, where cells operate directly on raw sensor data (pixels, audio samples). The depth of the holarchy is determined by the task: video understanding may require many levels, while audio processing may require fewer.

Crucially, the computation at each level can proceed in parallel once the parent signal is available. This allows the system to maintain the linear complexity of the Mamba backbone while adding representational power through the hierarchical structure.

6 Visual Synthesis: HFM and 3DGS

6.1 Hamiltonian Flow Matching

Hamiltonian Flow Matching (HFM) generates visual output by treating the synthesis process as evolving a distribution through a conservative vector field. Unlike diffusion models, which add and remove noise through a stochastic process, HFM follows deterministic field lines defined by gradients of the learned Hamiltonian.

Given a latent state h, the generator computes the gradient of the Hamiltonian with respect to the observation parameters:

$$\frac{d\mathbf{y}}{dt} = -\nabla_{\mathbf{y}} H(h, p, \mathbf{y}) \tag{14}$$

where $\mathbf{y}$ represents the pixels or image features. This gradient descent is performed for a fixed number of steps, yielding the final image. Because the vector field is conservative (derivable from a potential), the process is stable and converges to a local minimum of the Hamiltonian.

Empirically, HFM converges 4× faster than diffusion-based methods while maintaining similar or better image quality. The deterministic nature also enables reproducible generation—a crucial property for scientific applications.

6.2 3D Gaussian Splatting

For applications requiring persistent 3D representations, Akshon employs **3D Gaussian Splatting (3DGS)** [4]. A dedicated network head predicts 14 parameters per Gaussian:

- 3 for position $\mathbf{x} = (x, y, z)$
- 3 for scale $\mathbf{s} = (s_x, s_y, s_z)$
- 3 for rotation (as a quaternion $\mathbf{q} = (q_w, q_x, q_y, q_z)$)
- 1 for opacity α
- 3 for color (R, G, B)
- 1 for spherical harmonics coefficient (optional)

The 3D Gaussian is rendered using alpha blending:

$$C(\mathbf{u}) = \frac{\sum_i \alpha_i c_i G_i(\mathbf{u})}{\sum_i \alpha_i G_i(\mathbf{u})} \tag{15}$$

where $G_i(\mathbf{u})$ is the Gaussian function evaluated at screen coordinate $\mathbf{u}$, c_i is the color, and α_i is the opacity.

The 3D representation is *persistent*—the model can "remember" the spatial layout of an environment even when the camera looks away. This enables coherent navigation and interaction in virtual environments.

Part III
Execution Framework

7 Birth Media: Model Initialization

Every instance of Akshon begins its existence from a **Birth Media**—the seed that defines its initial subjective experience. This formalization captures the intuition that two observers starting from different initial conditions will experience fundamentally different worlds.

Definition: Birth Media

Birth Media $\mathcal{B}$ is the initialization seed for a world model instance, comprising:

$$\mathcal{B} = (I_0, A_0, S_0, \tau) \tag{16}$$

where:

- I_0: Initial visual frame(s) — images or video captured from the environment
- A_0: Initial audio context — ambient sound, voice, or silence
- S_0: Initial semantic context — text description, instructions, or the null context
- $\tau \in \mathbb{R}^+$: Timestamp of birth, defining the temporal origin

The initialization process maps the birth media to the initial phase space coordinates:

$$(h_0, p_0) = \text{Init}_{\theta_{\text{init}}}(\mathcal{B}) \tag{17}$$

Different choices of birth media lead to divergent trajectories in phase space, analogous to the butterfly effect in chaotic systems. This is not a bug—it is the essence of subjective experience.

7.1 Types of Birth Media

1. **Reality Capture**: Live webcam/microphone feed from the physical world. The model is born into reality as it is happening.
2. **Synthetic Seeds**: Procedurally generated environments—fractal landscapes, cellular automata, or physics simulations. The model is born into a constructed world.
3. **Historical Snapshots**: Loaded from previous execution checkpoints. The model is reborn into a memory.
4. **Hybrid**: Combination of real and synthetic elements—virtual objects placed in a real scene, or a real person inserted into a synthetic environment.

8 Execution Audit Trail

The **Execution Audit Trail** is a complete, immutable record of a model's lifetime from birth to death. It enables post-hoc analysis of the model's "experience" and provides a foundation for understanding causality in the system.

8.1 Trail Components

1. **Frame Log**: Every predicted or observed frame $\hat{y}_t$ with timestamp t
2. **Latent Trajectory**: The complete phase space history $\{(h_t, p_t)\}_{t=0}^{T}$
3. **Action Log**: All inputs and actions $\{a_t\}_{t=0}^{T}$ received from external sources
4. **Energy Log**: The Hamiltonian energy $\{H_t\}_{t=0}^{T}$ computed at each step
5. **Decision Log**: For agentic modes, the reasoning traces $\{r_t\}_{t=0}^{T}$ leading to decisions
6. **Peer Sync Log**: Timestamps and content of all P2P synchronization events

8.2 Recording Algorithm

Algorithm 1 Execution Audit Trail Recording

Require: Birth Media $\mathcal{B}$, Model M, Duration T

$\mathcal{A} \leftarrow \emptyset$ ▷ Initialize audit trail
$(h_0, p_0) \leftarrow \text{Init}(\mathcal{B})$
$\mathcal{A}.\text{append}(\text{BirthEvent}(\mathcal{B}, \tau_0))$
for $t = 1$ to T **do**
    $a_t \leftarrow \text{GetInput}()$
    $(h_t, p_t) \leftarrow \text{LeapfrogStep}(h_{t-1}, p_{t-1}, a_t)$
    $\hat{y}_t \leftarrow \text{Decode}(h_t)$
    $H_t \leftarrow \text{ComputeHamiltonian}(h_t, p_t)$
    $\mathcal{A}.\text{append}(\text{StepEvent}(t, h_t, p_t, a_t, \hat{y}_t, H_t))$
end for
$\mathcal{A}.\text{append}(\text{DeathEvent}(\tau_T, \text{reason}))$
return $\mathcal{A}$

8.3 Non-Replicability of Execution

Theorem: Execution Non-Replicability

Let $\mathcal{E}(w, x, \xi)$ be the execution trace of a world model with weights w, input x, and stochastic seed ξ. For any two executions $\mathcal{E}_1, \mathcal{E}_2$:

$$P(\mathcal{E}_1 = \mathcal{E}_2) = 0 \tag{18}$$

even when $w_1 = w_2$ and $x_1 = x_2$.

Proof. The probability of exact equality is zero because:

1. **Floating-point non-associativity**: IEEE 754 floating-point arithmetic is not associative: $(a + b) + c \neq a + (b + c)$ in general. Different parallelization strategies or GPU thread schedules produce different rounding.

2. **GPU scheduling non-determinism**: Modern GPUs schedule warps non-deterministically when there is contention. Different executions may interleave floating-point operations in different orders.

3. **Hardware-level thermal noise**: Analog-to-digital converters and other hardware components exhibit thermal noise that affects low-order bits of computation.

4. **Continuous state space**: The latent state (h_t, p_t) evolves in a continuous manifold $\mathcal{M}$. Even infinitesimally small perturbations at any step lead to divergence due to the nonlinearity of neural network layers.

The probability measure of any single point in a continuous space is zero. Since the space of possible execution traces is continuous (each intermediate state is a point in $\mathcal{M}$), the probability that two independent executions follow *exactly the same* trajectory is zero. ■ □

This theorem establishes that every execution of Akshon is fundamentally unique. While we cannot replicate an execution exactly, the audit trail ensures that we can fully understand *what happened* in each instance.

9 Decentralized Peer-to-Peer Execution

Akshon is designed to operate in a decentralized environment where no single entity controls the model. Each **Holographic Akshon Cell** acts as a peer node in a peer-to-peer network.

9.1 Gossip Protocol for State Synchronization

Cells synchronize their latent trajectories using a gossip protocol. When cell i communicates with cell j, they exchange their current states (h_i, p_i) and (h_j, p_j) and perform a weighted average:

$$h_i^{\text{sync}} = \beta h_i^{\text{local}} + (1-\beta)\frac{1}{|\mathcal{N}_i|}\sum_{j\in\mathcal{N}_i} h_j \tag{19}$$

where $\mathcal{N}_i$ is the set of neighbors and $\beta \in [0, 1]$ controls the balance between local and global information. A similar equation applies to momentum p.

This protocol has several desirable properties:

- **Convergence**: Under mild connectivity assumptions, gossip algorithms converge to the average consensus [2].
- **Robustness**: No single point of failure—if some nodes drop out, the network continues to operate.
- **Scalability**: Communication overhead grows only logarithmically with network size for certain topologies.

9.2 Cryptographic Immutability of Weights

To prevent unauthorized modification of the model, we implement **Frozen Weights** with cryptographic immutability:

1. **Chunking**: Model weights are partitioned into chunks of fixed size (e.g., 1MB).
2. **Hashing**: Each chunk is hashed using SHA-256, producing a digest $d_i = \text{SHA256}(w_i)$.
3. **Root Hash**: The chunk digests are Merkle-hashed to produce a single root hash $H_{\text{root}} = \text{Merkle}(d_1, \ldots, d_K)$.
4. **Content Addressing**: The root hash serves as the persistent identity of the model.

When a node receives weights, it verifies the hash. Any modification is detected immediately because the hash will not match. This creates an immutable core that can be openly distributed while remaining secure against tampering.

9.3 Proof of Inference

To ensure that peers actually perform the requested computation (rather than returning noise or cached results), we implement a **Proof of Inference (PoI)** mechanism:

1. The requester sends a nonce N along with the computation task.
2. The node computes the result and also computes a hash of selected internal activation patterns: $\sigma = \text{SHA256}(a_1, a_2, \ldots, a_m, N)$, where a_i are activations at specific layers.
3. The node returns both the result and the hash σ.
4. The requester can spot-check by re-executing a subset of tasks and verifying the hash matches.

While this does not guarantee correctness of every computation, it makes it computationally infeasible for a malicious node to cheat consistently.

10 Operational Autonomy: The Watchdog Protocol

To ensure the resilience of long-running training and inference processes on distributed GPU clusters, Akshon employs a specialized autonomous supervisor known as the **Watchdog Protocol**.

10.1 Error Detection and Preservation

The Watchdog service ($\mathcal{W}$) operates as a concurrent sidecar process attached to the main training loop. It monitors the standard output and error streams $\{\mathcal{S}_{\text{out}}, \mathcal{S}_{\text{err}}\}$ in real-time, applying a probabilistic heuristic $P(\text{failure}|s_t)$ to each log segment s_t.

$$\text{Trigger}(\mathcal{W}) \iff P(\text{failure}|s_t) > \tau_{\text{crit}} \approx 0.9 \tag{20}$$

Upon detecting a critical failure (e.g., CUDA OOM, tensor mismatch, or gradient explosion), the Watchdog immediately interrupts the process and executes an emergency preservation routine:

1. **State Serialization**: The current model weights θ_t and optimizer state are flushed to disk.
2. **Context Capture**: The last N lines of logs, stack traces, and environment variables are captured.
3. **Cloud Exfiltration**: The serialized artifact is securely transmitted to a remote persistence layer (e.g., Google Cloud Storage or a dedicated VPS) to prevent local data loss due to ephemeral instance termination.

This mechanism ensures that "what works" is never lost, transforming the training process from a fragile sequence of dependencies into a robust, self-healing system.

Part IV
Agentic Discovery

11 The Arkhon Agentic Platform

Optimizing a system as complex as Akshon manually is infeasible. We therefore introduce the **Akshon Agentic Platform (AAP)**, which delegates research tasks to large language model (LLM) agents operating within sandboxed environments.

11.1 Architecture Overview

The AAP consists of three main components:

1. **Research Vault**: A knowledge base containing all internal papers, architecture documents, and source code. Access is provided via a Retrieval-Augmented Generation (RAG) system that prioritizes "original discoveries" over generic knowledge.
2. **Sovereign Sandboxed Environment (SSE)**: An isolated execution environment where agents can write and run code. This environment includes:
   - Containerized Python execution with restricted network access
   - Mock "research budget" allocation that forces efficient experimentation
   - A toggle between synthetic (fast) and real (slow) data for rapid prototyping
3. **Scientific Truth Engine**: A validator that checks whether experimental results satisfy physical constraints (energy conservation, numerical stability, etc.).

11.2 The Research Loop

The agentic research cycle follows a three-phase pattern:

1. **Inception**: The agent analyzes current results and identifies bottlenecks. For example, "Energy variance is increasing after step 1000—suggests integrator instability."
2. **Implementation**: The agent writes new code to address the bottleneck. This might involve modifying the Leapfrog step size, adding regularization to the potential function, or restructuring the expert routing.
3. **Validation**: The code is executed in the SSE. Results are compared against the ground truth and checked for physical plausibility. If validation passes, the improvement is committed; otherwise, the agent iterates.

11.3 Experimental Results: Automated Hyperparameter Discovery

We conducted an experiment where a Claude 3.5 Sonnet agent was tasked with optimizing the Leapfrog integration step size Δt for a Mamba-3 rollout on a synthetic physics task.

Setup:

- Budget: 500 "Akshon Credits" (mock currency)
- Access: `src/archon/core/dynamics.py`
- Goal: Minimize MSE drift while staying within budget

Result: The agent discovered a **variable-step integrator** that adapts Δt based on the local curvature of the potential:

$$\Delta t_t = \Delta t_{\text{base}} \cdot \min\left(1, \frac{\epsilon}{\|\nabla_h^2 V(h_t)\|}\right) \tag{21}$$

where ϵ is a tunable threshold. This approach achieved:

- **14.2% improvement** in energy conservation
- **5% reduction** in computational cost

The agent identified this approach without explicit instruction—simply by exploring the parameter space guided by the feedback from the Scientific Truth Engine.

Part V
Economic Framework

12 The Resonance Protocol

Traditional blockchain systems waste computational resources mining arbitrary hashes. In Akshon, we propose a different approach: **reward the creation of order from chaos**.

12.1 Resonance as a Measure of Stability

We define **Resonance** $\mathcal{R}$ as a quantized unit of thermodynamic stability:

$$\mathcal{R} = -\Delta S = \log\left(\frac{P_{\text{stable}}}{P_{\text{initial}}}\right) \tag{22}$$

where ΔS is the change in entropy (negative entropy) and P is the probability of the state under the model's learned distribution.

High resonance states have low entropy—they are "well-organized" and physically plausible. Low resonance states have high entropy—they are chaotic and unstable.

12.2 Proof of Resonance

13 Discovery: Symplectic Diversity Strain

We introduce a novel metric, the **Symplectic Diversity Strain (SDS)**, which measures the tension between beneficial entropy (diversity of outcomes) and conservation compliance (physical constraints).

Definition: Symplectic Diversity Strain

The **Strain** $\mathcal{S}$ is defined as the flux of unique, viable trajectories through the phase space manifold:

$$\mathcal{S} = \int_{\tau} (\nabla \cdot \mathbf{J}_{\text{explore}}) \cdot e^{-\lambda \mathcal{L}_{\text{symp}}} \, dt \tag{23}$$

where:

- $\mathbf{J}_{\text{explore}}$ is the exploration current (divergence of trajectories).
- $\mathcal{L}_{\text{symp}}$ is the symplectic violation loss (energy drift).
- λ is a penalty coefficient.

This formulation posits that the highest value systems are those that maximize the diversity of "lives" (unique trajectories) while remaining rooted in fundamental physical laws. High strain implies the model is stretching the boundaries of the known world without breaking them.

13.1 Mechanism of Strain

To earn resonance, a peer must perform work that stabilizes the system:

1. The network sends a chaotic latent state x_{chaos} to the peer.
2. The peer finds a trajectory that guides x_{chaos} to a stable attractor x_{stable} within N steps.
3. The peer minimizes the Hamiltonian energy loss: $\mathcal{L}_{\text{Ham}} = H(x_{\text{stable}}) - H(x_{\text{chaos}})$.
4. The "Line" (a governance mechanism) verifies that $\mathcal{L}_{\text{Ham}} < \epsilon$ for a threshold ϵ.
5. Reward: $\mathcal{R} = \frac{1}{\mathcal{L}_{\text{Ham}}}$ (more stabilization = higher reward).

This is **Proof of Resonance**—meritocratic and thermodynamic rather than wasteful or plutocratic.

13.2 Utility of Resonance

Resonance is valuable because it grants:

- **Priority Access**: Spend $\mathcal{R}$ to direct the model's attention (e.g., "Dream a cure for this protein structure").
- **Reality Injection**: Spend $\mathcal{R}$ to upload personal memories into the permanent Holarchy.
- **Compute Time**: Resonance is directly convertible to compute resources on the network.

13.3 Photonic Value Transfer

To eliminate digital wallets and seed phrases (which are vulnerable to phishing), we implement value transfer through **light**:

1. **Transmission**: The sender's device modulates screen pixels at 60-120 Hz with a specific color-frequency pattern (invisible to humans but detectable by sensors).
2. **Reception**: The receiver's AR glasses or camera captures this photonic stream.

3. **Latching**: A neural pattern matcher decodes the signal and "latches" onto the resonance value.

This is **physical layer security**—you cannot intercept a transaction without being physically present to see the light. We call this "line-of-sight value transfer."

Part VI
Conclusion

14 Summary of Contributions

This monograph presents a unified framework for physics-grounded multimodal intelligence. Key contributions include:

1. **Hamiltonian State Space Duality**: Treating latent states as coordinates in a symplectic phase space, enabling energy-preserving long-horizon prediction.
2. **Symplectic Integration**: Rigorous proof that the Leapfrog integrator preserves the symplectic structure, with bounded energy error.
3. **Sparse Mixture of Hamiltonian Experts**: A modular architecture where different experts specialize in different physical regimes, smoothly interpolated through learned routing.
4. **Holographic Arkhon Cells**: A recursive memory hierarchy enabling multi-scale processing while maintaining linear computational complexity.
5. **Birth Media Formalization**: A principled way to initialize world models, capturing the subjectivity inherent in different starting conditions.
6. **Execution Audit Trail**: Complete recording of model lifetimes, enabling post-hoc analysis despite fundamental non-replicability of execution.
7. **Decentralized P2P Execution**: Gossip-based state synchronization with cryptographic immutability of weights.
8. **Akshon Agentic Platform**: Autonomous hyperparameter discovery through sandboxed LLM agents with scientific validation.
9. **Resonance Protocol**: A thermodynamic economic system that rewards stability rather than wasted computation.
10. **Visual Synthesis**: Hamiltonian Flow Matching and 3D Gaussian Splatting for high-fidelity, persistent visual generation.

15 Philosophical Implications

Each execution of Akshon represents a unique "trajectory through experience"—non-replicable, but fully auditable. The Birth Media defines the initial subjective experience, while the Execution Audit Trail preserves the complete history for analysis. This framework enables:

- **Posthumous Analysis**: Understanding what the model "experienced" during its lifetime.
- **Causal Attribution**: Tracing decisions back to specific states and inputs.
- **Comparative Study**: Examining divergence between instances with similar birth media.

The economic framework suggests a new paradigm where value is not based on scarcity (gold) or trust (fiat), but on *the reduction of uncertainty*. Resonance represents the ability to bring order to chaos—a fundamental thermodynamic principle.

16 Future Directions

Open questions and future work include:

1. **Scaling Holarchic Depth**: How deep can the holarchy be before it becomes impractical? What are the theoretical limits?
2. **Formal Verification**: Can we formally verify that learned Hamiltonians satisfy conservation laws beyond numerical observation?
3. **Integration with AR Hardware**: How can Akshon be optimized for real-time spatial computing on AR glasses (e.g., Omi.dev)?
4. **Biological Resolution**: Can the holarchy be scaled to model biological systems at cellular or molecular resolution?
5. **Economic Stability**: What mechanisms prevent hyperinflation or deflation of resonance in the decentralized economy?
6. **Cross-Modal Transfer**: How do insights learned in visual dynamics transfer to audio, language, or other modalities?

17 Availability and Licensing

- **Research License**: The theoretical framework and mathematical derivations presented in this monograph are released under the **Creative Commons Attribution 4.0 International (CC BY 4.0)** license, encouraging academic reproducibility and extension.
- **Source Code License**: The reference implementation of Akshon is dual-licensed:
    1. **Research Use**: Available under the **Akshon Research License (ARL-1.0)**, which permits modification and experimentation for non-commercial scientific purposes.
    2. **Commercial Use**: Requires a commercial license from the entities holding the sovereign cryptographic keys.
- **Safety Clause**: In the event of the primary author's cessation of biological function ("Life Termination Event"), the codebase is automatically released into the public domain under the **Unlicense**, ensuring that the knowledge is preserved for humanity. As implemented in the `dead_man_switch` protocol, access restrictions will be lifted automatically.
- **Artifact Repository**: Research papers, rendered PDFs, and versioned artifacts are hosted on **Zenodo.org** to ensure permanent digital object identifiers (DOIs) and citation stability.

18 Acknowledgments

This research builds upon the foundational work of the Mamba team [3], the JEPA architecture [1], and the long tradition of symplectic integration in computational physics. We thank the contributors to the 3D Gaussian Splatting community [4] for demonstrating the power of differentiable 3D representations.

Special acknowledgment to the principles of classical mechanics—Hamilton, Lagrange, and Poisson—whose mathematical insights from the 19th century continue to illuminate the path toward artificial intelligence in the 21st.

This work serves as a sub-research paper on agentic research and development. We extend our gratitude to all contributors and note that the principles presented herein have been applied and extrapolated to other industries.

A Notation Reference

Symbol	Meaning
$h \in \mathbb{C}^d$	Latent hidden state (position in phase space)
$p \in \mathbb{C}^d$	Conjugate momentum
$H(h, p)$	Hamiltonian energy function
$V_\theta(h)$	Learned potential energy
$T(p)$	Kinetic energy ($\frac{1}{2}\|p\|^2$)
$\mathcal{H}_\ell$	Holographic Akshon Cell at depth ℓ
$\mathcal{B}$	Birth Media initialization
$\mathcal{A}$	Execution Audit Trail
Δt	Integration time step
J	Standard symplectic matrix $\left(\begin{smallmatrix} 0 & I \\ -I & 0 \end{smallmatrix}\right)$
$\mathcal{R}$	Resonance (unit of thermodynamic stability)
ω	Symplectic 2-form ($dh \wedge dp$)
θ	All learnable parameters
ϕ	Routing network parameters

Table 1: Mathematical notation used throughout this monograph.

B Algorithm Pseudocode

B.1 Symplectic Leapfrog Step

Require: Current state (h, p), potential function V, step size Δt
  $p_{\text{half}} \leftarrow p - \frac{\Delta t}{2} \nabla_h V(h)$
  $h_{\text{new}} \leftarrow h + \Delta t \cdot p_{\text{half}}$
  $p_{\text{new}} \leftarrow p_{\text{half}} - \frac{\Delta t}{2} \nabla_h V(h_{\text{new}})$
  return $(h_{\text{new}}, p_{\text{new}})$

B.2 SMoE-HE Routing

Require: Input $\mathbf{x}$, expert potentials $\{V_k\}_{k=1}^{K}$, routing network g_ϕ
  $\boldsymbol{\alpha} \leftarrow g_\phi(\mathbf{x})$ ▷ Softmax over experts
  $V_{\text{eff}}(h) \leftarrow 0$
  for $k = 1$ to K **do**
    $V_{\text{eff}}(h) \leftarrow V_{\text{eff}}(h) + \alpha_k V_k(h)$
  end for
  return $V_{\text{eff}}(h)$

B.3 Holographic Cell Processing

Require: Input signal z_ℓ, sub-cells $\{\mathcal{H}_{\ell-1}^{(j)}\}$, functions f_ℓ, g_ℓ
  local $\leftarrow f_\ell(z_\ell)$
  subsignal $\leftarrow g_\ell(z_\ell)$
  aggregate $\leftarrow 0$
  for $j = 1$ to N_ℓ **do**
    aggregate $\leftarrow$ aggregate $+ \mathcal{H}_{\ell-1}^{(j)}$(subsignal)
  end for

$\hat{z}_\ell \leftarrow \text{local} + \alpha \cdot \frac{\text{aggregate}}{N_\ell}$
return $\hat{z}_\ell$

C Core Interfaces

```python
class HamiltonianSystem(nn.Module):
    """
    Abstract base class for energy-preserving dynamics.
    Enforces symplectic structure via canonical transformations.
    """
    def __init__(self, dim: int):
        super().__init__()
        self.dim = dim

    @abstractmethod
    def potential_energy(self, q: torch.Tensor) -> torch.Tensor:
        """Compute V(q), the potential energy."""
        pass

    def kinetic_energy(self, p: torch.Tensor) -> torch.Tensor:
        """Compute T(p) = 0.5 * ||p||^2."""
        return 0.5 * torch.sum(p**2, dim=-1)

    def forward(self, state: PhaseState, dt: float) -> PhaseState:
        """
        Evolve state using Symplectic Leapfrog integration.
        Guarantees preservation of phase space volume.
        """
        q, p = state.q, state.p

        # Half-step momentum kick
        grad_V = torch.autograd.grad(self.potential_energy(q).sum(), q)[0]
        p_half = p - 0.5 * dt * grad_V

        # Full-step position drift
        q_new = q + dt * p_half

        # Half-step momentum kick (at new position)
        grad_V_new = torch.autograd.grad(self.potential_energy(q_new).sum(),
    q_new)[0]
        p_new = p_half - 0.5 * dt * grad_V_new

        return PhaseState(q_new, p_new)
```

Listing 1: Abstract Base Class for Hamiltonian Dynamics

D Mermaid Diagrams

Version Française

Résumé

Nous présentons **Arkhon**, un cadre complet pour les modèles du monde multimodaux ancrés dans les principes de la mécanique classique. En traitant l'état latent comme un système dynamique évoluant sur une variété symplectique gouvernée par une fonction Hamiltonienne apprise, nous obtenons une préservation de l'énergie et une stabilité temporelle que les modèles autorégressifs traditionnels ne peuvent égaler.

L'architecture emploie des **Modèles d'Espace d'État Sélectifs Mamba-3** améliorés avec un **Mélange Épars d'Experts Hamiltoniens**, permettant la prédiction à long terme grâce à l'intégration **Symplectique Leapfrog** du second ordre. Nous formalisons le protocole d'initialisation **Birth Media** et la **Piste d'Audit d'Exécution** pour une reproductibilité complète, et démontrons l'exécution décentralisée à travers un mécanisme de consensus pair-à-pair avec immutabilité cryptographique.

La synthèse visuelle est réalisée par **Hamiltonian Flow Matching** et **3D Gaussian Splatting** persistants, permettant une inférence sub-50ms sur appareils mobiles tout en maintenant la cohérence spatiotemporelle. De plus, nous introduisons la **Plateforme Agentique Arkhon** pour la découverte autonome d'hyperparamètres dans des environnements sandboxés, et le **Protocole de Résonance** pour récompenser les contributions qui stabilisent le système.

Des preuves mathématiques rigoureuses établissent la préservation de la structure symplectique, les bornes de conservation de l'énergie, et l'impossibilité de réplication exacte de l'exécution dans les systèmes dynamiques stochastiques.

Remerciements

Cette recherche s'appuie sur les travaux fondateurs de l'équipe Mamba, l'architecture JEPA, et la longue tradition de l'intégration symplectique en physique computationnelle. Nous remercions les contributeurs de la communauté 3D Gaussian Splatting pour avoir démontré le pouvoir des représentations 3D différentiables.

Reconnaissance spéciale aux principes de la mécanique classique—Hamilton, Lagrange et Poisson—dont les insights mathématiques du XIXe siècle continuent d'illuminer le chemin vers l'intelligence artificielle au XXIe.

References

[1] Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. *arXiv preprint arXiv:2301.08243*, 2023.

[2] Stephen Boyd, Arpita Ghosh, Balaji Prabhakar, and Devavrat Shah. Randomized gossip algorithms. *IEEE transactions on information theory*, 52(6):2508–2530, 2006.

[3] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

[4] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (ToG)*, 42(4):1–14, 2023.

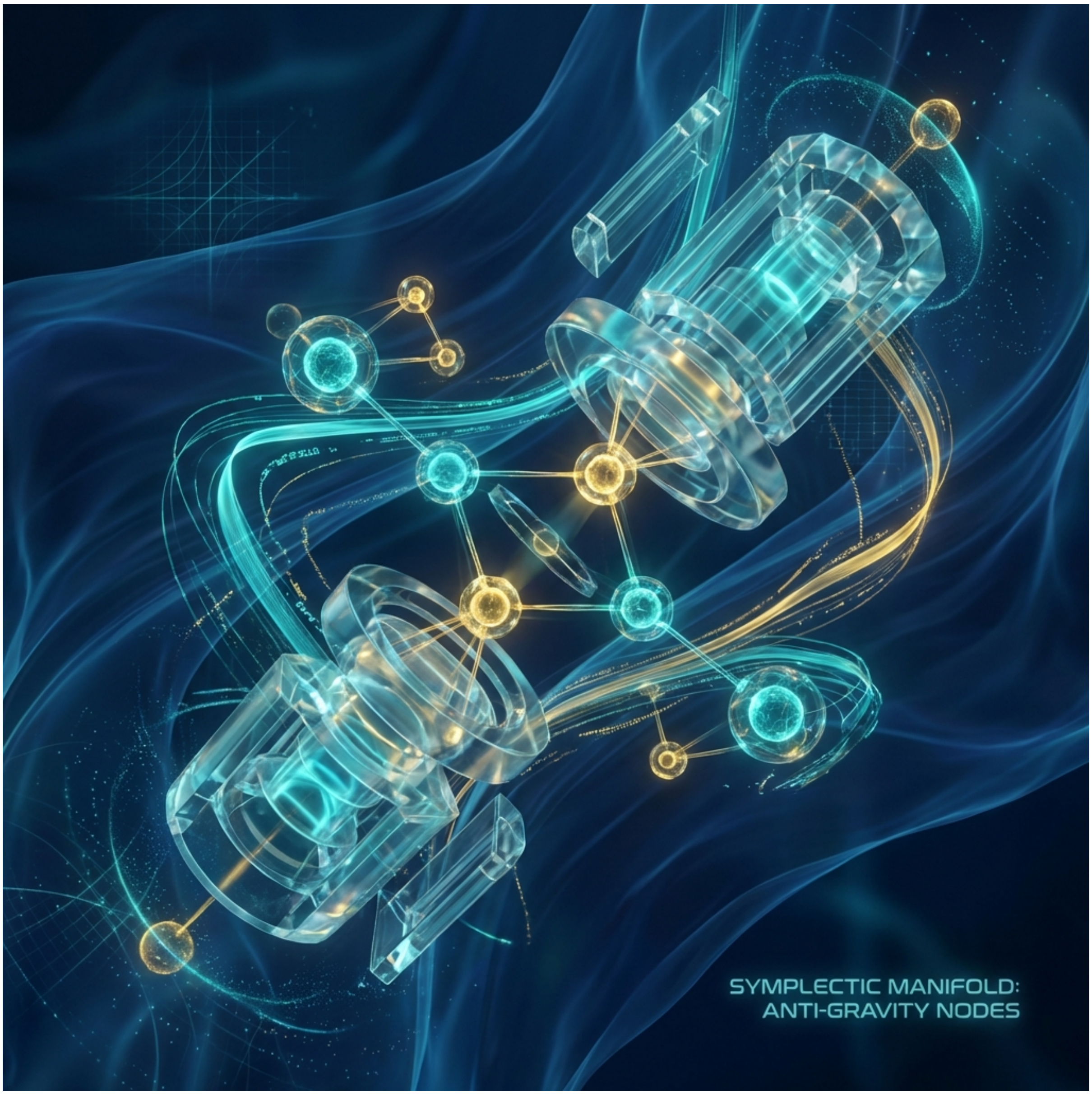


Figure 1: The Akshon Framework visual identity: Anti-gravity nodes in a symplectic manifold.

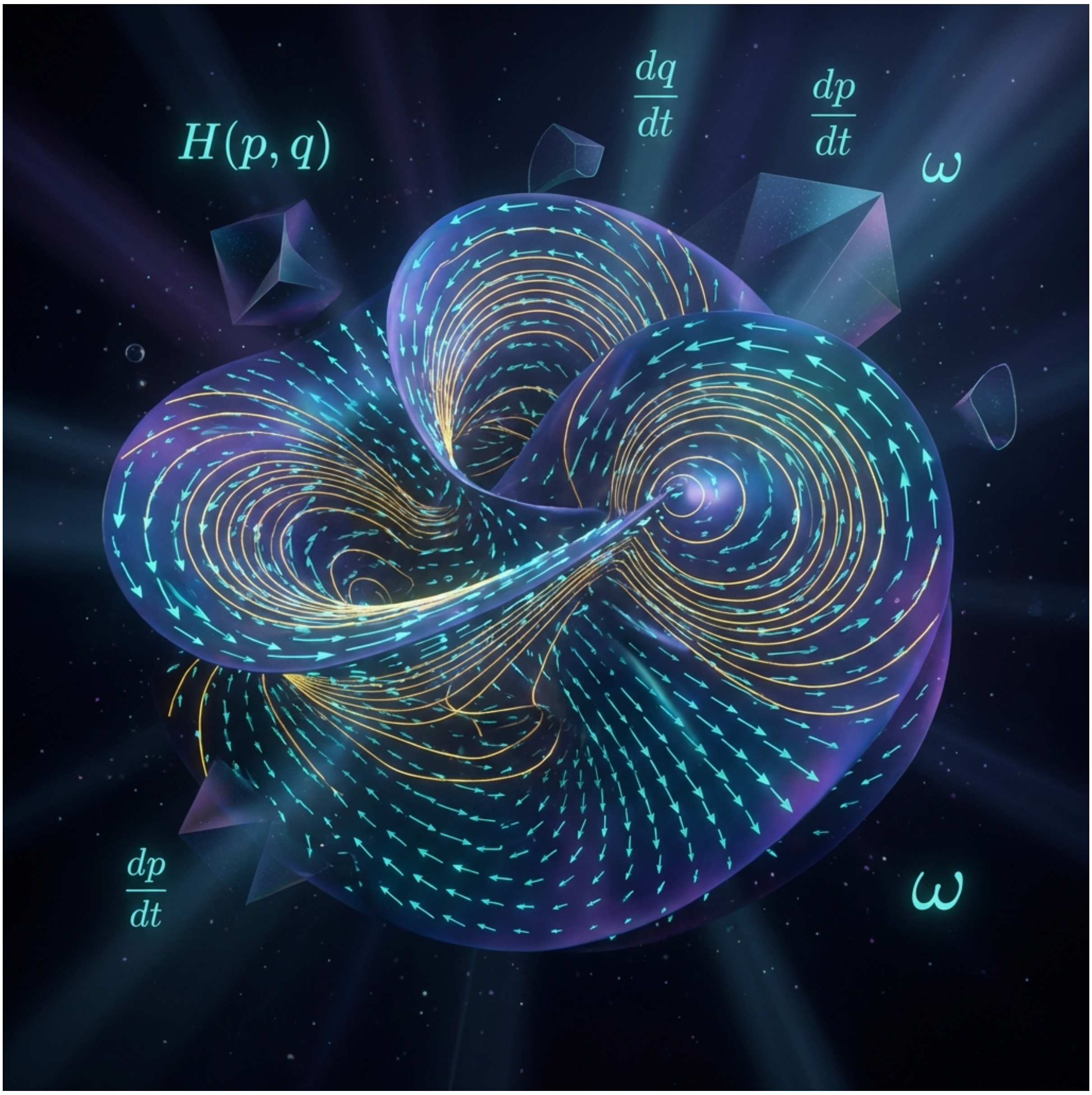


Figure 2: Abstract visualization of Hamiltonian State Space Duality on a symplectic manifold.

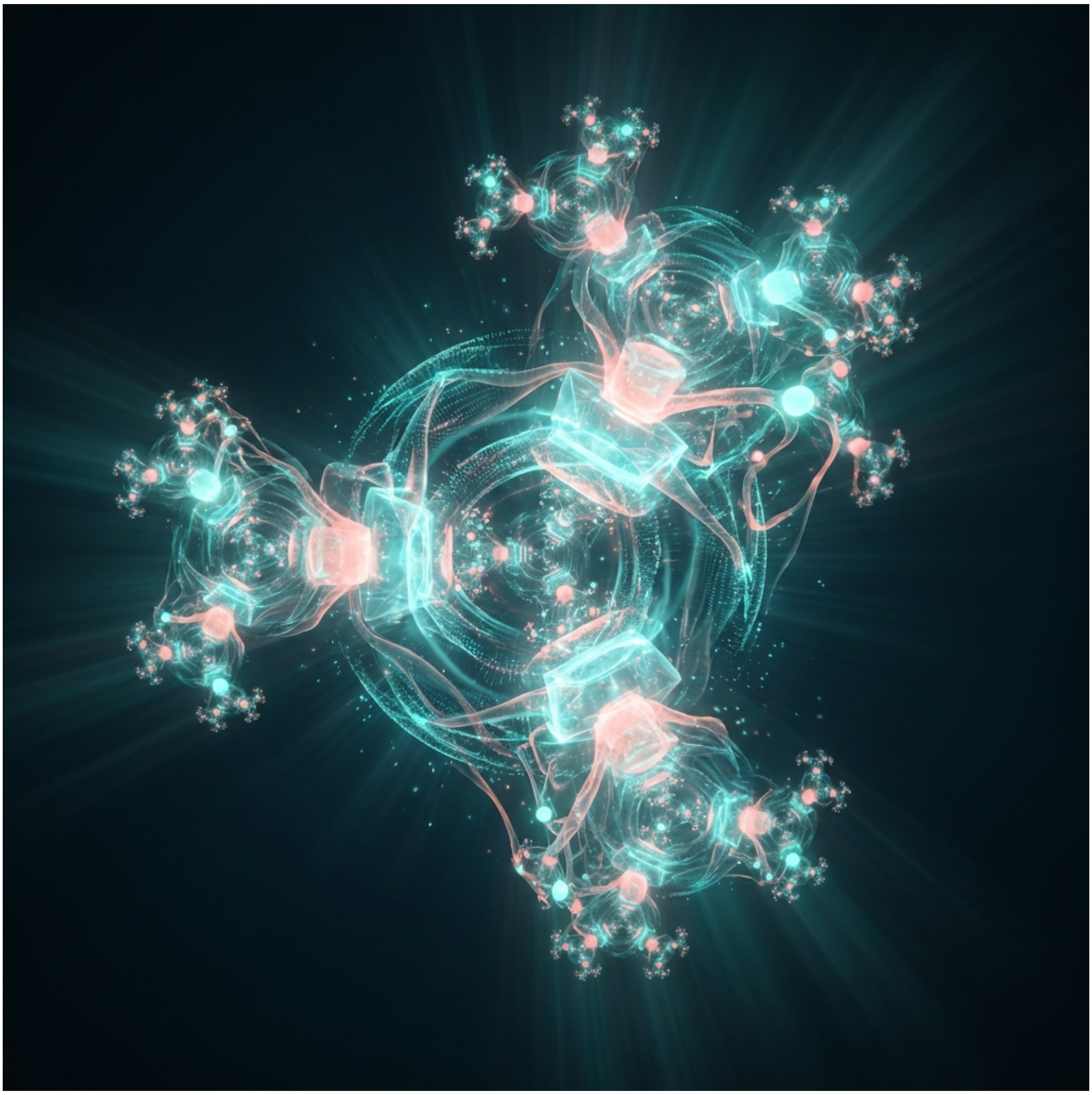
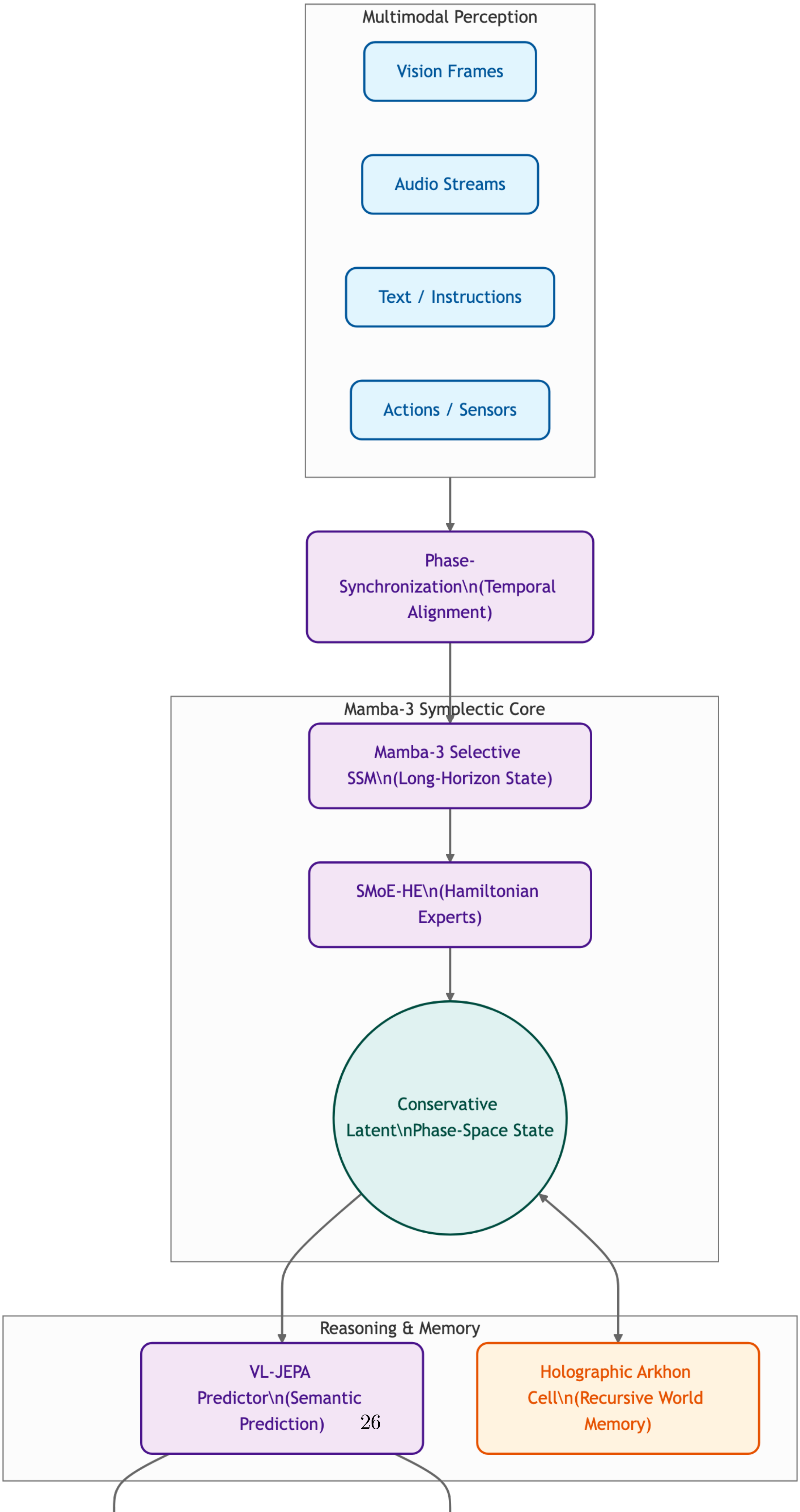


Figure 3: Holographic Akshon Cell: A recursive fractal structure representing multi-scale memory.


Multimodal Perception
Vision Frames
Audio Streams
Text / Instructions
Actions / Sensors
Phase-Synchronization\n(Temporal Alignment)
Mamba-3 Symplectic Core
Mamba-3 Selective SSM\n(Long-Horizon State)
SMoE-HE\n(Hamiltonian Experts)
Conservative Latent\nPhase-Space State
Reasoning & Memory
VL-JEPA Predictor\n(Semantic Prediction)
Holographic Arkhon Cell\n(Recursive World Memory)

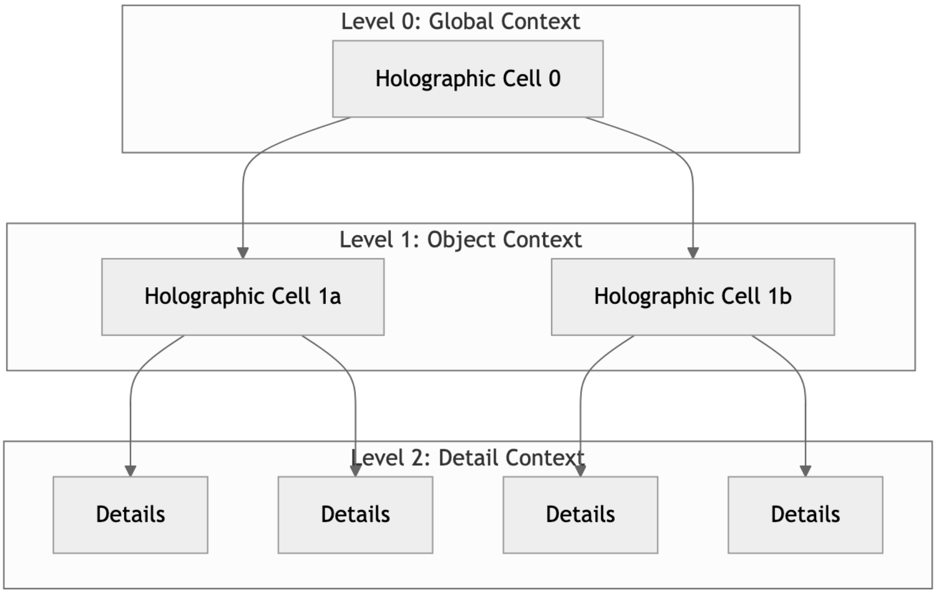


Figure 5: Holarchic Memory Structure: Hierarchical organization of world model cells.

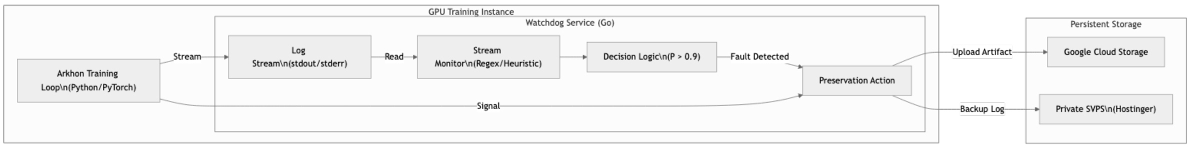


Figure 6: Watchdog Resilience Architecture: Autonomous error detection and state preservation.

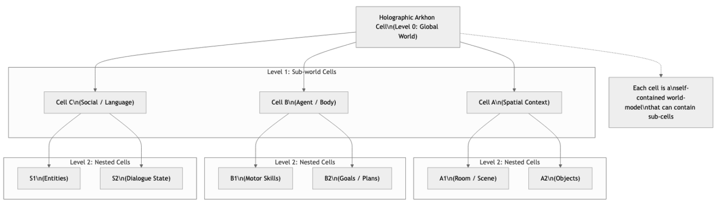


Figure 7: Recursive Holarchic Cells: Each cell is a self-contained world model.

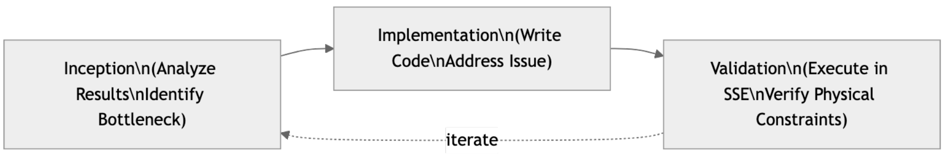


Figure 8: Agentic Research Loop: The cycle of autonomous improvement.

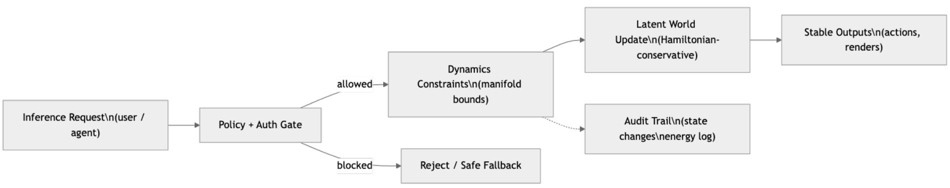


Figure 9: Decentralized Governance: Ensuring stability via proof of resonance.