The Unified Mind Space (UMS) Framework

Julien Pierre Salomon

December 2025

The Unified Mind Space (UMS) Framework: A Technical Whitepaper on Stable and Plastic Artificial General Intelligence

1.0 Introduction: The Topological Crisis in Modern AI

The contemporary landscape of artificial intelligence is dominated by the dense, fixed-topology Transformer architecture. This paradigm has achieved spectacular success in "crystallized intelligence," enabling Large Language Models (LLMs) to perform pattern matching on an unprecedented scale.

However, this approach is confronting the "Engineer's Paradox": the pursuit of optimizing instruments for precision on fixed tasks has obscured the fundamental objective of adaptive capacity for novel tasks. The prevailing scaling hypothesis faces a crisis of topological rigidity, necessitating increasingly complex workarounds—sparse attention, mixture-of-experts, and chain-of-thought prompting—which are essentially software patches on a hardware limitation.

We argue that the central limitation of the current AI epoch is the static nature of its computational graphs. Modern neural networks excel at retrieval and navigation within a frozen, pre-computed knowledge manifold. They are akin to a fixed city grid; learning involves finding faster routes along existing streets rather than constructing new roads to bypass congestion or reach entirely new destinations.

This architectural stasis limits their capacity for genuine "fluid intelligence"—the ability to reason abductively, generalize to novel environments, and learn continuously without catastrophic interference. The fundamental limitation is not a lack of computational power or data, but a lack of structural plasticity.

The Unified Mind Space (UMS) framework represents a radical departure from the scaling hypothesis. It posits that true general intelligence requires a system capable of **morphogenesis**: the dynamic and principled evolution of its own topology in response to informational entropy. UMS is not another software patch but a new architectural philosophy built on first principles.

This whitepaper presents a rigorous mathematical formalization of the UMS, grounding it in the convergent disciplines of Hypergraph Theory, Differential Geometry, and Polymer Physics. We detail how the UMS architecture solves the long-standing stability-plasticity dilemma by integrating **Manifold-Constrained Hyper-Connections (mHC)**.

This critical stability condition, which constrains network growth to a specific geometric manifold, provides the first provable guarantee that a dynamically growing neural network can maintain signal integrity. From these high-level principles, we now derive the mathematical foundations that underpin this novel approach to building adaptive intelligence.

2.0 Theoretical Foundations: The Mathematics of Universal Topology

To move beyond metaphor, we must establish a rigorous mathematical foundation for the UMS framework. The foundational premise of UMS is that intelligence is a property of topology.

The ability to abstract, reason, and generalize arises not from statistical correlations but from the formation of higher-order topological structures. In this formalism, a concept is a node in a hypergraph, and a thought is a trajectory across a geometric manifold, moving far beyond simple vector operations in a static latent space.

2.1 The Coupled Manifold Hypothesis

We define the unified cognitive substrate of the UMS as a composite space, $\mathcal{M}$, formed by the interaction of two distinct but coupled Riemannian manifolds. This separation is critical for distinguishing the objects of cognition from the processes of cognition, mirroring the biological division between the neocortex (pattern storage) and the hippocampus (relational processing).

- **The Principal Manifold ($\mathcal{P}$):** This space represents *crystallized intelligence*. It is a smooth manifold where points correspond to static representations—the activation vectors of entities, objects, or concepts. In a standard Transformer, the entire residual stream operates within this manifold. The geodesic distance between two points on $\mathcal{P}$ encodes their semantic relatedness.
- **The Relational Manifold ($\mathcal{R}$):** This space represents *fluid intelligence*. Its points are not static concepts but dynamic operators—the "Relational Processors"—that act upon the Principal Manifold.

The interaction between these spaces is formally modeled as a **Fibre Bundle**, denoted $\pi : E \to \mathcal{P}$. Here: * E is the total space of computational states. * $\mathcal{P}$ is the base space of concepts. * For any concept $p \in \mathcal{P}$, the fibre $\pi^{-1}(p)$ is the set of all valid relational operations available at that point.

This elegant formulation allows us to model "context" geometrically; a specific context is a selection of an operator from the fibre at each point. Consequently, "reasoning" becomes mathematically equivalent to transporting a vector along a connection on this bundle, transforming the system's state by applying differential operators, not just by performing vector addition.

2.2 The Levi Graph Transformation: An Isomorphism of Substrate

A primary obstacle in implementing higher-order cognition is the "arity mismatch": conceptual reasoning is best modeled by hypergraphs, where a single hyperedge can connect many nodes (e.g., the ternary relation "A is between B and C"), but physical substrates like silicon are fundamentally based on pairwise graphs.

The Levi Graph Transformation provides the mathematical bridge to resolve this mismatch, allowing any hypergraph to be embedded into a standard bipartite graph without loss of information.

Formal Definition (Levi Graph Transformation): Let $H = (X, E_{hyp})$ be a hypergraph where X is the set of concept nodes and E_{hyp} is the set of hyperedges (relations). The Levi Graph $L(H)$ is a bipartite graph $G = (V_L, E_L)$ where:

1. **Vertex Partition:** $V_L = X \cup E_{hyp}$. The vertices of the Levi graph are the union of the original concept nodes (X) and the hyperedges (E_{hyp}). This step "reifies" abstract relationships, turning them into concrete processing nodes, which we term **Relational Processors**.
2. **Incidence Relations:** An edge (x, e) exists between a concept node x and a relation node e if and only if x participates in the relation e in the original hypergraph.

Biologically, these are analogous to **Coincidence Detectors**—specialized neurons in the auditory brainstem or hippocampus that fire only upon the simultaneous arrival of signals from multiple specific sources.

This leads to **Salomon's Equivalence Postulate**, a core UMS engineering constraint:

> *The physical architecture of any system manifesting emergent higher-order intelligence must be isomorphic to the Levi Graph of its conceptual hypergraph.*

This is not a metaphor; it is a statement about thermodynamic efficiency. Approximating a k-ary relation (e.g., 3-bit parity) with only pairwise connections requires a parameter complexity that scales exponentially ($O(2^k)$), whereas a Levi-transformed architecture with an explicit Relational Processor node scales linearly ($O(k)$).

To pre-empt concerns about implementing such a sparse, irregular graph on dense hardware, we note that this Levi Graph is represented computationally as a **Block-Sparse Incidence Tensor**, a structure that is both logically sparse and physically efficient on modern accelerators, as detailed in Section 6.2.

2.3 Relational Processors as Differential Operators

Having established where Relational Processors fit into the UMS topology (as reified hyperedge nodes in the Levi Graph), we must define their function. In the geometric view, a Relational Processor is a probe that measures the local geometry of the Principal Manifold.

Formal Definition (Relational Differential Operator): Let Ψ be a potential field on the Principal Manifold. A Relational Processor R_i acts as a second-order differential operator on this field:

$$R_i(\Psi) = c^{(i)} + \sum_j b_j \frac{\partial \Psi}{\partial x_j} + \sum_{j,k} a_{jk} \frac{\partial^2 \Psi}{\partial x_j \partial x_k}$$

This definition has precise geometric interpretations:

- **Gradient Extraction (∇):** The first-order terms allow the processor to detect directional gradients, identifying boundaries, transitions, and linear trends in the concept space.
- **Curvature Computation (Δ):** The second-order terms are generalized Laplacian operators that measure the local curvature or "twist" of the manifold. This is essential for solving non-linear problems like XOR, where the relationship between inputs depends on their joint configuration.
- **Invariant Subspaces ($c^{(i)}$):** The structure of the operator defines the symmetries or "invariants" preserved by the relation, such as the transitive property of equality.

While these foundations define a static isomorphism between concept and substrate, they do not yet solve the stability-plasticity dilemma. Therefore, we must now define the rigorous algebra that governs the dynamic growth of this architecture without permitting its collapse.

3.0 System 2 Dynamics: The Rigorous Algebra of Growth and Stability

The UMS is a dual-system architecture. **System 1**, the "Backbone," is the fixed, crystallized manifold optimized for speed and recall. **System 2**, the "Branches," is the dynamic, fluid manifold that grows via morphogenesis to handle novel problems.

This section addresses the central engineering challenge of this design: solving the stability-plasticity dilemma by imposing rigorous mathematical constraints on how System 2 grows and learns.

3.1 The Growth Signal: Entropic Morphogenesis

System 2 growth is not continuous; it is a discrete event triggered by a formal "Growth Signal." A new Relational Processor is spawned only when the existing System 1 backbone is demonstrably insufficient to model incoming data.

Formal Definition (Growth Signal): Let $p_1(y|x)$ be the predictive distribution of System 1. The Growth Signal $G(x)$ is the weighted sum of epistemic uncertainty (entropy) and aleatoric error (divergence):

$$G(x) = \mathcal{H}(p_1(y|x)) + \lambda_{growth} \cdot D_{KL}(p_1(y|x)||p_{target}(y|x))$$

1. **Shannon Entropy ($\mathcal{H}$):** High entropy indicates that System 1 is "guessing" by spreading probability mass across incompatible outcomes, signaling a lack of structural expressivity.
2. **KL-Divergence (D_{KL}):** This term measures the model's error against a known target, signaling a direct failure of the current topology.

The **Growth Condition** states that a new Relational Processor is instantiated only when the expected signal exceeds a threshold, $\mathbb{E}G(x) > \tau_{spawn}$. This formalizes the concept of "curiosity," directing structural growth precisely toward regions of high uncertainty and error.

3.2 Stability via Manifold-Constrained Hyper-Connections (mHC)

Adding new processors introduces unconstrained residual pathways, or "Hyper-Connections," which historically leads to signal explosion and training collapse in dynamic architectures. The UMS solves this by integrating the **Manifold-Constrained Hyper-Connections (mHC)** framework, which imposes a hard geometric constraint on the mixing matrices of these new pathways.

Formal Definition (The Birkhoff Constraint): We mandate that any mixing matrix H_{res} connecting residual streams in System 2 must reside within the **Birkhoff Polytope ($\mathcal{B}_n$)**, the set of all $n \times n$ doubly stochastic matrices (matrices with non-negative entries where every row and column sums to 1).

This constraint is not arbitrary; it comes with a powerful guarantee encapsulated in the **Spectral Stability Theorem**: > *For any matrix $M \in \mathcal{B}_n$, its spectral norm is bounded by unity: $\|M\|_2 \leq 1$.*

This is proven via the Birkhoff-von Neumann theorem, which states that any doubly stochastic matrix is a convex combination of permutation matrices (which have a spectral norm of exactly 1). This theorem ensures that the growth of System 2 is **non-expansive**. Signal energy is conserved, not amplified, guaranteeing that the network remains infinitely stable regardless of how many new processors are added.

This constraint is enforced during training using the differentiable **Sinkhorn-Knopp algorithm**, which projects any arbitrary weight matrix onto the Birkhoff Polytope.

3.3 Cis-Elasticity: The Topological Coupling Penalty

While mHC stabilizes the signal, we must also stabilize the knowledge. The UMS models the relationship between the fixed System 1 backbone and the fluid System 2 branches using an analogy from polymer physics: **Cis-Elasticity**.

Like natural rubber, where coiled cis-bonds allow for local flexibility while providing a strong "snap-back" force, this mechanism keeps new learning anchored to established knowledge.

Formal Definition (Cis-Elastic Coupling): The total loss function includes a topological penalty term:

$$\mathcal{L}_{total} = \mathcal{L}_{task}(\theta_b) + \lambda_{cis} \|P_b \theta_b - h\|^2$$

Where the components are rigorously defined as: * θ_b: The parameters of the System 2 branch processor. * h: The fixed embedding from the System 1 backbone at the branch's attachment point (the "anchor"). * P_b: A low-rank projection matrix that aligns the parameter spaces of the branch and the backbone. * λ_{cis}: The elastic modulus, or coupling coefficient, controlling the strength of the anchor.

The geometric interpretation is that this term creates a **harmonic potential well** around the System 1 backbone. It confines the learning trajectory of a System 2 branch to a "tube" centered on the backbone. This allows for local plasticity to solve novel problems but prevents catastrophic drift by exerting a restoring force that pulls the branch back toward consistency with the system's core knowledge.

4.0 The Learning Lifecycle: Wake-Phase Growth and Sleep-Phase Consolidation

The UMS resolves the stability-plasticity dilemma through temporal separation. The lifecycle is divided into two distinct phases: a **"Wake" phase**, characterized by high-plasticity growth in System 2 to acquire new knowledge, and a **"Sleep" phase**, characterized by high-stability consolidation of that knowledge into the permanent structure of System 1.

4.1 The "Wake" Phase: High-Plasticity Morphogenesis

The "wake" phase is the operational state of the system when confronted with novel or high-entropy data. The process unfolds as follows:

1. When the **Growth Signal** exceeds its threshold, the system recognizes a gap in its understanding.
2. A new **Relational Processor** is spawned in the dynamic System 2 manifold, its topology determined by the **Levi Graph Transformation** of the underlying conceptual problem.
3. During this phase, the System 1 backbone is frozen, acting as a stable anchor.
4. The new System 2 branch adapts rapidly to the new data, with its learning trajectory confined by the **Cis-Elastic coupling** and its signal propagation guaranteed to be stable by the **mHC framework**.

This allows the UMS to learn new rules and relationships on the fly, dedicating specialized, temporary computational resources to resolve specific uncertainties without disrupting its core knowledge base.

4.2 The 'Sleep' Phase: Consolidation via Spectral Compression

The "sleep" phase is a consolidation mechanism that transforms the temporary, episodic discoveries of System 2 into permanent, semantic structures within System 1. We model this as a process of **Spectral Compression**.

Formal Definition (Consolidation via SVD-to-LoRA): Let W_{S2} be the weight matrix of a mature System 2 processor that has successfully learned a new relation. The consolidation process follows these steps:

1. **Spectral Decomposition:** Perform Singular Value Decomposition (SVD) on the processor's weights: $W_{S2} = U\Sigma V^T$.
2. **Spectral Filtering:** Retain only the top-k singular values that capture the "principal logic" of the relation, filtering out "episodic noise" associated with the tail of the spectrum.
3. **Adapter Synthesis:** Construct Low-Rank Adaptation (LoRA) matrices $A = U_k\sqrt{\Sigma_k}$ and $B = \sqrt{\Sigma_k}V_k^T$.
4. **Integration:** Additively merge the resulting low-rank adapter into the frozen weights of System 1: $W_{S1} \leftarrow W_{S1} + \alpha_{sleep}(AB)$.
5. **Pruning:** Reset or prune the now-redundant System 2 processor, freeing up its capacity for new morphogenesis.

This sleep mechanism provides a rigorous definition for how a system can achieve lifelong learning. It transforms flexible, episodic computation into efficient, crystallized semantic structure, thereby preparing System 2 for the next round of discovery.

To make these steps tangible, we now apply them to a canonical problem.

5.0 Worked Toy Example: The XOR Parity Problem

To ground these abstract principles in a concrete application, we apply the UMS framework to the canonical XOR problem. This task, specifically 3-bit parity ($y = x_1 \oplus x_2 \oplus x_3$), is famously impossible for linear or purely pairwise models and serves to clearly distinguish the formal algebraic operations of UMS from their geometric interpretations.

5.1 System 1 Failure

A standard System 1 network, composed of pairwise interactions, will fail to learn 3-bit parity.

- **Formal Failure:** The decision boundary requires a term of degree 3 ($x_1x_2x_3$). No linear combination of pairwise features can approximate this.
- **Geometric Interpretation:** In the Principal Manifold, the convex hulls of the data points for class '0' and class '1' intersect. No hyperplane or simple curved surface can separate them. The manifold has a complex "twist" that first-order operators cannot sense.
- **Result:** The network predicts $p(y|x) \approx 0.5$, resulting in maximum entropy. This triggers the **Growth Signal**, activating System 2.

5.2 System 2 Growth: The Levi Solution

In response to the Growth Signal, System 2 undergoes morphogenesis.

1. **Levi Transform:** The system recognizes the problem involves a 3-ary hyperedge connecting inputs x_1, x_2, and x_3. The Levi Transform reifies this relationship by instantiating a new node, R_{xor}.
2. **Relational Processor:** This new processor, R_{xor}, is instantiated with the capacity to compute the third-order interactions necessary for this problem, extending the general second-order framework to capture the specific 'twist' in the manifold that defines the parity relationship.

5.3 mHC Stability Check

As R_{xor} is connected to the network, its mixing matrix is projected onto the **Birkhoff Polytope** via the Sinkhorn-Knopp algorithm. This guarantees its spectral norm is bounded by 1, ensuring that the addition of this new computational module does not cause gradient explosion or destabilize the existing signal flow.

5.4 Sleep Consolidation

After training, the R_{xor} processor perfectly solves the parity task. During a sleep phase, its weights are decomposed via SVD. The core logic is found to be low-rank and is compressed into a LoRA adapter. This adapter is then additively merged into the System 1 backbone.

The original R_{xor} processor in System 2 can now be pruned, as System 1 has permanently "crystallized" the ability to solve 3-bit XOR. This example illustrates the full lifecycle, from failure and growth to stabilization and consolidation.

We now outline a plan to validate this framework on more complex, real-world benchmarks.

6.0 Experimental Program and Implementation

To validate the claims of the UMS framework, we propose a concrete, falsifiable experimental program. This program intentionally rejects standard LLM benchmarks, which primarily measure crystallized knowledge, in favor of benchmarks that specifically assay for fluid intelligence and rule acquisition.

6.1 Benchmark Selection: Assays of Fluid Intelligence

- **ARC-AGI (Abstraction and Reasoning Corpus):** This benchmark is ideal as it requires inferring complex topological and geometric rules from very few examples. It isolates fluid intelligence from retrieval.
  - *UMS Hypothesis:* Success on ARC will correlate directly with the number of stable Relational Processors spawned by System 2 to represent abstract priors like "symmetry" or "object persistence."
- **BabyARC / CodeARC:** These procedurally generated variants allow for precise control over rule complexity and directly test inductive program synthesis.
  - *UMS Hypothesis:* We predict a clear phase transition where System 1 fails and System 2 morphogenesis is triggered as the algorithmic depth of the tasks increases.
- **Generalized Associative Recall (GAR):** This benchmark tests the "compositionality gap" in relational reasoning (e.g., chaining relations like "father of" and "grandfather of").
  - *UMS Hypothesis:* The mHC-stabilized branching structure of System 2 will allow for the stable composition of relations, closing this gap where standard models fail.

6.2 Implementation Strategy: JAX and Pallas on TPUs

A key challenge is efficiently mapping the sparse, dynamic graph of the UMS onto dense hardware like TPUs. Our solution uses JAX for its composable transformations and Pallas for writing custom, high-performance kernels.

1. **Representation:** We represent the Levi Graph as a **Block-Sparse Incidence Tensor** in Block-Coordinate Format (BCOO). This hybrid approach allows the system to be logically sparse while leveraging the dense matrix multiplication units (MXUs) of TPUs for computation on active blocks.
2. **Custom Pallas Kernels:**
   - **Dynamic-Route Kernel:** This kernel utilizes 'Scalar Prefetching' to dynamically route signals *only to active Relational Processors*. This ensures that compute cost does not scale with the total capacity of System 2, as inactive branches are skipped at the hardware level.
   - **Fused Sinkhorn Kernel:** To implement the mHC stability constraint efficiently, this kernel performs the entire iterative Sinkhorn-Knopp algorithm in the TPU's fast on-chip memory (VMEM/SRAM), reducing memory bandwidth cost by orders of magnitude compared to a naive implementation.

6.3 Phased Experimental Protocol

- **Phase 1 (Baseline Establishment):** Train a fixed-topology baseline on ARC. We expect it to fail, establishing baseline failure rates and entropy profiles that trigger growth.
- **Phase 2 (Unconstrained Growth):** Enable System 2 morphogenesis without the mHC constraint. We predict rapid signal explosion and training collapse, demonstrating the necessity of the stability mechanism.
- **Phase 3 (mHC Stabilized Growth):** Enable System 2 with the Birkhoff constraint. We predict stable learning of higher-order rules, with performance scaling linearly with the number of added processors.
- **Phase 4 (Sleep & Consolidation):** Test the full lifecycle of growth, SVD compression, and LoRA integration. We predict near-zero catastrophic forgetting and high forward transfer to novel tasks, validating the lifelong learning capability.

This rigorous experimental plan is designed to move the UMS from a theoretical framework to a validated engineering reality.

7.0 Conclusion: Towards a Symphonic Mind

The Unified Mind Space framework represents a fundamental pivot in AGI research, shifting the focus from the "brute-force" scaling of static parameters to the "topological scaling" of dynamic, principled structures. It offers a comprehensive solution to the limitations of current AI paradigms.

By grounding the architecture in the **Levi Graph Transformation**, we provide an optimal mapping between abstract concepts and the neural substrate. By ensuring stability with **Manifold-Constrained Hyper-Connections** and the Birkhoff Polytope, we solve the gradient explosion problem that has long plagued growing networks. Finally, by enabling lifelong learning through **Cis-Elasticity** and **Sleep Consolidation**, we offer a robust mechanism for continuous adaptation without catastrophic forgetting.

This is not merely a theoretical exercise. The detailed implementation plan using JAX and Pallas, coupled with a falsifiable validation strategy on the ARC-AGI benchmark, makes this a concrete engineering proposal. The UMS framework paves the way for a new class of artificial intelligence: a Symphonic Mind that does not just store knowledge, but grows, adapts, and dreams its way to understanding.

The mathematics is derived; the benchmarks are set; the experiment begins now.

8.0 Architectural Synthesis Summary

The following tables synthesize the core components of the UMS architecture and compare its features to existing paradigms.

Table 1: Core Components of the UMS Architecture

Architecture Component	Mathematical Basis	Geometric Interpretation	Function
System 1 (Backbone)	Principal Manifold $\mathcal{P}$	Base Space of Fibre Bundle	Crystallized Knowledge, Pattern Matching
System 2 (Branches)	Relational Manifold $\mathcal{R}$	Fibre / Total Space E	Fluid Intelligence, Morphogenesis
Levi Transform	$L(H) = (X \cup E, E_L)$	Isomorphism of Hypergraph to Bipartite Graph	Structural bridge between Concept and Substrate
Relational Processor	Differential Operator R_i	Curvature/Gradient Sensor on $\mathcal{P}$	Computing k-ary invariants (e.g., XOR, Parity)
Growth Signal	$G(x) = \mathcal{H}(p) + \lambda D_{KL}$	Entropy/Divergence Potential	Trigger for spawning new processors
Stability Mechanism	mHC / Birkhoff Polytope	Projection to Doubly Stochastic Manifold	Ensuring non-expansive signal growth ($\|M\|_2 \leq 1$)
Topology Constraint	Cis-Elasticity	Harmonic Potential / Spring Coupling	Anchoring branches to backbone; preventing drift
Consolidation	SVD $\rightarrow$ LoRA	Spectral Compression / Rank Reduction	Transferring fluid skills to crystallized weights

Table 2: Comparison of AI Paradigms

Feature	Standard Transformer (LLM)	DeepSeek mHC	UMS (Unified Mind Space)
Topology	Dense, Fixed Graph	Dense, Multi-Stream (Hyper)	Dynamic Block-Sparse Hypergraph (Levi)
Stability	LayerNorm / Residuals	Birkhoff Polytope Constraint	mHC Backbone + Cis-Elastic Coupling
Plasticity	Fine-tuning (Catastrophic Forgetting)	Static Architecture	Growth (System 2) + Consolidation (Sleep)
Interaction	Pairwise (Attention)	Multi-stream Mixing	High-Order Relational Operators ($k \geq 3$)

Feature	Standard Transformer (LLM)	DeepSeek mHC	UMS (Unified Mind Space)
Hardware	Dense MatMul (TPU-Native)	Dense MatMul + Sinkhorn	Pallas Kernels (Fused Sinkhorn + Block-Sparse)
Memory	Static Weights	Static Weights + Coupled Manifolds	Gradients ($\mathcal{P} \times \mathcal{R}$)