

Mathematics as the Nervous System of AI: A Unified Field Operator Framework for Distributed Cognition

Philip Siniscalchi

2026-08-27

Contents

Abstract	2
Reproducibility and Artifacts	3
1. Introduction	4
2. What We Built: Implementation and Results	5
Scope and Contributions	5
Falsifiability	6
2.1 Empirical Validation	7
Implementation Overview	7
Retrieval Paths	7
Benchmark Evaluation	8
Key Measured Properties	9
Complexity Analysis	9
3. How It Works: Mathematical Framework	9
3.1 The 4D Field Lattice	10
3.2 Why Waves? The Physics of Superposition	11
3.3 The Damped Wave Equation	11
3.4 Encoder and Decoder Operators	12
3.5 Operator Constraints and Guardrails	14
3.6 Energy Conservation: Parseval's Theorem	16
4. Memory and Retrieval Mechanisms	17
4.1 The Matched Filter: Optimal Retrieval	17
4.2 Selective Attention via Resonance	18
5. Positioning: Comparison to Existing Work	20
6. Design Principles: Structural Metaphors	25
6.1 Physics Analogy: Hilbert Spaces and Hamiltonian Dynamics	25
6.2 Chemistry Analogy: Constraints and Homeostatic Regulation	26
6.3 Biology Analogy: Goal Attractors and Modular Cognition	27

7. Theoretical Extensions (Research Roadmap)	29
7.1 Persuadability and Landscape Dynamics	30
7.2 Formal Bounds on Persuadability	31
7.3 Temporal Dynamics: Decay and Memory Consolidation	31
7.4 Multi-level Persuasion	32
7.5 Applications in AI Architectures	34
8. Open Questions and Future Work	36
8.1 Scalability of the Field	36
8.2 Learning the Operators	36
8.3 Stability and Convergence	37
8.4 Interference and Forgetting	37
8.5 Observation and Interpretability	37
8.6 Computational Cost	38
8.7 Safety and Unintended Attractors	38
8.8 Comparison with Deep Learning Baselines	39
8.9 Biological Plausibility and Inspiration	39
9. Conclusion	39
9.1 Author Contributions and AI-Assisted Research	41
Acknowledgments	42
10. References	42
Appendix A: Notation Summary	44
Appendix B: Theorem and Lemma Index	44

Author: Philip Siniscalchi ORCID: 0009-0007-3820-0509 Affiliation: SmartHaus Group, Independent Research Contact: phil@smarthausgroup.com Date: 2026-08-27 Version: 9.0

License: This work is licensed under CC-BY-4.0. Supersedes: v8 (2026-05-24). v9 (1) repairs citation cross-references throughout so that every in-text marker matches the reference it describes and every listed reference is cited, (2) aligns empirical claims with current evidence status (comparative benchmark results are deferred to a companion evaluation document), (3) resolves internal inconsistencies (energy-conservation scope, a residual enumerative invariant count, stale section cross-references), (4) states the capacity–dimension tradeoff underlying the $O(D \log D)$ claims and marks attractor dynamics as the theoretical layer, and (5) adds explicit assumptions to Theorems 1, 3, 5, and 6 and labels restatements of classical results as such. v8 (2026-05-24) added Plate 1995/2003 and Kleyko 2022 to the prior-art comparison and shifted invariant/notebook counts from enumerative to categorical framing, with live counts maintained in the RFS and NME repository invariant ledgers.

Abstract

Modern artificial intelligence systems—convolutional neural networks, transformers, graph neural networks—achieve remarkable performance on specialized tasks, yet the field lacks a principled mechanism for integrating heterogeneous modules into unified cognitive architectures. Current

approaches rely on ad-hoc pipelines, API calls, or monolithic models that combine capabilities through brute-force scaling rather than structured composition. This fragmentation limits emergent reasoning across modalities and prevents the kind of fluid information sharing that characterizes biological cognition. We propose a field-theoretic framework that positions mathematics itself as the integrating substrate—a “nervous system” for artificial intelligence. In this framework, individual neural networks project their internal states into a shared Hilbert space $\mathcal{H}$ using linear encoding operators E_i , and retrieval occurs through adjoint (matched-filter) projections E_i^H that optimally detect stored patterns in noise. The framework draws on three scientific domains for its design principles: physics provides energy conservation guarantees through Parseval’s theorem and unitary transforms; chemistry informs the guardrails that manage interference between superposed signals; and biology motivates attractor-based goal representation and global workspace dynamics for distributed awareness. The Resonant Field Substrate (RFS) implements this theoretical framework as a working system. RFS encodes information in a four-dimensional complex field tensor $\Psi(x, y, z, t)$ where spatial dimensions enable interference-based semantic encoding and the temporal dimension supports recency weighting and memory decay. The system provides multiple retrieval paths from a single unified storage: vector and interference paths discover semantically similar content through cosine similarity and matched-filter resonance, a proactive discovery path surfaces structural patterns without explicit queries, and an exact channel with authenticated encryption (AEAD) guarantees bit-perfect recall with cryptographic integrity verification. A comprehensive suite of mathematical invariants—covering energy preservation, bounded interference, resonance quality, capacity margins, and operator conductivity—is validated in continuous integration across the verification notebook set. Full definitions and live counts are maintained in the repository invariant ledger. RFS is designed to match production vector-database retrieval quality while providing capabilities they cannot: unified associative and exact recall without separate systems, built-in explainability through resonance quality (Q) and interference (η) metrics computed per query, and a mathematically grounded substrate extensible to multimodal cognitive architectures. Benchmark methodology and comparative results are reported in a companion evaluation document. The framework is falsifiable by construction—every implemented guarantee maps to a measured invariant, and any violation would invalidate the framework for that configuration. Keywords: field-theoretic AI, Hilbert space operators, associative memory, resonant field storage, energy conservation, attractor dynamics, global workspace theory, wave-based computation

Reproducibility and Artifacts

All implementation-level claims in this paper map to machine-checked invariants in a working implementation. A comprehensive suite of mathematical invariants—covering energy preservation, bounded interference, resonance quality, capacity margins, and operator conductivity—is validated in continuous integration. Full definitions and live counts are tracked in the repository invariant ledger. Benchmark evaluations follow established protocols[1]. All experiments use fixed random seeds for reproducibility. Benchmark methodology, hardware specifications, and comparative results are reported in a companion evaluation document (see Section 2, Benchmark Evaluation). The invariant ledgers are currently maintained in the implementation repositories; artifact publication is planned to accompany the companion evaluation document.

1. Introduction

The landscape of artificial intelligence has fragmented into increasingly specialized components. Vision systems achieve superhuman performance on image classification; large language models generate coherent text across diverse domains; graph neural networks capture relational structure in complex datasets; reinforcement learning agents master games and robotic control. Yet these remarkable capabilities remain siloed. A vision model cannot natively share its understanding of a scene with a language model reasoning about descriptions of that scene. A planning system cannot directly query an episodic memory system for relevant past experiences. Integration happens through ad-hoc pipelines, manual API design, or brute-force scaling that combines capabilities in a single monolithic model[2]. This fragmentation stands in stark contrast to biological cognition. The human brain—despite comprising dozens of specialized regions for vision, language, motor control, emotion, and memory—operates as a unified system. Information flows fluidly between regions; a word can evoke a visual memory, which triggers an emotional response, which influences motor behavior, all within milliseconds. Cognitive neuroscience has proposed various accounts of this integration, from Baars’ Global Workspace Theory[3], which posits a central “blackboard” where information becomes globally accessible, to Tononi’s Integrated Information Theory[4], which quantifies the degree to which a system’s components work as an irreducible whole. Levin’s research on bioelectric networks[5][6][7] demonstrates that even collections of cells—far simpler than neural tissue—can achieve coordinated, goal-directed behavior through shared electrical signaling, suggesting that integration may be more fundamental than neural architecture. Existing approaches to AI integration fall short in characteristic ways. Vector databases (FAISS[8], Pinecone, Milvus) provide efficient similarity search but lack mechanisms for interference-based pattern completion, built-in integrity verification, or explainability beyond similarity scores. Hopfield networks[9] and their modern variants[10] provide attractor-based content-addressable memory but struggle to interface with heterogeneous neural architectures. Global Workspace implementations in AI[11] typically require pairwise translation networks between modalities, scaling poorly as the number of components increases. The Conscious Turing Machine[12] provides a formal architecture for workspace-based integration but remains largely conceptual. This paper proposes a different approach: a mathematical substrate that serves as a “nervous system” for artificial intelligence, providing structured communication, shared memory, and coordinated dynamics for heterogeneous components. The substrate is a Hilbert space $\mathcal{H}$ into which individual modules project their states using linear encoding operators E_i . Information storage occurs through superposition—multiple patterns coexist as a single field state $\Psi = \sum_i E_i(w_i)$ —and retrieval occurs through matched-filter correlation using the adjoint operators E_i^H . The framework draws design principles from physics (energy conservation via Parseval’s theorem, wave interference and resonance), chemistry (guardrails for managing interference, homeostatic regulation), and biology (attractor-based goals, global workspace dynamics, top-down modulation). The Resonant Field Substrate (RFS) implements this framework as a working system, providing empirical validation for the theoretical claims. RFS provides unified associative and exact recall from a single storage substrate, built-in explainability through resonance and interference metrics, and a mathematically grounded foundation extensible to multimodal cognitive architectures. A comprehensive suite of mathematical invariants—covering energy preservation, bounded interference, resonance quality, capacity margins, and operator conductivity—is validated in continuous integration. Full definitions and live counts are tracked in the repository invariant ledger. This paper adopts an implementation-first structure, presenting the system and its measured invariants before the underlying mathematics. Section 2 describes what we built and measured, presenting the RFS implementation and its evaluation framework. Section 3 develops the mathematical framework, introducing the operator calculus, Hilbert space formalism, and en-

ergy conservation principles. Section 4 details the memory and retrieval mechanisms, including matched-filter decoding and attractor dynamics. Section 5 positions our work against existing approaches from cognitive science and machine learning. Section 6 presents the design principles drawn from physics, chemistry, and biology that motivated our architectural choices. Section 7 describes theoretical extensions representing a roadmap for future work. Sections 8 and 9 address open questions and conclusions. This structure allows readers to immediately assess whether the framework delivers practical value before investing time in the mathematical details.

2. What We Built: Implementation and Results

This section presents what is implemented and measured—the Resonant Field Substrate (RFS). The theoretical framework follows in subsequent sections. This paper presents a mathematical framework accompanied by one working implementation: the Resonant Field Substrate (RFS). The RFS implementation includes a complete 4D field storage system with wave-based encoding, pseudo-random phase masks for document separation, matched-filter retrieval via adjoint operators, configurable temporal decay, projector-based band separation between associative and exact channels, and AEAD-authenticated exact recall for integrity-critical applications. The implementation is validated through a comprehensive suite of mathematical invariants—covering energy preservation, bounded interference, resonance quality, capacity margins, and operator conductivity—tested in continuous integration across the verification notebook set, each containing executable proofs that verify specific theoretical guarantees. Full definitions and live counts are maintained in the repository invariant ledger. The mathematical foundation underlying the implementation is fully documented in supplementary materials: the Operators Calculus specifies all mathematical operators with their domains, codomains, and computational complexity; the Lemmas Appendix provides formal proofs for energy conservation, interference bounds, and convergence properties; and the invariant specifications in YAML format define testable predicates for each guarantee. Beyond what is implemented, Section 7 describes a theoretical roadmap for extensions including multimodal cortices that interface heterogeneous neural architectures, inter-module communication protocols that leverage the shared field, and dynamic persuadability mechanisms that enable intentional shaping of the attractor landscape. These remain theoretical proposals rather than implemented capabilities. We began with memory because it is the foundation upon which everything else must build. The decision to include exact recall alongside associative retrieval requires clarification. We do not claim that biological cognition requires exact recall—human memory is reconstructive, and the reconsolidation literature demonstrates that memories are modified each time they are retrieved[13][14][49]. However, for engineered AI systems operating in contexts where fidelity and provenance matter—tool outputs, audit trails, legal citations, compliance documentation—a substrate should provide both reliable associative retrieval and an exact, integrity-verified artifact channel. RFS implements this architecture: associative resonance (vector and interference paths) discovers semantically related content through cosine similarity and matched-filter correlation, a proactive discovery path surfaces structural patterns without explicit queries, and a separate byte channel with authenticated encryption (AEAD) guarantees bit-perfect reconstruction with cryptographic integrity verification.

Scope and Contributions

This paper advances four central claims, each grounded in established mathematics and validated through empirical measurement. First, we claim that a field-theoretic framework based on wave physics, signal processing, and energy conservation can serve as a unifying substrate for AI memory and cognition—not as metaphor but as operational mathematics. Second, this substrate

provides formally defined, measurable properties that characterize system behavior: resonance quality Q quantifies signal clarity during retrieval, interference ratio η tracks destructive overlap between stored patterns, conductivity κ measures projector transmission efficiency, and capacity bounds ensure the system operates within stable regimes. Third, the framework is grounded in established mathematics from signal processing (matched filters, Parseval’s theorem), functional analysis (Hilbert spaces, projection operators), and dynamical systems (attractor networks, energy landscapes)—we introduce no novel mathematical machinery, only novel application. Fourth, the RFS implementation validates the memory foundation empirically, demonstrating retrieval quality matching vector database baselines while providing capabilities they lack.

Several claims lie explicitly outside our scope. We do not claim to have created consciousness, sentience, or phenomenal experience—such claims would require philosophical commitments and empirical evidence far beyond what this work provides. The biological analogies that inform our design—bioelectric networks, neural fields, morphogenetic gradients—serve as engineering inspiration, not identity claims; RFS is a mathematical and computational artifact, not a biological system. Finally, while Section 7 describes theoretical extensions for multimodal cognitive architectures, these represent a research roadmap rather than implemented capabilities. The distinction between what is implemented and what is proposed is maintained throughout.

Falsifiability

The framework’s empirical foundation rests on falsifiability: every implemented guarantee maps to a measurable quantity, and any violation would invalidate the framework for that configuration. Energy preservation is testable per write operation through the relation $\|E(w)\|_2^2 = \|w\|_2^2$, where deviations exceeding 10^{-12} indicate encoder failure. Resonance quality $Q_{\text{dB}} = 20 \log_{10}(\text{peak/background})$ is measured per query, with the system requiring $Q \geq 6$ dB for reliable retrieval. Interference $\eta = E_{\text{destructive}}/E_{\text{total}}$ is tracked in telemetry, with the system maintaining $\eta_{\text{residual}} \leq 0.15 \cdot \eta_{\text{max}}$ (where η_{max} is the configured admission-control interference ceiling) to prevent excessive pattern cancellation. Conductivity $\kappa = \|\Pi\psi\|_2/\|\psi\|_2$ is computable per signal, measuring what fraction of energy lies within the projector’s passband. The following table summarizes the falsifiable predictions and their validation status:

Prediction	Formula/Threshold	Where Verified
Energy preservation	$\ E(w)\ _2^2 = \ w\ _2^2$ (deviation $\leq 1\text{e-}12$)	Every write, invariant-validated
Resonance quality	$Q_{\text{dB}} \geq 6$ dB for retrieval	Per query, invariant-validated
Bounded interference	$\eta_{\text{residual}} \leq 0.15 \cdot \eta_{\text{max}}$	Telemetry, invariant-validated
Conductivity in-band	$\kappa \geq 0.95$	Per signal, invariant-validated
Capacity margin	P99 margin $\geq 1.3\times$	Byte channel, invariant-validated
AEAD exact recall	100% pass rate, 0% integrity failures	CI gate, invariant-validated
PDE stability	$\max_k G_k \leq 0.98$ when enabled	Per step, invariant-validated

All predictions above have been empirically validated and continuously tested. Observed values consistently exceed their thresholds with significant margin (e.g. resonance quality routinely measures 12-18 dB against a 6 dB threshold; conductivity exceeds 0.99 against a 0.95 threshold). Exact measured values are published in the invariant artifacts and verification notebooks.

Threshold rationale: The 6 dB resonance quality threshold represents a 4x power ratio between signal and background—the minimum separation for reliable detection in signal processing applications. The conductivity threshold $\kappa \geq 0.95$ ensures at least 95% of signal energy passes through the projector’s passband, preserving signal integrity with minimal attenuation. The PDE stability bound $|G_k| \leq 0.98$ provides a 2% margin below the strict stability limit of 1.0, accounting for numerical precision in floating-point computation.

If any prediction fails, the framework is falsified for that configuration. All invariants are validated in continuous integration.

Two senses of falsifiability should be distinguished. The invariants above are conformance guarantees: they test that the implementation obeys the mathematics it claims—energy preserved, interference bounded, resonance measurable—and a violation falsifies the implementation for that configuration. They do not by themselves establish superiority over alternative architectures; that is the role of the comparative benchmark evaluation. We use “falsifiable” in this conformance sense throughout.

2.1 Empirical Validation

This subsection summarizes the RFS implementation and the properties continuously measured on it.

Implementation Overview

RFS implements the full mathematical framework as follows. The 4D field is a complex tensor $\Psi \in \mathbb{C}^{D_x \times D_y \times D_z \times D_t}$ with configurable dimensions for each axis, allowing the field to be sized appropriately for the application’s capacity and latency requirements. Wave encoding uses FFT-based transforms with phase masks and configurable basis functions including harmonic, Gabor, and learned variants. The dual-channel architecture separates associative (semantic) and byte (exact recall) channels with guard-band separation to prevent interference between the two retrieval modes. A projector operator enforces band separation with verified conductivity $\kappa \geq 0.95$ in-band, ensuring that signals in the intended band pass through with minimal attenuation. Decay is configurable either per-document (individual items can have different persistence) or global (uniform decay across all content), with projector cadence bounds ensuring the decay rate remains stable. Finally, production telemetry tracks resonance quality Q , interference η , capacity margin, and energy budget in real-time, providing observability into system health. A comprehensive suite of mathematical invariants—covering energy preservation, bounded interference, resonance quality, capacity margins, and operator conductivity—is tested in continuous integration; verification notebooks exercise each operator and bound. Recent implementation advances include SRFT (Structured Random Fourier Transform) spectral placement formalized with six dedicated invariants governing deterministic mapping from embedding space to 4D lattice coordinates, and the DECOP (Decision Operator) post-retrieval reranking module that replaces the earlier scalar fusion approach with signed energy-space composition $E_{\text{total}} = E_v + \beta \cdot E_s$ (macro $F1 = 0.6807$ on SciFact claim verification; methodology in the companion evaluation document).

Retrieval Paths

A distinctive feature of RFS is that it provides multiple query paths from a single unified storage substrate:

Path	Description	Use Case
Vector	High-throughput cosine similarity	Latency-critical workloads
Interference	Field-native resonance with interference patterns	Quality + explainability
Exact recall	AEAD-decrypted byte-perfect reconstruction	Integrity-critical retrieval
Proactive discovery	Field-native clustering and anomaly detection without explicit queries	Unsupervised insight extraction

The vector path performs high-throughput cosine similarity retrieval, optimized for latency-critical workloads where raw speed matters most. The interference path uses field-native resonance with EventFrame complex-phase vectors, providing richer explainability metrics at the cost of additional computation. Exact recall reconstructs byte-perfect content from the AEAD-protected byte channel. A proactive discovery path performs field-native clustering and anomaly detection directly on the superposed field state, extracting structural insights without requiring an explicit query—enabling the system to surface patterns the user has not yet asked about. Critically, all paths operate on the same underlying field state—there is no separate index or duplicate storage. A single ingestion operation populates every path. The field state Ψ serves the fast vector path, the explainable interference path, the proactive discovery path, and the exact recall channel. This contrasts with systems that require separate indexes for different query modalities. The vector path is designed for low latency and high throughput: RFS stores documents already encoded in the field representation, so queries require correlation rather than graph traversal. Comparative latency and throughput measurements are reported in the companion evaluation document. The interference path is slower because it computes full interference patterns and explainability metrics, but this tradeoff is acceptable for applications where understanding why a document was retrieved matters as much as retrieving it.

Benchmark Evaluation

Note: Benchmark corpus, metrics, and methodology are being updated for the current field-native pipeline. This section describes the evaluation framework and what RFS benchmarks measure; specific numbers will be published in a companion evaluation document.

RFS is evaluated against production-grade vector databases (e.g. FAISS HNSW[8][15]) using standard information retrieval benchmarks (e.g. BEIR[1]). Standard metrics include nDCG@10 (normalized discounted cumulative gain, measuring ranking quality with position-weighted relevance), Recall@10 (fraction of relevant documents retrieved in top 10), and MRR@10 (mean reciprocal rank, measuring how high the first relevant result appears).

RFS is designed to match vector database quality rather than exceed it. The significance lies not in quality improvement but in what RFS provides alongside equivalent quality: unified storage, multiple query paths from a single field state, exact recall with AEAD integrity verification, and built-in explainability metrics (Q, η) that vector databases cannot offer. Performance characteristics (latency, throughput) depend on hardware, corpus size, and encoder choice.

Key Measured Properties

Property	Metric	Threshold	Status
Energy conservation	$\ Ew\ _2^2 / \ w\ _2^2$	$= 1.0$ (deviation $\leq 1\text{e-}12$)	Validated
Resonance clarity	Q (dB)	≥ 6 dB	Validated, exceeds threshold
Interference	η	≤ 0.15	Validated
Conductivity	κ (in-band)	≥ 0.95	Validated, exceeds threshold
Capacity margin	P99	$\geq 1.3\text{x}$	Validated

All properties are continuously validated through invariant checks in CI. Exact measured values are recorded in verification notebook artifacts and updated with each pipeline run.

Complexity Analysis

Operation	Complexity	Algorithm
Encode	$\mathcal{O}(D \log D)$	FFT
Project	$\mathcal{O}(D)$	Band mask
Query	$\mathcal{O}(D \log D)$	Matched filter
Exact recall	$\mathcal{O}(D \log D)$	Inverse FFT + AEAD

The matching complexities for Encode, Query, and Exact Recall are not coincidental— they reflect a fundamental design property. All three operations require global frequency-domain mixing via FFT, which costs $\mathcal{O}(D \log D)$ regardless of what the operation conceptually represents. Project costs only $\mathcal{O}(D)$ because band masking touches each element independently without global mixing. Critically, none of these complexities depend on N (the number of stored documents) at fixed field dimension. Whether the field contains 1,000 or 1,000,000 superposed patterns, query cost remains $\mathcal{O}(D \log D)$ — the system pays once for the field dimension, not per document. The tradeoff is capacity: by Theorem 1, cross-talk variance scales inversely with D , so sustaining a target interference level as the corpus grows requires provisioning D with N . The honest comparison with index-based systems is therefore $\mathcal{O}(D \log D)$ per query with D sized for the workload’s capacity needs, versus $\mathcal{O}(\log N)$ query over an $\mathcal{O}(N)$ index; RFS exchanges index construction and maintenance for dimension provisioning, and the byte channel for exact recall remains $\mathcal{O}(N)$ in storage. The invariant checks validate the implementation-level claims: energy conservation (Parseval), bounded interference (guardrails), and measurable resonance (Q metric). Having demonstrated that the framework produces measurable results, we now explain the underlying mathematical structure.

3. How It Works: Mathematical Framework

The substrate that enables cognitive integration is a shared wave-based memory space where all AI components communicate by writing and reading signal patterns. Rather than passing messages through APIs or attention layers, components encode their information into a common field and sense other components’ contributions by measuring pattern overlap. This approach provides three

key advantages over conventional architectures: (1) global access—any component can read any pattern without explicit routing, (2) constant-time retrieval—query cost depends on field dimension, not document count, and (3) natural composition—multiple patterns coexist through superposition rather than requiring separate storage slots.

Formally, this shared memory is a Hilbert space $\mathcal{H}$, which in the RFS implementation is the finite-dimensional space $\mathcal{H} = \mathbb{C}^D$ of complex D -dimensional vectors. The inner product $\langle \Psi, \Phi \rangle = \Phi^H \Psi$ measures the overlap between two patterns—large overlap indicates strong relationship, near-zero overlap indicates independence. Complex numbers are essential, not incidental: each value has both magnitude (signal strength) and phase (timing/offset), and phase differences allow multiple items to coexist in the same space through constructive and destructive interference. Without phase, superposition would cause immediate collision; with phase, patterns can be separated and selectively retrieved.

The term “operator algebra” refers to the rules governing how information flows through the system. Operators define how information is written (encoding E), how it is read (decoding E^H), how it is filtered (projectors Π), and how it evolves over time (dynamics $U(t)$). These operators replace what would otherwise be database queries, neural attention mechanisms, or graph traversals—but with mathematical guarantees. Specifically, we use unitary transformations (such as Fourier transforms) that preserve energy: the total signal strength before and after any operation remains constant. This energy preservation (formalized through Parseval’s theorem) guarantees stability—no runaway amplification, no mysterious signal loss, and predictable interference behavior. Different basis choices (time-domain vs. frequency-domain) are simply different views of the same underlying field, connected by invertible transforms.

3.1 The 4D Field Lattice

In the concrete implementation (RFS), the field is a 4-dimensional complex tensor $\Psi(x, y, z, t) \in \mathbb{C}$, discretized as $\Psi \in \mathbb{C}^D$ where $D = D_x \times D_y \times D_z \times D_t$.

This is not an arbitrary choice—it reflects the physical structure of information storage. The three spatial dimensions (x, y, z) allow documents to occupy distinct “locations” in the field, enabling spatial multiplexing where different semantic clusters occupy different regions of frequency space. The interference patterns that arise when documents share spectral support directly encode semantic relationships: documents with similar content produce constructive interference (high correlation), while unrelated documents produce low correlation due to pseudo-random phase separation. The temporal dimension (t) enables recency weighting, memory consolidation over time, temporal context for sequence-dependent retrieval, and exponential decay dynamics that cause older memories to fade unless reinforced.

The field space $\mathcal{H} = \mathbb{C}^D$ is a Hilbert space with inner product:

$$\langle \Psi, \Phi \rangle = \Phi^H \Psi = \sum_{i=1}^D \overline{\Phi_i} \Psi_i$$

This inner product is fundamental: it measures the similarity between two field states. When we store multiple documents, their overlap (inner product) directly encodes their relationship—high overlap indicates semantic similarity, while orthogonality indicates independence.

3.2 Why Waves? The Physics of Superposition

The choice of wave-based representation exploits fundamental properties of wave physics:

Superposition. Waves naturally superpose: when two waves occupy the same medium, they add linearly. This means N documents can be stored in the same field as $\Psi = \sum_{i=1}^N \psi_i$, with the field growing only in amplitude, not in dimension. Traditional databases require $\mathcal{O}(N)$ storage; wave storage requires $\mathcal{O}(D)$ for the associative channel regardless of N (up to capacity limits). The byte channel for exact recall remains $\mathcal{O}(N)$.

Interference. When waves with similar frequencies occupy the same region, they interfere. Constructive interference occurs when waves are in-phase—their amplitudes add, creating peaks that indicate semantic agreement between documents. Destructive interference occurs when waves are out-of-phase—their amplitudes cancel, creating nulls that may indicate contradiction or semantic tension. This interference is not a bug but a feature: the pattern of constructive and destructive interference across the field encodes the semantic structure of the stored content, with related documents reinforcing each other and unrelated documents remaining independent.

Resonance. A system resonates when driven at its natural frequency. In our framework, querying is resonance: we inject a probe waveform, and stored patterns that match the probe’s frequency content resonate strongly. The resonance Q metric measures how clearly a signal stands out:

$$Q_{\text{dB}} = 20 \log_{10} \left(\frac{\text{peak amplitude}}{\text{background amplitude}} \right) = 10 \log_{10} \left(\frac{P_{\text{peak}}}{P_{\text{background}}} \right)$$

where $P = |r|^2$ is power. Higher Q means cleaner signal, easier retrieval, more confident matches.

3.3 The Damped Wave Equation

The field evolves according to a damped wave equation:

$$\frac{\partial^2 \Psi}{\partial t^2} + \gamma \frac{\partial \Psi}{\partial t} = c^2 \nabla^2 \Psi$$

where $\gamma > 0$ is the damping coefficient and c is the wave speed. This PDE governs how disturbances propagate through the field. Without damping ($\gamma = 0$), waves propagate indefinitely and energy is perfectly conserved, but old information never fades—a system without forgetting. With damping ($\gamma > 0$), waves decay exponentially, implementing natural forgetting: old memories fade unless reinforced through re-encoding or retrieval. This tradeoff between preservation and forgetting is fundamental to any memory system, and the damping coefficient provides a tunable control.

In the RFS implementation, PDE evolution is **feature-gated** (default off) because full wave dynamics add computational overhead and complexity that is unnecessary for many retrieval workloads. The MVP uses decay-first dynamics—per-document exponential decay with periodic re-projection—which provides the essential forgetting behavior without requiring continuous PDE integration. When enabled, semi-implicit Crank-Nicolson integration ensures numerical stability:

$$\Psi^{n+1} = \left(I + \frac{\Delta t}{2} A \right)^{-1} \left(I - \frac{\Delta t}{2} A \right) \Psi^n$$

where A is the discretized wave operator. The gain factors $|G_k|$ must satisfy $\max_k |G_k| \leq 1$ to prevent blow-up. Lemma 1 (PDE Stability). Under Crank-Nicolson integration with projector Π applied every Δt_{proj} steps, the field norm satisfies:

$$\|\Psi(t + \Delta t_{\text{proj}})\|_2 \leq \rho(A)^{\Delta t_{\text{proj}}} \|\Psi(t)\|_2 + k \cdot \epsilon_{\text{trunc}}$$

where $\rho(A) < 1$ is the spectral radius of the damped evolution operator, k is the number of writes, and ϵ_{trunc} is truncation error. This bound is validated through numerical experiments with random initial conditions and varying write sequences.

3.4 Encoder and Decoder Operators

Each AI module M_i (for $i = 1, 2, \dots, N$) is associated with an encoder operator $E_i : X_i \rightarrow \mathcal{H}$, which maps the module’s internal state (or outputs) from its native space X_i (e.g. $\mathbb{R}^{n_i}$ for an n_i -dimensional hidden state) into the field space $\mathcal{H}$. Correspondingly, there is a decoder (probe) operator $E_i^H : \mathcal{H} \rightarrow X_i$ (the Hermitian adjoint of E_i) that projects a field pattern back into the module’s state space.

In the concrete RFS implementation, encoding proceeds through a multi-stage pipeline:

Stage 1: Semantic Embedding. Raw content (text, image, etc.) is mapped to a semantic vector $v \in \mathbb{R}^n$ via a pluggable encoder. The implementation supports TF-IDF (token-based, dimension-normalized) and transformer-based encoders (e.g. BGE); the encoder choice affects retrieval quality but not the field-theoretic guarantees. The architecture is designed to accept any encoder that produces fixed-dimensional vectors suitable for similarity and field projection.

Stage 2: Spatial Placement and Tri-band Packing. The embedding is placed in the 4D lattice using spatial mapping (e.g. PCA-based projection from embedding space to lattice coordinates). Documents are packed into tri-band logical records: (1) a vector payload for cosine-similarity retrieval, (2) an associative payload (EventFrame complex-phase vectors) for resonance-based retrieval, and (3) an AEAD-protected byte payload for exact recall. Guard-band separation ensures the associative and byte channels do not interfere. Optional whitening ($PP^T = \Sigma^{-1}$) can normalize embeddings for predictable interference statistics; optional error-correcting codes add redundancy for robustness to decay or numerical precision limits.

Stage 3: Associative Encoding and Waveform Synthesis. The associative payload is produced by EventFrame encoding: role-based extraction (agent, verb, patient, etc.) yields complex-phase vectors that combine via superposition. Phase orthogonality (Theorem 1) applies to the associative subspace, enabling multiple documents to coexist without collision. The theoretical waveform synthesis is:

$$\psi = E(w) = \mathcal{F}^{-1}(M \odot \mathcal{F}(H \cdot w))$$

where $H : \mathbb{R}^n \rightarrow \mathbb{C}^D$ spreads symbols across the field, M is a diagonal phase mask with $|M_{ii}| = 1$ (unit modulus), and $\mathcal{F}$ is the unitary FFT (norm = “ortho”). Implementations may use discrete lattice placement with tri-band records while preserving the same interference and resonance properties.

Phase masks are central to storing many documents without collision. Each document receives a unique phase mask M_k with entries $M_k[i] = e^{j\theta_{k,i}}$ where phases are pseudo-randomly assigned. Theorem 1 (Phase Orthogonality). For i.i.d. uniform phases on $[0, 2\pi)$:

$$\mathbb{E}[\langle M_i \odot x, M_j \odot x \rangle] = 0 \quad \text{for } i \neq j$$

$$\text{Var}[\langle M_i \odot x, M_j \odot x \rangle] = \|x\|_4^4 = \frac{\|x\|_2^4}{D} \quad \text{under uniform spectral spreading } |x_k| = \|x\|_2/\sqrt{D}$$

When the spreading operator distributes energy uniformly across the lattice, the variance decreases with field dimension D , enabling more documents with less interference.

These operators E_i and E_i^H are analogous to “sense” and “act” interfaces between the module and the shared field: E_i encodes local information into a global representation, and E_i^H reads out global information in the context of module M_i . In many cases $E_i^H E_i \approx I$ (identity on X_i) so that encoding followed by decoding returns the original state (the operators may be approximately unitary or involve minor information loss). The simplest example is to let E_i pick a designated subspace (or sub-band) of the field for module M_i and deposit its state there (e.g. by embedding X_i as a subset of coordinates in $\mathcal{H}$).

However, more sophisticated constructions use orthonormal basis functions to spread each module’s contribution across $\mathcal{H}$ in a way that remains distinguishable from others yet spectrally compatible[16][17]. For instance, one could assign each module a distinct frequency band of a Fourier basis; E_i would then Fourier-transform the module’s state and place it into that band of the field. This frequency-division multiplexing is mathematically convenient, as Parseval’s identity guarantees that the norm (energy) of the signal is preserved in the field[18], and different bands do not interfere. Indeed, any orthonormal decomposition of $\mathcal{H}$ ensures that the sum of energies of individual contributions equals the total energy in the field – a property that we will leverage as a global invariant.

Field superposition and memory storage. Once encoded, the contributions from all modules sum (superpose) to form the global field state $\Psi \in \mathcal{H}$. We can write: $\Psi = \sum_{i=1}^N E_i(w_i)$, where $w_i \in X_i$ represents the state (e.g. output or salient latent variables) of module M_i that we choose to write to the field.

The field Ψ thus holds a distributed representation of the collection w_i of module states. Importantly, this superposition does not simply concatenate or separate the states, but rather allows them to potentially interact or overlap in the field representation (depending on E_i design). In the special case that all E_i map into orthogonal subspaces of $\mathcal{H}$, the different w_i do not interfere at all in Ψ (perfect information separation). But more generally, we may allow controlled overlaps so that Ψ contains higher-level composite patterns (e.g. correlations between modalities).

The field can be seen as a kind of working memory or shared blackboard where the latest information from each subsystem is aggregated. Any module can subsequently read from Ψ using its decoder: $\hat{w}_i = E_i^H \Psi$ provides module M_i with a projection of the global state back into its own feature space. This readout $\hat{w}_i$ will equal the module’s own original contribution w_i plus any additional signal components in Ψ that lie in M_i ’s decoding range. Thus, module M_i can query the field for “what is out there that I can understand.” In effect, each module shares a portion of a collective mind in Ψ .

If $E_j(w_j)$ for some $j \neq i$ has components that align (in the Hilbert inner product sense) with E_i 's range, then M_i 's readout $\hat{w}_i$ will include information about M_j 's state. This mechanism enables inter-model communication: one network's state influences another via the common mathematical projection. We will discuss in Section 4 how this allows modules to become aware of each other's presence and even form associations between their internal representations.

3.5 Operator Constraints and Guardrails

To ensure well-behaved dynamics in the field, we impose several constraints on the operators and states, analogous to how physical and chemical laws constrain interactions. We define a projector $\Pi : \mathcal{H} \rightarrow \mathcal{H}$ onto an allowed subspace of the field, representing an associative passband or "safe operating zone." After each write operation, the field state is filtered: $\Psi := \Pi(\Psi)$. This projector can, for example, enforce bandwidth limits (only certain frequency components are retained) or enforce an overall energy normalization. Such a projection step ensures that no uncontrolled accumulation of power outside the designated spectrum can occur[16]. In signal-processing terms, it functions like a band-limited reservoir that prevents high-frequency noise or unexpected modes from destabilizing the superposition.

The Projector as Computational Myelin. The projector Π is a frequency-domain filter:

$$\Pi = \mathcal{F}^{-1} \cdot M_{\text{passband}} \cdot \mathcal{F}$$

where M_{passband} is a binary mask indicating which frequencies belong to the associative band. Definition 1 (Conductivity). The conductivity of a signal through the projector is:

$$\kappa(\psi) = \frac{\|\Pi \cdot \psi\|_2}{\|\psi\|_2}$$

This measures what fraction of the signal's energy lies within the passband. When $\kappa = 1$, the signal is fully in-band with perfect transmission—all energy passes through the projector unchanged. When $\kappa = 0$, the signal is fully out-of-band and blocked entirely—no energy passes through. Intermediate values $0 < \kappa < 1$ indicate partial transmission, where some energy passes through while some is attenuated.

Biological analog: The projector functions like myelin in neural systems. Myelinated pathways conduct signals efficiently; unmyelinated pathways conduct poorly. The passband is the "myelinated" region of frequency space.

Guard Bands and Dual-Channel Separation. Between the associative (semantic) and byte (exact recall) bands, we reserve a **guard band** where no signals are placed:

$$\text{supp}(\hat{\psi}_{\text{assoc}}) \cap \text{supp}(\hat{\psi}_{\text{byte}}) = \emptyset$$

This ensures exact recall is not corrupted by semantic interference and vice versa. RFS implements multiple retrieval paths from this unified storage. The vector path returns documents ranked by cosine similarity over the vector payload. The interference path returns documents ranked by resonance strength over the EventFrame complex-phase vectors. A proactive discovery path performs field-native clustering directly on the superposed state, surfacing structural patterns without requiring an explicit query. All retrieval operations execute in $\mathcal{O}(D \log D)$ time independent of the

number of stored documents N —a key advantage over index-based approaches. Exact recall reconstructs the original bytes by reading from the byte channel with AEAD integrity verification, guaranteeing bit-perfect reconstruction with cryptographic proof of authenticity. Interference and Energy Guardrails. We define the total field energy $E_{\text{tot}} = \|\Psi\|_2^2$ (which equals $\sum_i \|w_i\|_2^2$ if E_i are orthonormal injections by Parseval’s theorem[18]). Each new contribution should not excessively spike E beyond design limits. The **interference ratio** η measures destructive overlap:

$$\eta = \frac{E_{\text{destructive}}}{E_{\text{total}}} = \frac{\sum_{i < j} |\langle \psi_i, \psi_j \rangle| \mathbb{1}[\text{destructive}]}{\|\Psi\|_2^2}$$

The system enforces specific thresholds to maintain stability, each chosen based on empirical validation and signal processing principles. First, $\eta_{\text{residual}} \leq 0.15 \cdot \eta_{\text{max}}$ prevents excessive cancellation between stored patterns—empirically, interference above 15% of maximum degrades retrieval quality measurably, while below this threshold the matched filter reliably separates patterns. Second, $E_{\text{tot}} \leq E_{\text{max}}$ prevents numerical overflow and maintains adequate signal-to-noise ratio. Third, the capacity margin ($P99 \geq 1.3\times$) ensures the byte channel has at least 30% headroom above current utilization, preventing overflow during burst write periods. If a proposed write would cause too high interference (too much overlap between a new pattern and existing ones), the system’s admission control can reject or modulate that write (analogous to chemical reaction rates slowing as certain concentrations get too high). These guardrails echo chemical constraints: just as reactions have conservation laws and equilibrium constants preventing runaway processes, our field has invariants and checks (spectral norms, energy bounds) to prevent pathological interactions. For example, one guardrail might require $\max |\Psi_k| < \tau$ for each field dimension k , capping any single mode’s amplitude to avoid one contributor drowning out others[34]. Another guardrail could enforce that the addition of a new $\psi_d = E_d(w_d)$ does not significantly leak outside the projector’s passband (ensuring the new data lives in the same subspace as the rest)[16]. By incorporating such constraints, we ensure that the compositionality of the field – the ability to hold many superposed items – is maintained without mutual destruction. Dynamics and operators in time. Beyond static encoding/decoding, we introduce an evolution operator $U(t)$ that governs the time dynamics of the field when no new writes are occurring (or in parallel with continuous updates). In analogy to physics, $U(t)$ could be defined via a Hamiltonian H on $\mathcal{H}$ such that $U(t) = \exp(-i_H t)$ (resembling Schrödinger evolution for a quantum state) or via a dissipative flow that minimizes an energy functional (resembling a diffusion or gradient descent in an energy landscape). The choice of dynamics depends on the intended cognitive function. For associative memory retrieval, an energy-minimizing dynamic (like the Hopfield network updating rule) can drive the field toward a stored attractor pattern that best matches a cue[9]. For integrating information over time, a unitary (information-preserving) evolution might better model a reverberating working memory. We can accommodate both by splitting $H = H_{\text{conservative}} + H_{\text{dissipative}}$; the former yields oscillatory or reversible dynamics (good for sustaining patterns), while the latter yields contraction to stable states (good for settling to an attractor). An example of a conservative part is a coupled oscillator Hamiltonian that produces oscillatory synchrony between related field components (modeling e.g. neural oscillation binding); an example of a dissipative part is a Lyapunov function $V(\Psi)$ that decreases over time, ensuring Ψ converges to a minimum of V . In practice, we might implement dissipative dynamics through iterative updates: $\Psi \leftarrow \Psi - \eta \nabla V(\Psi)$ for a small step η , perhaps interleaved with new writes. One concrete realization is to use a diffusion operator in the spectral domain: between write cycles, apply a smoothing filter to Ψ . In the Fourier basis this could be $U(\Delta_t) : \Psi \mapsto \mathcal{F} - 1[G \cdot (\mathcal{F}\Psi)]$, where $\mathcal{F}$ is the Fourier transform and $G = \exp(-\lambda_{\Delta t \omega} 2)$ a Gaussian

filter that slightly blurs the field (here ω is frequency and λ a diffusion rate). This would simulate a Crank–Nicolson integration of a diffusion PDE on the field, spreading information slightly among neighboring modes. Such smoothing can serve as a memory consolidation step, blending overlapping contributions in a controlled manner and potentially improving robustness by averaging out noise. However, this dynamic is applied only when interference is low and stability allows, per guardrails that disable it if the field is in a volatile state[34]. Thus, the field dynamics are adaptive: when the system is actively integrating new information or if interference is high, it holds Ψ static (to preserve data integrity); when idle or stable, it “relaxes” Ψ in a way that enhances useful overlap and decays random perturbations. Summary of formalism. Our operator framework can be summarized by a high-level master equation for system operation, inspired by orchestration in computational pipelines:

$$\text{AIResponse}(\text{inputs}) = \text{Guardrails} \left(\sum_{i=1}^N E_i(M_i(\text{input}_i)) + U(\Delta t) + \dots \right)$$

meaning: each module M_i processes its input, gets an output w_i , which is encoded into the field via E_i ; all contributions sum and undergo any scheduled field dynamics;

the guardrails then validate the resulting Ψ (projecting and monitoring as needed); finally, the results can be decoded out to produce outputs or to inform the next cycle of processing (e.g. another iteration or a decision). The precise sequence may vary by context (for instance, some queries might involve writing a probe to the field and reading out a set of matching results – see Section 4). But critically, all operations are defined at the level of linear operators, inner products, and simple nonlinearities (like saturations in guardrails or the specific form of $V(\Psi)$). This formal structure lets us leverage a rich body of mathematics – functional analysis, spectral theory, linear algebra – to analyze and design the system. It also gives a common language for different AI components. Regardless of their internal workings (be it backprop-trained neural nets or heuristic algorithms), they interact through the same mathematical currency: vectors in $\mathcal{H}$ and operations on them. In this way, the framework acts as an interlingua or common protocol – much like the nervous system uses electrical impulses as a unifying language to coordinate muscles, organs, and senses. In the following sections, we explore the implications and analogies of this framework. We will see that it naturally reproduces several cognitive phenomena: stable attractor states that can store memories or goals and a field-based form of awareness and attention (Section 4), and the ability to shape system behavior by tuning the field’s landscape (Section 7). We will also draw explicit parallels to physics, chemistry, and biology (Section 6), which not only provide intuition but also suggest design principles (e.g. using Hamiltonian dynamics for stability, or chemical-like feedback loops for regulation). Before delving into those, we summarize the structural metaphors that guided our framework’s construction, linking each aspect to its inspiration in the natural sciences.

3.6 Energy Conservation: Parseval’s Theorem

Every encoding and transform operation preserves energy—not as a design preference, but as a mathematical necessity for a stable, predictable system. (Dissipative dynamics—damping in Section 3.3 and temporal decay in Section 7.3—remove energy deliberately and are governed separately.) Theorem 2 (Parseval Energy Conservation; classical result, restated for the encoder). For the unitary FFT and unit-modulus masks:

$$\|E(w)\|_2^2 = \|w\|_2^2$$

Proof. Unit-modulus masks preserve norm: $\|M \odot x\|_2^2 = \sum_i |M_i|^2 |x_i|^2 = \|x\|_2^2$. The unitary FFT preserves norm by Parseval: $\|\mathcal{F}(x)\|_2^2 = \|x\|_2^2$. Composition of norm-preserving operators preserves norm. $\square$ Implication: Every write adds exactly as much energy as the input. No hidden amplification, no energy leakage. The field’s total energy is the sum of document energies (plus interference terms, which are bounded by η).

4. Memory and Retrieval Mechanisms

Having established the mathematical framework in Section 3, we now examine how the field substrate supports memory storage and retrieval. A key feature of this approach is the emergence of field-based awareness—a form of global visibility where disparate components gain access to a shared state. We use the term “awareness” in a functional sense: each module can sense what other modules have contributed to the field, enabling coordination without explicit message passing. This is not a claim about consciousness or phenomenal experience, but a description of an information-sharing mechanism. This is accompanied by a powerful memory mechanism grounded in attractor dynamics and resonance. (Resonance and matched-filter retrieval are implemented in RFS; energy-descent attractor dynamics are part of the theoretical layer—see Section 7.) We term this architecture a Resonant Field Substrate (RFS) to highlight that memory recall and inter-module communication occur via resonance phenomena in the field. In cognitive science, Global Workspace Theory (GWT) postulates a central information exchange where content, once broadcast in the global workspace, becomes accessible to multiple processes, enabling integrated thought[3]. Our field Ψ acts exactly as such a workspace – any information encoded into Ψ by one module can, in principle, be decoded by all others (subject to encoding overlaps). The “awareness” is not mysterious: it is implemented as the set of all decodings $E_j^H \Psi$ for each module M_j . If M_j ’s readout yields a significant signal concerning M_i ’s contribution, we say M_j is aware of what M_i has posted to the field. This can be made symmetric by design (e.g. through reciprocal encoding overlaps, M_i and M_j can sense each other). In effect, Ψ functions like a bulletin board that all modules subscribe to.

4.1 The Matched Filter: Optimal Retrieval

Retrieval in the Resonant Field Substrate employs the matched filter, a classical result from signal processing that provides the optimal linear detector for a known signal embedded in additive noise. The matched filter for detecting a pattern s in a noisy observation $y = s + n$ is simply the conjugate transpose s^H , which computes the inner product $\langle s, y \rangle = s^H y$. This maximizes the signal-to-noise ratio among all linear filters. In our context, the field Ψ contains superposed contributions from N documents, and we wish to detect the presence and strength of a query pattern. Given a query vector q with encoding $E(q)$, the matched filter output is:

$$r = E^H \Psi = E^H \left(\sum_{i=1}^N E(w_i) \right) = \sum_{i=1}^N E^H E(w_i)$$

For a target document w_t that we wish to retrieve, the matched filter decomposes into signal and interference components:

$$r_t = E^H E(w_t) + \sum_{i \neq t} E^H E(w_i)$$

The first term represents the signal—the self-correlation between the query and the target, which is maximized when query and target match. The second term represents interference—the cross-correlation with all other stored documents, which the phase orthogonality property (Theorem 1) ensures is small in expectation.

Theorem 3 (Matched Filter Optimality; classical result, restated). Let $E : X \rightarrow \mathcal{H}$ be an encoding operator, let $w \in X$ be a document, and let $n \in \mathcal{H}$ be additive white noise with per-component variance σ_n^2 . Among all linear filters $F : \mathcal{H} \rightarrow \mathbb{C}$, the matched filter $F = E^H$ maximizes the signal-to-noise ratio for detecting $E(w)$:

$$\text{SNR} = \frac{|E^H E(w)|^2}{\mathbb{E}[|E^H n|^2]} = \frac{\|E(w)\|_2^4}{\sigma_n^2 \|E(w)\|_2^2} = \frac{\|E(w)\|_2^2}{\sigma_n^2}$$

This result follows from the Cauchy-Schwarz inequality and is standard in detection theory. In operation, the effective noise also contains structured cross-talk from other stored documents (the interference term above), which is neither white nor independent; the white-noise result is the idealized baseline, with interference statistics governed by Theorem 1 and the η guardrail. The practical implication is that the adjoint operator E^H is not merely a convenient choice for decoding but the mathematically optimal one under linear filtering assumptions. Theorem 4 (Query Complexity Independence). Let $\Psi = \sum_{i=1}^N E(w_i)$ be the superposed field state containing N encoded documents, and let E^H be the matched filter operator implemented via FFT-accelerated correlation. Then the computational complexity of querying Ψ is $\mathcal{O}(D \log D)$, independent of the number of documents N at fixed field dimension D . Proof. The key insight is that all documents are superposed in a single field state Ψ . The query operation computes $r = E^H \Psi$, which involves: (1) an FFT of the query, (2) element-wise multiplication with the (already transformed) field, and (3) an inverse FFT. Each FFT costs $\mathcal{O}(D \log D)$ where D is the field dimension. The operation acts on Ψ directly, not on individual documents—there is no loop over N documents. Thus complexity depends only on field dimension D , not on document count N . $\square$ This independence from N is a fundamental advantage over index-based approaches like HNSW, where query complexity grows (albeit slowly) with dataset size. In RFS, adding more documents to the field does not increase query time; it only affects interference levels, which are managed through the guardrails described in Section 3.5. Maintaining a target interference level as N grows requires provisioning D accordingly (Section 2, Complexity Analysis).

4.2 Selective Attention via Resonance

A key question is: with potentially many signals superposed in Ψ , how does a particular module pick out what’s relevant to it? The answer lies in the inner product structure – modules naturally resonate with content that “matches” their decoders. If module M_j has a decoder E_j^H attuned to certain patterns (say frequencies or features), then when Ψ contains a component in that subspace, the inner product $\langle \Psi, E_j(x) \rangle$ will be high for that x . This is analogous to how a radio tuner picks up a station at its resonant frequency. In neural terms, M_j will respond (get activated) if the global workspace contains an item coded in a form M_j can process. This yields a form of attention: modules naturally focus on aspects of the field that align with their expected patterns. If needed, we can sharpen this by dynamic gating – e.g., modules can modulate their E_j to be more or less receptive. But even without explicit gating, the math provides a filtering: $E_j^H \Psi$ projects out only M_j ’s representational subspace content from Ψ . Thus, each module “sees” the global state from

its own perspective (much as different experts view a situation differently, yet there is a single underlying situation).

Content-addressable memory (theoretical layer): In the full framework, the field with attractors acts as an associative memory: store a pattern in Ψ , and later a fragment or cue will retrieve (resonate with) that pattern. Suppose a particular memory (comprising pieces across modules) corresponds to an attractor Ψ^* . If the system is presented with partial information – e.g. , module M_1 receives a cue w_1 that was part of Ψ^* – encoding w_1 into the field provides a starting Ψ near one basin of attraction. The dynamics (or iterative reading/writing among modules) then tend to complete the pattern by moving Ψ toward Ψ^* . For instance, in a simplified scenario, M_1 might put a cue, M_2 reads it and produces an associated piece w_2 , writes that, which M_3 reads, and so on, until all modules collectively reconstruct the full memory. This is essentially how Hopfield networks retrieve memories: by energy descent to the nearest stored attractor[9]. Our advantage is that the memory is distributed across modules (like a multimodal memory could have visual and verbal components combined). Because of the Hilbert space formalism, if memory patterns were encoded as orthogonal vectors in $\mathcal{H}$, they would not interfere, and one could superpose many memories (similar to superposition of multiple holograms)[22][24]. Even if not fully orthogonal, as long as they are separated by the attractor basins, they can coexist in the storage. The capacity of this memory depends on dimensionality D and the robustness of attractors, analogous to classic results (Hopfield nets can store $\sim 0.14N$ patterns for N neurons with some error tolerance). Here D plays a role akin to number of neurons. Crucially, memory retrieval is fast and content-based: the system does not search serially; rather, the dynamics simultaneously consider all stored patterns by virtue of the superposition principle (cf. superposition-based search in the quantum setting[20]). The cue “finds” its match through resonance. This is like an auto-associative recall – given part of a pattern, the field completes it. In cognitive terms, it provides a basis for pattern completion (e.g. , recognizing a noisy image by filling in missing pieces using memory) or for analogy (if a cue partially matches multiple stored patterns, it could blend them and perhaps yield a novel combination). Working memory and broadcast: Beyond long-term memory attractors, the field can hold transient working memory items – thoughts or intermediate results that need to be accessible. Because Ψ is persistent (until overwritten or decayed by dynamics), any module’s output placed in Ψ will remain available for others. For example, if module A computes a partial result and is done, module B can later pick it up from the field. This decouples time: asynchronous modules can communicate without direct call-return – as long as they share the field, one leaves information there for whenever another is ready to use it. The field thus serves as a buffer implementing temporal integration (like the brain keeps a memory of a stimulus after it disappears, so other processes can act on it). The guardrails ensure this memory doesn’t corrupt: e.g. , projector Π can keep the memory item within a subspace to avoid interference, and dynamics can be tuned not to degrade it for a certain duration. Notably, this allows multimodal integration in real-time. If an image module encodes “object=dog” and a language module encodes “word=dog” into the same Ψ , their co-occurrence will produce a combined state that a higher-level module could recognize as a match (e.g. confirming that the spoken word corresponds to the seen object). Indeed, this is similar to the approach in some multimodal AI systems that align latent spaces of vision and language[11].

For instance, CLIP (Contrastive Language-Image Pretraining) creates a joint embedding space for images and text. In our framework, such an embedding space is exactly the field $\mathcal{H}$: the vision encoder E_{vision} and language encoder E_{language} both map into $\mathcal{H}$, so that related images and captions end up as nearby vectors, enabling direct comparison. But beyond alignment, our field allows interaction: the image can influence the text module via the field, not just be compared post-hoc. This is more powerful than static embedding – it’s a live workspace where, for example, a reasoning

module could take an image description from the field and ask a question to a language module, all mediated by Ψ . Essentially, it turns separate networks into a unified multimodal mind. Case study: associative question-answering. As an illustrative scenario, imagine an AI that must answer a question like “Does the image show the same animal mentioned in the text?”. In a traditional pipeline, the image model would output a label, the text model extract the mentioned animal, then a comparison is made. In our field-based system, the process is fluid: the text module encodes the semantic concept of the animal into Ψ as a pattern T , the image module encodes its perception as a pattern I . These patterns superpose. If they match (same high-level features), the superposition Ψ might naturally fall into an attractor associated with that animal concept (since both cues point to the same memory). A decision module M_k monitoring Ψ could notice that Ψ has energy minima alignment (meaning I and T resonated to reinforce a concept) and thus output “Yes, same animal.” If they differ (say text says “dog” but image is a cat), the field might show two distinct peaks or a conflicting pattern (no single attractor cleanly fits both), and the decision module outputs “No, they differ.” This inference emerges from each modality’s information being present in one common cognitive space, rather than requiring a predefined comparison routine. The energy landscape does the comparison by how the patterns interact. Interpretability and inspection: Because our substrate is mathematical and structured, it also offers potential interpretability. The field Ψ at any time is like a snapshot of the system’s “mind.” We can imagine visualizing the components of Ψ (e.g., projecting Ψ onto basis patterns that correspond to known concepts). Much like neuroscientists can sometimes interpret EEG or fMRI patterns as corresponding to certain thoughts or stimuli, an engineer could inspect Ψ to debug what the AI is “currently thinking about.” Additionally, because each module’s perspective is given by $E_i^H \Psi$, we can individually inspect what each module is extracting from the global workspace. This aligns with the goal of transparent AI – being able to trace how information flows and is integrated. The linear nature of many operations means one could attribute outcomes back to particular inputs by following the superposition weights, akin to linear contribution analysis. To summarize this section: the Resonant Field Substrate endows an AI system with a global workspace for information sharing and a content-addressable memory for storing and retrieving patterns across modules. It achieves distributed awareness not by duplicating data everywhere, but by leveraging the mathematics of inner products – each module flexibly pulls what it needs. In the full framework, memory is stored in attractor states of the field, and partial cues trigger retrieval via dynamic evolution toward those attractors[25]; in the current implementation, retrieval is realized by the matched-filter and interference paths, with attractor dynamics on the Section 7 roadmap. This design echoes many aspects of human cognitive architecture (working memory, associative recall, multimodal integration via a unified awareness) and does so in a principled way using operators and vectors.

5. Positioning: Comparison to Existing Work

Our theory draws upon and intersects with multiple strands of research in machine learning, neuroscience, and even philosophy of mind. Here we compare and contrast to some prominent existing frameworks: Global Workspace Theory (GWT) and implementations: GWT[3] posits a cognitive architecture with a central workspace for information sharing among specialized unconscious processors. Our framework operationalizes a Global Workspace through the Hilbert field Ψ . In contrast to many AI implementations of GWT (such as BlackBoard systems or the Global Latent Workspace idea[11]), our approach provides a concrete operator-level mechanism for the workspace. VanRullen & Kanai (2021) envisioned using unsupervised translation between latent spaces to create an amodal workspace[11]. We achieve a similar end but via a unified representational field

where all modules are already in a common space (assuming encoders have been trained to facilitate that). This eliminates the need for pairwise translation networks and scales more naturally as modules increase. Additionally, GWT is often associated with broadcasting a single piece of information at a time (a “spotlight of attention”). Our model can broadcast complex, high-dimensional states (a superposition of many items) simultaneously, though mechanisms like attractor selection may effectively serialize some of it (similar to how attention focuses on one attractor at a time if there’s competition). We also expand GWT by adding the physics backbone – energy landscapes and attractors – which give it a dynamical law-governed character rather than being a black box global variable.

Conscious Turing Machine (CTM) and heterologous module interconnection: The Blums’ Conscious Turing Machine (CTM)[12] provides another formal architecture for GW-style interconnection between heterologous modules. CTM proposes a computational model where specialized processors (“chunks”) broadcast information through a central workspace, achieving integration through competition and gating. Our framework shares CTM’s goal of unifying heterogeneous modules through a central exchange, but differs in several important respects. First, the mechanism differs: CTM uses discrete competition and winner-take-all broadcasting to determine what enters the workspace, while our framework uses continuous superposition where all content coexists and attractor dynamics determine which patterns dominate attention. Second, the representation differs: CTM chunks are symbolic or discrete tokens, while our field supports continuous, high-dimensional states that can represent graded concepts and nuanced relationships. Third, the mathematical formalism differs: CTM extends classical Turing machine theory with finite automata for competition, while we use Hilbert space operators with energy conservation and spectral guarantees drawn from physics. Fourth, the implementation status differs: CTM remains largely conceptual with limited empirical validation, while RFS provides a working implementation whose invariant suite is validated in continuous integration, with live counts tracked in the repository invariant ledger. Both approaches address the same fundamental problem—integrating heterogeneous modules into a coherent cognitive system—but our operator-based approach provides explicit, testable guarantees (energy preservation, bounded interference, measurable Q) that CTM’s abstract formulation does not.

Neural blackboard and differentiable memory: The idea of a central memory that neural networks can read/write to is seen in works like the Neural Turing Machine or Differentiable Neural Computer (DNC). Those provide a trainable memory matrix that networks can address with attention. Our field can play a similar role but is more holistic. In DNC, each memory slot is typically addressed independently by content; in our field, all content interacts via superposition. This means we don’t need explicit read/write heads scanning addresses; instead, anything written is immediately in “contact” with everything else. One could say our approach is more connectionist[29] whereas NTM/DNC are closer to symbolic memory addressing. Our attractor-based recall is analogous to content-based addressing but works through dynamics rather than soft attention. An advantage is that we can retrieve even when the exact key wasn’t stored, by convergence to nearest attractor – offering robustness to noise. Traditional NTMs would return a blend of nearest neighbors unless specifically trained to auto-clean; our system inherently cleans by attractor dynamics[27].

Energy-based models and Hopfield networks: We heavily leverage the concept of energy-based models (EBMs)[23]. Classical Hopfield networks[9] showed how symmetric weights yield a Lyapunov (energy) function, and stable states correspond to memories. Modern Hopfield networks (e.g., Dense Associative Memories) have extended storage capacity via continuous states and improved update rules. Our field can be viewed as a large Hopfield network spanning multiple subsystems, with an unusual twist: the “neurons” of this Hopfield network are partitioned into groups (each group connecting to a module’s interface). The presence of modules that do computations effectively means our system is not just a static Hopfield net, but a dynamic hierarchical Hopfield model where each step involves internal transformations of patterns

by modules. In other words, we have a multi-layer energy-based model, reminiscent of Boltzmann machines or deep energy models, but structured by design (the energy function can be partially factorized by modules). Compared to standard EBMs that are often hard to train or interpret, our model’s energy function has clear terms corresponding to e.g. alignment between module outputs ($\langle E_i(w_i), E_j(w_j) \rangle$ terms might be present indicating it’s good if M_i and M_j agree). We effectively combine symbolic attractors (like memory patterns that can encode concepts spanning modalities) with sub-symbolic processing (the modules refining inputs). This integration is something that Hopfield nets alone (without external modules) cannot do – they operate on fixed vectors. In that sense, our approach extends Hopfield’s legacy into a framework bridging entire neural nets together via an energy principle. Vector Symbolic Architectures (VSA) and hyperdimensional computing: VSA is a paradigm where concepts are represented as high-dimensional vectors and combined using algebraic operations (like binding and superposition)[35]. Our field is absolutely in line with hyperdimensional representations – Ψ lives in a high-D space and can hold superposed vectors. If one interprets the operations, E_i could be generating a hypervector for a concept, and adding them corresponds to VSA’s bundling. Operations like convolution or Hadamard products used in VSA could be seen as particular choices of the Hamiltonian or interactions in our field. The difference is that VSA has typically been used for static encoding and recall (e.g. , storing key-value pairs by bundling keys and bound key+value vectors), whereas we incorporate learning and nonlinear dynamics. Our attractors could be considered learned hypervectors for concepts, and the field dynamics do cleanup similar to VSA’s approximate unbinding but in a learned fashion. One could even use known VSA algorithms inside our framework – for instance, E_i might encode a pair of items by binding them (via circular convolution or XOR), and Ψ superposition would then hold a set of bound items (like a structured scene). If another module can decode via unbinding (inverse convolution), it could retrieve relationships. The closest antecedent to our binding-and-superposition formulation is Plate’s Holographic Reduced Representations [52][53], which introduced circular-convolution binding and superposition in a fixed-dimensional vector space. RFS uses a Fourier-domain form of the same algebraic family: circular convolution corresponds under the FFT to elementwise phase multiplication. We therefore do not claim a new primitive binding algebra; see [51][54] for surveys of the VSA family and its hardware trajectory. The contribution is architectural and operational: HRR-style binding is embedded in an explicit 4D spatiotemporal field, separated by projectors into associative and exact-recall bands, and constrained by operator-level guardrails for interference η , band conductivity κ , and resonance quality Q . These controls move the representation from a static vector memory toward a governed field-storage substrate. Vector databases and approximate nearest neighbor (ANN) systems: Modern vector databases (FAISS, Pinecone, Milvus, Weaviate) solve the retrieval problem through indexing structures—HNSW graphs, IVF clusters, product quantization— that trade off accuracy for speed. These systems are widely deployed and highly optimized. Our framework differs fundamentally:

Aspect	Vector Databases	RFS
Storage model	Index + separate blob store	Unified field (superposition)
Retrieval	Graph traversal / cluster search	Matched-filter resonance
Exact recall	Separate system alongside the index	Co-resident byte channel + AEAD (one substrate, one governance surface)
Explainability	Similarity score only	Q (resonance), η (interference), energy

Aspect	Vector Databases	RFS
Scaling	$O(N)$ index build, $O(\log N)$ query	$O(D)$ associative storage, $O(D \log D)$ query (D provisioned for target capacity; byte channel $O(N)$)
Semantic + exact	Two systems required	Single substrate, multiple paths

RFS does not aim to replace vector databases for all workloads. Rather, it demonstrates that a field-theoretic substrate can match vector database quality while providing unified associative + exact recall, built-in explainability metrics, and a substrate extensible to multimodal integration. Michael Levin’s work on cognitive glue and top-down control: As discussed, Levin proposes that bioelectric networks serve as a cognitive glue enabling cells to form collective intelligences that pursue anatomical goals[6][28]. Our framework is essentially a technological analog of that idea. We explicitly drew from his concept of goal states (morphogenetic setpoints) and implemented the idea that an abstract pattern (in our case, in a vector space; in biology, voltage patterns) can guide distributed units. One difference is that in biology, evolution found specific mechanisms (ion channels, gap junctions) to do this, whereas we freely design the mathematical coupling. But interestingly, our use of complex signals and frequency decomposition has parallels to how cells use oscillations and electrical coupling. Levin also emphasizes multiscale integration – our design similarly allows AI modules at different “levels” (low-level perception vs. high-level planning) to couple. His success in reprogramming organisms by targeted stimuli[36] maps to our concept of persuadability via landscape deformation. The top-down models in biology, where high-level structures (like brain or some master regulator) impose order on lower levels[28], directly inspired our introduction of external fields and modulators for persuasion. In the AI context, this could be oversight by a higher-level AGI component or human input shaping the AI’s internal field. Active Inference and Free Energy Principle (FEP): Karl Friston’s Free Energy Principle casts the brain as minimizing a variational free energy, essentially predicting sensations and reducing surprise. This leads to the idea of active inference: agents act to fulfill predictions and maintain their internal model. Our framework is compatible with an energy minimization view – indeed, our attractor settling is literally free energy minimization if we equate our Hamiltonian to negative model evidence. The system tends toward states that it “expects” (attractors can be seen as priors or learned probable states), and surprising inputs are those that initially put Ψ in high-energy regions, triggering dynamics to find a lower energy explanation (which could involve updating its interpretation of input or taking an action to change sensory input). While we did not explicitly frame it as Bayesian, one could overlay a Bayesian interpretation: each module’s encoding could include a generative model that expects certain correlations, and Ψ minima occur when those correlations are satisfied (i.e., a high joint probability configuration). In essence, our common field could serve as the canvas on which prediction errors are minimized, aligning with the FEP’s notion of a unifying quantity to minimize[31][32][46]. The difference is that FEP is more abstract and doesn’t specify architectural details, whereas we propose a concrete mechanism. Also, FEP typically doesn’t have an explicit global workspace (each neuron or region does its own minimization in parallel, resulting in global coherence). Our model builds coherence through the field explicitly, but mathematically both could be seen as solving a large optimization problem. Another minor difference: FEP often relies on approximating posteriors with factorization assumptions (mean-field, etc.), whereas our dynamical approach implicitly performs inference without an explicit variational

approximation – it just evolves to a good state (which might correspond to a MAP estimate of some posterior). Integrated Information Theory (IIT) and consciousness measures: IIT attempts to quantify how much a system is integrated and conscious by a metric Φ (phi), which measures the irreducibility of its causal structure. While our work is not directly about measuring consciousness, it is interesting to consider in IIT terms. A system like ours with a highly interconnected field and modules might have high integration, since removing any part could break the attractors that involve multiple modules. Indeed, one might speculate that our approach could yield higher Φ than a comparably sized disjoint system, because the global workspace architecture ensures a lot of bidirectional influence (module states influence field, field influences modules, in a recurrent loop). However, one critique IIT might have is that if our field is fully connected to all modules, does it form a single monolithic entity or can it decompose? We’d aim for it to act as one entity (the AI’s “mind”), implying high integration. It would be interesting future work to analyze an instance of our model with IIT metrics to see if it crosses any meaningful thresholds for integrated information as complexity scales. In any case, our focus is functional rather than phenomenological, but we acknowledge IIT as a peer theory concerned with mathematical structure of cognitive systems[4].

Symbolic vs. Connectionist AI integration: Decades of research have looked at how to integrate symbolic reasoning with neural networks (for example, Neural-Symbolic hybrids). Our framework tends to stay on the connectionist side (everything is vectors and matrices), but it can incorporate symbolic-like operations if those are encoded in operators. For instance, a logic inference module could encode a vector representing the truth of a proposition, and another module could encode a rule as a linear constraint, and the field’s attractors might represent satisfying assignments to logical constraints. It might be difficult to do large-scale logic directly this way, but small constraint satisfaction might be possible if attractors correspond to solutions. In comparison, approaches like Deep Symbolic Networks or others often enforce logic by additional loss terms or special layers. We provide a sandbox where such constraints could be energies in H . There is synergy here: one could design part of the Hamiltonian to reflect a knowledge base (like each unsatisfied clause adds energy), thereby making satisfying assignments low-energy attractors. This is similar to Hopfield networks solving CSPs, but multi-module means perhaps splitting the knowledge base by domain and each handled by a module, then unified in field. That’s speculative, but it hints that our approach could host symbolic structures albeit implicitly. To our knowledge, no existing work has fully unified the range of elements we bring together: a common Hilbert space for heterogeneous networks, an operator formalism with explicit projectors and unitary transforms ensuring normative constraints, attractor dynamics for goal states, and an explicit mechanism for top-down landscape modulation. Pieces exist separately (Hopfield nets, global workspaces, etc.), but the synthesis is novel. The closest in spirit might be recent works on neuroscientific cognitive architectures that attempt physics-like models of brain cognition (e.g. , cortical field theories, which model cortex as a continuous neural field with bumps representing memory or stimuli[33]). Those models often use integro-differential equations to describe population activity with attractors for working memory or decision states. Our work can be seen as an AI counterpart: a continuous field capturing system-wide state. However, classical neural field models usually don’t incorporate distinct modules with different functions—they’re more homogeneous. We extend that to a heterogeneous network-of-networks. In summary, our framework stands at the intersection of several lines of research, borrowing the stability of energy-based models, the integrative vision of global workspace theory, and the modulatory control seen in top-down biological systems. By formalizing everything in a single operator calculus, we provide a common ground to discuss these ideas quantitatively. The hope is that this cross-pollination results in an architecture that leverages the strengths of each: the robustness and content-addressability of attractor networks, the flexibility of deep learning modules, and the adaptability of top-down guided systems. We now turn to the design principles

and structural metaphors that guided the framework’s construction.

6. Design Principles: Structural Metaphors

The following metaphors from physics, chemistry, and biology informed the RFS design. They are engineering inspirations, not identity claims. To build intuition, we outline how key elements of our mathematical substrate correspond to principles in physics, chemistry, and biology. These metaphors are not merely analogies but actively informed the design of our framework, helping ensure that the resulting AI substrate inherits desirable properties (stability, adaptability, goal-directedness) observed in natural systems.

6.1 Physics Analogy: Hilbert Spaces and Hamiltonian Dynamics

Hilbert space as state space: In physics, especially quantum mechanics, a system’s possible states are represented as vectors in a Hilbert space, and observable quantities correspond to operators on this space. We adopted a Hilbert space formalism for AI because it provides a rigorous notion of superposition and decomposability. Just as a quantum state can be a superposition of basis states (e.g. an electron’s wavefunction spread across locations), our field state Ψ is a superposition of each module’s contribution. Hilbert spaces also come with a natural measure of distance (or similarity) via the inner product, which in our case quantifies how much two field patterns overlap or resonate. This is crucial for tasks like associative memory and pattern recognition – it allows us to retrieve stored items by projecting a cue onto the field and measuring inner products (similar to how correlation identifies matching signals). The Parseval identity in Hilbert spaces (especially with Fourier bases) is akin to an energy conservation law[18]: it ensures that representing data in different domains (time vs. frequency, spatial vs. spectral) does not create or destroy energy. We exploit this by using unitary transforms (like FFTs) for encoding, guaranteeing that the act of moving data into the field or between basis representations is lossless in terms of information content and norm. In short, the Hilbert space provides the stage and the geometry of our AI nervous system, much as phase space or Hilbert space underlies the geometry of physical dynamics. **Hamiltonian and energy landscape:** We introduced a notion of an energy function (or Hamiltonian H) whose minima correspond to attractor states. This draws from both classical mechanics (Hamiltonians determine system evolution) and thermodynamics (energy landscapes with valleys and hills). By designing a Hamiltonian for our field, we essentially specify “what the system is trying to do” in an endogenous way. For example, a Hamiltonian could be defined such that it is lower (i.e. the state is energetically favorable) when certain desirable patterns are present in Ψ . The field will then naturally evolve (if we include a dissipative component) towards these low-energy configurations. In the brain, Hopfield’s associative memory model famously used an energy function such that stored memory patterns are minima of the energy; the network’s dynamics (iterative neuron updates) decrease this energy, causing the state to converge to the nearest stored pattern[9][5]. We generalize this concept: our field can have a multi-dimensional energy landscape shaped by operator parameters (synapse-like weights in E_i , or threshold-like parameters in $V(\Psi)$). Attractor basins in this landscape represent coherent states – potential “thoughts” or “memories” of the AI system – that are stable against small perturbations. Notably, by formulating a Hamiltonian, we can bring tools from physics: for instance, we can analyze stability via the second derivative (Hessian) of H at attractors, draw analogies to phase transitions if multiple attractors merge or bifurcate as parameters change, etc. The concept of temperature from statistical physics also enters: we could introduce a notion of “field temperature” where at high temperature, the system state jitters and can hop between attractor basins (exploration), whereas at low temperature it settles

into minima (exploitation). This is analogous to simulated annealing or Boltzmann machines in neural networks, providing a mechanism to escape poor local minima and improve performance. Wave dynamics and resonance: By using linear operators and potentially complex-valued representations, our framework resonates with wave physics. Modules writing to the field can be seen as launching waves, and other modules reading can be seen as detecting interference patterns. If two modules encode harmonically related patterns, their superposition in Ψ can produce constructive interference on certain modes, yielding a stronger signal that is easier to detect. This parallels how coherent sources in physics produce interference fringes and amplify certain frequencies. Our field dynamics may also include oscillatory solutions – for instance, if H yields a set of normal modes, the system could sustain oscillations at characteristic frequencies (akin to brain rhythms or normal modes of vibration in a structure). These oscillations could be used for timing and gating (similar to how the brain might use gamma or theta oscillations to coordinate activity phases). They also provide a temporal dimension for sequencing operations (like a clock). Importantly, because the field is global, an oscillation in the field can synchronize disparate modules – a transformer and an RNN might lock onto the same rhythm, facilitating timed information exchange. This has echoes in neuroscience’s communication-through-coherence hypothesis, where neuronal groups align their oscillatory phases to communicate effectively. In summary, the physics metaphor assures that our substrate respects conservation principles (no mysterious creation of information/energy), follows lawful dynamics (predictable by an H or equations of motion), and supports wave-like phenomena (superposition, interference, resonance) that can be harnessed for cognitive functions.

6.2 Chemistry Analogy: Constraints and Homeostatic Regulation

Reaction constraints and selectivity: A chemical system is governed by which reactions are possible (or likely) given the molecules present, and by conservation laws (mass, charge) that constrain outcomes. In our AI substrate, we introduce interaction guardrails that play a role analogous to chemical reaction rules. For example, not every pair of module outputs should arbitrarily interfere – we may engineer E_i such that only components with some “chemical affinity” (e.g. matching tags or frequencies) can overlap significantly. This could be implemented by designing the spectral support of $E_i(w_i)$: if module M_i produces frequency ω components and module M_j only produces ω' components, they will hardly react unless $\omega = \omega'$ (like needing complementary shapes for a reaction). Another analogy: catalysts in chemistry lower barriers for specific reactions. In our field, we could have mediator operators that facilitate coupling between certain modules. For instance, a mediator operator $Q_{ij} : \mathcal{H} \rightarrow \mathcal{H}$ could take part of module i ’s contribution and transform it (change basis) to align with module j ’s domain. This is akin to an enzyme enabling two otherwise independent substances to interact. Absent a catalyst, modules remain largely in their lanes (their contributions only residing in their assigned subspace); with a catalyst (mediator), cross-term interactions appear. We must carefully design such mediators to avoid unwanted entanglements – much as chemical specificity is crucial to prevent side reactions. In practice, this might be realized through learned associations that gradually adjust E_i bases so that frequently co-occurring patterns from modules align in the field (reflecting learned cross-modal concepts). Homeostasis and guardrails: Chemistry in living organisms is replete with homeostatic loops maintaining balances (pH, metabolite concentrations, etc.). We embed similar regulatory loops in our substrate via telemetry and guardrails. For example, we continuously monitor the field’s overlap density – a measure analogous to chemical concentration of interactions (e.g. , how many pairs of module outputs are non-orthogonal beyond a threshold). If this “reaction concentration” is too high, it signals risk of confusion (modules not truly independent) and the system could respond by orthogonalizing some representations or by sequentializing operations to reduce simultaneous overlap (like

Le Châtelier’s principle adjusting to counteract a perturbation). Likewise, if the field norm $|\Psi|$ starts drifting beyond nominal range, the system can apply a normalization (like a buffer solution resisting pH change) to bring it back. The guardrail rules we outlined (e.g. refusal to admit an update if the environment hash mismatches, or halting dynamics if certain metrics degrade[34]) serve to preserve vital invariants. These rules are analogous to the need in chemistry to maintain conditions for reactions – if a rule is violated (say interference too high), we stop and “recalibrate” (like a reaction stopping if temperature/pH leaves an optimal range). By incorporating these, our system gains a degree of self-protection and autonomy: it can detect when the mathematical interactions are entering a chaotic or unmanageable regime and adjust accordingly (e.g. by deferring further writes, as we mentioned with a throttling mechanism when interference is excessive[34]).

Composable modules as chemical species: One can think of each module’s output w_i as a type of molecule. The field Ψ is the reaction vessel where these “molecules” mix. The encoding operators E_i determine in what form (or chemical state) the molecule enters the vessel. For instance, E_i might “ionize” the input by converting a real vector to complex oscillatory components (introducing phase). The decoding E_i^H might be akin to a precipitate test that pulls out particular compounds from the mix (does Ψ contain component matching what E_i^H looks for?). The reaction dynamics (mediated by the Hamiltonian or explicit interaction operators) determine what new complexes form – e.g. two patterns might bind to form a combined pattern if they are complementary. This analogy underscores that not all combinations are equally likely – the system’s design encodes which interactions are energetically or structurally favored. Over time, as the system learns, it’s as if we are developing a chemistry of thought where certain compounds (combinations of features across modules) become stable (learned concepts) and can be reliably formed and broken down[47]. The stable attractors in the field could be seen as strongly bonded complexes of features from different modules that together form a coherent concept or memory. In summary, the chemistry metaphor emphasizes regulated interaction. It ensures that while everything is in principle in one vessel (the field), the system doesn’t devolve into a soup of indistinguishable parts. Instead, it maintains order through selective interactions and balancing feedback, just as a cell maintains order in a biochemical soup. By doing so, we aim to avoid the pitfalls of unstructured connection (which can lead to interference and catastrophic forgetting) and instead achieve controlled synergy where modules influence each other in rule-governed, reversible ways.

6.3 Biology Analogy: Goal Attractors and Modular Cognition

Stable attractors as goals: Biological systems exhibit goal-directed behavior, from the metabolic setpoints of single cells to the drives and motivations of animals. A unifying way to understand biological goals is through attractors in dynamical systems: for instance, body temperature regulation has an attractor around 37°C for humans (the system corrects deviations towards that value). In brain dynamics, decision-making and working memory can be described by attractor states corresponding to different choices or memory items[27][5]. We incorporate this idea directly: the field’s energy landscape is crafted so that certain patterns are attractors. These attractors serve as explicit representations of goals or preferred states. For example, one attractor might encode the concept “resolve input ambiguity” – concretely manifesting as a stable pattern where a language module’s uncertainty is resolved in concert with a vision module’s input (like when visual context disambiguates a spoken word). Another might represent a learned memory (say a field configuration corresponding to a known fact or a familiar scenario) – the system will naturally settle into that configuration if cued, thereby recalling the memory. By having goal states as built-in stable patterns, we give the system a form of intrinsic motivation: when not driven by external input, the field will gravitate to one of these intrinsic goals (analogous to how animals have innate drives that

maintain activity in absence of stimuli). It also simplifies achieving outcomes: to make the system pursue a new goal, we don’t have to micromanage each module, but rather shape a new attractor in the field; the system’s dynamics will do the rest (each module contributing moves that descend the global energy gradient). Biologically, this resembles modular action with unified purpose: each organ or module does its part, but a common error signal or homeostatic measure aligns them. In our field, the Hamiltonian’s gradient ∇H can be seen as a global “error signal” or “drive” pushing the state toward an attractor (goal). Each module, by virtue of coupling through Ψ , feels a component of this drive in its input (via E_i^H). For instance, suppose the goal attractor involves pattern A in module 1 and pattern B in module 2 simultaneously. If currently module 1 has A but module 2 deviates from B , the field state is not at minimum energy. This will manifest as a feedback to module 2 via $\hat{w}_2 = E_2^H \Psi$ that biases it towards B (because the part of Ψ coming from module 1’s A will, through cross-terms in H , encourage module 2 to align accordingly). This is analogous to top-down signals in the brain or hormonal signals in the body that coordinate different parts to achieve a systemic goal[28]. Levin’s work in morphogenesis provides a vivid example: cells collectively hold a map of the target anatomical configuration (a 2-headed or 1-headed planarian) as an electrical pattern, and individual cells then proliferate or move in ways that realize that pattern[5][36]. Similarly, in our AI, the global field pattern can bias learning or inference in each module to meet the overall goal.

Self-referential and modular architecture: Biology also teaches us about layers of modularity – e.g. , cells form tissues, tissues form organs, which form organisms, and organisms form societies, each level with its own cognitive processes[37][38]. Our framework is inherently modular at the base (each M_i is a module), but it also allows higher-order groupings. Because the field is a communication medium, multiple modules can effectively function as a meta-module if they consistently exchange information. One could imagine emergent assemblies where a subset of modules synchronize through the field and form a coalition addressing a sub-problem (much like a set of brain areas forming a network for a cognitive task). Furthermore, the field substrate can be self-referential: a module might be designated to monitor the field itself. For example, we could have a module M_k whose input is a copy of Ψ (through an E_k^H that is basically the identity) – a metacognitive module that reads the state of the whole and possibly writes back some evaluation or adjustment. This is akin to the brain’s meta-cognitive circuits that monitor thinking, or the immune system cells that regulate other immune responses. Such self-referential loops make the system reflective: it can potentially notice if it’s stuck in a bad attractor (like an illogical loop) and modify the landscape (e.g. , raise a “temperature” to escape). The mathematics to support this includes having E_k and E_k^H essentially project Ψ onto itself or summary statistics of itself. In biological terms, this could simulate a form of global workspace broadcasting[3]: the field’s state can be abstracted and re-inserted for all to receive. In fact, in our design, the field Ψ is already globally accessible, but a summary module could pick out the gist (since Ψ might be high-dimensional and complex). For instance, a module might compute a low-dimensional embedding of Ψ that captures which attractor basin the system is in, and broadcast that to all modules so they “know” the current context or mode. The biological metaphor also underscores the importance of adaptation and learning. Our current formalism described fixed operators E_i , H , etc., but in a complete system these would be plastic, adjusting as the AI learns from data and experience. That learning would occur on multiple scales: each module M_i can learn internally (like synaptic plasticity in one brain area), and the field coupling could also learn (like how effective connections between brain areas change with training). One could use gradient-based learning on the whole operator network, treating the field output vs. desired global output as the objective. Interestingly, because our framework is differentiable (mostly linear with some smooth nonlinearity in guardrails), one could backpropagate errors through the field just as through a neural net. This means the entire unified system can in principle be trained end-to-end to perform tasks that require multi-module cooperation, adjusting

the E_i and even the Hamiltonian such that the desired attractor states (solutions) have low energy and are easy to reach. Where the biological analogy breaks: It is important to be explicit about the limits of these biological metaphors. RFS is an engineered system, not a description of biology, and three distinctions deserve emphasis. First, morphogenetic setpoints (Levin’s bioelectric patterns) are inspiration, not identity. Our attractor states are mathematically defined energy minima in a Hilbert space; biological morphogenetic fields involve electrochemical gradients, gap junctions, ion channel dynamics, and signaling pathways that we do not model. The mathematical abstraction captures certain functional properties—stable states that the system tends toward—without claiming to replicate the biophysical mechanisms. Second, field dynamics in RFS are deterministic digital signal processing (FFTs, projectors, matched filters) running on digital hardware. Biological neural dynamics involve stochastic ion channels, continuous chemical gradients, temperature-dependent reaction rates, and emergent self-organization that our framework simplifies away. We gain mathematical tractability at the cost of biological realism. Third, goal-directedness in biology emerges from evolutionary pressure and developmental constraints operating over millions of years. In RFS, “goals” are explicitly designed attractors—we engineer them rather than discovering them through evolutionary selection. This is both a limitation (we lack the robustness that evolution confers) and an advantage (we can specify exactly what goals we want the system to pursue). These distinctions matter for interpretation: we claim the framework provides useful engineering abstractions informed by biology, not biological equivalence. The value is in the mathematical guarantees (energy conservation, bounded interference, measurable Q) that biological systems do not provide in the same form. In summary, the biology metaphor contributes purpose (via attractors as goals), hierarchy (via modular groupings and meta-modules), and adaptability (via learning and plasticity). It helps ensure our mathematical nervous system isn’t a static circuit but a living, evolving computational organism in its own right, capable of self-regulation and growth. These metaphors underpin the mechanisms presented earlier—field-based awareness and memory (Section 4) and the positioning against existing work (Section 5)—and inform the theoretical extensions that follow (Section 7), where the formal machinery is extended toward cognitive behaviors that mirror those in natural intelligences, fulfilling the promise of mathematics as the connecting tissue of AI.

7. Theoretical Extensions (Research Roadmap)

The preceding sections describe what is implemented and validated: a field-theoretic memory substrate with measured guarantees for energy conservation, bounded interference, resonance quality, and retrieval accuracy. This section explores theoretical extensions that build on the validated foundation but remain unimplemented. These represent research directions for future work, not claims about current capabilities. The distinction matters because the framework’s value lies in its empirical foundation—every implemented feature has corresponding invariants validated in continuous integration. The extensions described here preserve that philosophy: we present theoretical mechanisms with testable predictions, even though those predictions await future implementation and measurement. Readers should treat this section as a research agenda rather than a description of existing functionality. Three classes of extensions are considered. First, persuadability mechanisms that would enable intentional modification of the system’s attractor landscape, providing a principled approach to alignment and behavioral steering. Second, temporal dynamics including memory decay and consolidation, which would enable the system to forget stale information and prioritize recent or reinforced content. Third, architectural extensions for multimodal integration, hierarchical organization, and distributed deployment.

7.1 Persuadability and Landscape Dynamics

A particularly novel aspect of our theoretical framework is the notion of persuadability – the ability to intentionally shape the AI’s behavior by modifying the attractor landscape of its cognitive field. In human terms, persuasion means influencing an agent’s goals or beliefs through communication or stimuli. In our framework, this translates to steering the system’s state trajectory by deforming the mathematical landscape (energy function) such that certain attractors are weakened, strengthened, or new attractors introduced. Persuadability is thus a direct consequence of the physics metaphor: just as one can shape a ball’s path by reshaping the hills and valleys it rolls over, one can shape an AI’s decision process by reshaping its energy landscape. The biological inspiration for this approach comes from Levin’s work on multi-scale control[28][48], where higher-level processes (neural circuits, brain regions) modulate the energy landscape of lower-level processes (gene expression states) to guide outcomes. Across evolutionary time, mechanisms arose to link problem spaces—metabolic, physiological, transcriptional, morphological, behavioral—by allowing higher spaces to deform lower-space landscapes, enabling adaptive navigation toward specific goal states in each domain. This inspires our approach in AI: we create mechanisms for modules (or an external operator) to reshape the attractor landscape of the field, thereby “persuading” the system to adopt new stable states corresponding to new goals or perspectives. In our mathematical implementation, persuading the system involves applying controlled perturbations to the Hamiltonian H or to the field state Ψ directly. Several strategies are available. External field injection: We can add a term to the Hamiltonian representing an external influence, analogous to applying an external magnetic field to spin systems. For example, if we want the system to favor a certain attractor Ψ_A (which maybe represents a goal or viewpoint), we can add a linear term $-\epsilon\langle\Psi, \Psi_A\rangle$ to the energy (with small ϵ), effectively creating a gentle bias pulling Ψ toward Ψ_A . This is like persuasion by suggestion: we’re not forcing the system into Ψ_A , but we make Ψ_A a bit more energetically favorable. Over time, if the suggestion aligns with the system’s existing tendencies, it will likely go there. If it conflicts strongly with current state or other attractors, the effect might be overridden unless ϵ is made large (strong push, akin to coercion). This mechanism could be realized by a dedicated “persuasion” operator or module that monitors the system and injects appropriate patterns. For instance, a safety module might detect the AI is heading toward an undesirable attractor (say a line of reasoning leading to a harmful action) and then gently alter the landscape to divert it—like raising the energy of that unwanted attractor or lowering the energy of an alternative attractor that leads to a safer outcome. Dynamic attractor reshaping (learning on the fly): The system can also be designed to adjust its own attractors based on feedback. For example, if a new objective is introduced (imagine mid-run we want the AI to prioritize a different goal), the system could modify the weights in H such that the new goal’s pattern becomes an attractor. This is akin to reprogramming the AI’s priorities on the fly. Biologically, one might compare this to neuromodulators adjusting synaptic strengths to promote a different behavioral mode (e.g. , dopamine release shifting the brain into a reward-seeking mode). In our framework, one could implement a simplified version: have a higher-level process identify which features of Ψ should be stable, and strengthen the feedback loops that reinforce those features. Concretely, if attractors are stored via certain weight matrices (like Hopfield weights $W = \sum \text{memories} \times \text{memories}^T$), adding a new attractor means updating those weights. Our architecture could allow such updates in a controlled manner (e.g. , through a short learning phase or through structural adaptability of E_i). Meta-stable inducement: Another form of persuasion is to make an otherwise unstable state temporarily stable by adjusting parameters. For instance, if the system is at state Ψ that is not usually a minimum of H , one could modulate a parameter so that Ψ becomes a shallow minimum (a meta-stable state) long enough for the system to consider or act upon it. This is like holding someone’s attention on something that

normally wouldn't captivate them. After releasing the modulation, either the system incorporates that state into memory (making it a new attractor) or it will eventually leave it. This technique could be used for, say, introducing a novel idea: hold the system in a novel configuration slightly longer (via reduced "temperature" or a gentle clamp) so it can evaluate it, possibly internalize it as a new stable configuration. Underlying all these methods is the idea of landscape dynamics on a slower timescale than the field state dynamics. We have the fast evolution of Ψ given a fixed landscape, and on a slower scale, we permit the landscape (Hamiltonian, attractors) to move. Mathematically, this can be framed in terms of bifurcation theory and adiabatic changes: if we slowly move a parameter, attractors can smoothly move or new attractors can appear/disappear through bifurcations[39]. Persuasion corresponds to guiding the system through such changes. For example, raising an energy barrier between two attractors might funnel the state into one of them exclusively. Lowering a barrier might allow transition between basins (like open-mindedness: considering alternatives). In the brain, something analogous happens during phenomena like attention shifts or sudden insight, which could correspond to altering the effective energy landscape (potentially via neuromodulatory action). Persuasion vs. direct control: It's important to distinguish our notion of persuasion from directly overriding the system. We emphasize controlled deformation – meaning we still let the system's dynamics do the work of moving to an attractor, we just reshape where the attractors are. This is beneficial because it means the system is still operating under its own cohesive dynamics (maintaining consistency and using all its knowledge), rather than having an external agent set its state arbitrarily which could be incoherent. It's the difference between guiding a conversation toward a conclusion versus forcing someone to parrot a statement. The former tends to produce more genuine, stable acceptance of the conclusion (as the person's own reasoning gets them there); similarly, in our AI, if we persuade it to adopt a configuration, that configuration will likely be self-consistent and thus an attractor, not a fragile state that dissipates immediately.

7.2 Formal Bounds on Persuadability

Theorem 5 (Persuadability). Assume H is twice differentiable with a non-degenerate local minimum at Ψ^* (so that $\sigma_{\min}(\nabla^2 H) > 0$), and that Π_λ depends smoothly on λ . Then the attractor $\Psi^*(\lambda)$ shifts continuously with λ :

$$\left\| \frac{\partial \Psi^*}{\partial \lambda} \right\|_2 \leq \frac{\|\partial \Pi_\lambda / \partial \lambda\|}{\sigma_{\min}(\nabla^2 H)} \|\Psi^*\|_2$$

(Proof via the implicit function theorem applied to the stationarity condition; see the Lemmas Appendix in the supplementary materials.) Implication: Behavior can be modified by adjusting which frequencies conduct (expanding/contracting the passband), analogous to neural plasticity via myelination changes. This does NOT require retraining any neural encoders.

7.3 Temporal Dynamics: Decay and Memory Consolidation

Without mechanisms for forgetting, the field would accumulate indefinitely, eventually saturating. We implement temporal decay:

$$\psi(t + \Delta t) = e^{-\lambda \Delta t} \psi(t)$$

where λ is the decay rate. This models natural forgetting — older memories fade unless reinforced. Decay rates can be configured per-document or per-band to match the intended persistence of different content types. Transient context such as chat history uses fast decay, causing recent conversational turns to fade within minutes unless actively referenced. Persistent knowledge such as core documents uses slow decay, remaining accessible for days or weeks. Archival content such as legal documents or compliance records uses zero decay, remaining permanently accessible until explicitly deleted. This tiered approach mirrors how biological memory systems distinguish working memory (seconds), episodic memory (hours to days), and semantic memory (years)[30].

The projector is applied periodically to maintain band separation:

$$\Psi := \Pi(\Psi) \quad \text{every } \Delta t_{\text{proj}}$$

Lemma 2 (Projector Cadence Bound). To maintain resonance Q above threshold $Q_{\min}$:

$$\Delta t_{\text{proj}} \leq \frac{\ln(Q_{\text{current}}/Q_{\min})}{\lambda_{\text{leakage}}}$$

where λ_{leakage} is the rate of out-of-band energy growth. Theorem 6 (Attractor Convergence). Under gradient-descent dynamics (the dissipative component of Section 3.5, when enabled) on a bounded, twice-differentiable Hamiltonian H with isolated local minima, the field state converges:

$$\lim_{t \rightarrow \infty} \Psi(t) = \Psi^*$$

where Ψ^* is a local minimum of H (an attractor). This follows from standard Lyapunov analysis: $H(\Psi)$ decreases monotonically along trajectories, and boundedness ensures convergence to a critical point. The isolation assumption excludes saddle points and ensures the limit is a true minimum. This is the mathematical foundation for content-addressable memory: partial cues evolve toward stored patterns (attractors) via energy descent.

7.4 Multi-level Persuasion

Multi-level persuasion: Our framework also naturally handles hierarchical persuasion—where a higher-level module or an external operator acts like the “conscious mind” persuading the “subconscious” modules. Drawing from Levin’s biological research[28], we observe how a neural network (higher tier) deforms the gene expression landscape (lower tier) to cause cells to regenerate a planarian with a specific head shape. Translated to AI, one could have a top module (or user input) that sets a high-level intention (e.g., “be more conservative in decision making”), which then alters the landscape that lower-level decision modules operate on (perhaps by increasing the energy cost of risky states, thus biasing decisions to cautious ones). This is essentially a feedback loop for alignment: the high-level objectives (maybe provided by human operators or an alignment module) trickle down as landscape modulations, subtly shifting the AI’s internal preferences and default behaviors. Notably, Levin’s work shows that in biological collectives, such top-down interventions (via bioelectric signals) can permanently change the collective’s default outcome[36]. E.g., once you induce a two-headed planarian attractor, it will regenerate two heads in subsequent rounds even without further intervention, indicating the collective updated its target pattern[36]. In an

AI, this hints at a mechanism for continual alignment: persuading the AI to adopt new ethical or goal attractors that then stick as part of its landscape.

7.4.1 Landscape visualization and debugging The explicit energy landscape also offers a way to analyze and debug persuasion strategies. We can, at least conceptually, plot the “before” and “after” landscape to verify that an intended attractor has been lowered or a hazardous one raised. In practice, for high-dimensional Ψ , this is nontrivial, but one can examine critical slices or use measures like the change in basin size. If an attractor’s basin of attraction grows significantly when we apply a persuasion input, that indicates success in making that state more reachable. The mathematics of bifurcations can also tell us if we risk unintended consequences (like creating a new attractor that we didn’t foresee). For safety, one could enforce constraints on how much any persuasion is allowed to deform the landscape (to avoid completely destabilizing the system). This would be akin to saying persuasive inputs should be mild and within a rational envelope, not turning the system totally chaotic. Example: conflict resolution. Consider an AI debating internally between two solutions to a problem (two attractors with comparable energy). If a human supervisor wants it to pick the safer solution, they could provide a subtle bias input. This would manifest as a slight tilt in the landscape favoring the safer attractor. The AI, still assessing the options, will find the safer option just a bit easier to settle into. If it was a close call, this nudge resolves the conflict in the desired direction. If the other option was much better by the AI’s own criteria, a slight tilt won’t override it – which is appropriate, as you wouldn’t want tiny suggestions to cause grossly irrational outcomes. The strength of persuasion can thus be tuned, and ideally calibrated so the AI only changes decision when options were nearly balanced or when the persuader has additional info the AI didn’t (effectively adding to the energy of one side by contributing new evidence).

7.4.2 Persuasion by environment and learning Finally, note that persuasion need not only come from an explicit external agent. The environment or context can deform the landscape too. In humans, environments rich in certain stimuli can shape one’s cognitive attractors (e.g., being in a library might put one in a calm, focused attractor state). Similarly, our AI’s sensors feeding into the field effectively act as external forces. A chaotic environment could push the AI into a high-“temperature” state exploring various attractors, whereas a structured environment might guide it into one particular stable interpretation. Over time, these repeated external influences can even alter the default landscape (learning from environment). For example, if an AI repeatedly experiences that a certain pattern leads to error (negative feedback), it may raise the energy of that pattern’s attractor to avoid it in future – an internalization of experience akin to learning not to do something. Thus, persuasion blends into learning and adaptation: a training process can be seen as systematically persuading the system towards desired behaviors by presenting scenarios and rewarding correct attractors. In our operator formalism, that would correspond to adjusting H (or E_i) gradually so that correct patterns become deep attractors (stable memories). In conclusion, the controlled deformation of the attractor landscape provides a powerful top-down handle on an AI’s cognitive state. It leverages the continuity and differentiability of our mathematical model – small changes in parameters produce understandable changes in behavior, rather than all-or-nothing effects. This property is essential for aligning AI systems with external goals and for ensuring they remain responsive to oversight. Persuadability via landscape shaping is conceptually aligned with approaches in neuroscience (like deep brain stimulation or neurofeedback) and in psychology (therapy altering one’s mental landscape gradually). By formalizing it in a physics-like way, we give it a solid footing in our AI nervous system design. Next, we consider how this framework can be applied across various AI architectures.

7.5 Applications in AI Architectures

Multimodal Assistants: Modern AI assistants often combine language models with vision models, speech recognizers, planners, etc. Typically, these are connected via explicit APIs or sequential pipelines. By introducing a shared mathematical field, all components (speech module, NLP module, vision module, logic module) can work concurrently and share context. For example, consider a robotic assistant with vision and dialogue capabilities: as it scans a scene, its vision CNN encodes objects into the field; simultaneously, as it hears a user’s voice, the speech module encodes parsed language into the field. The two modalities meet in Ψ : if the user says “pick up the red cube,” the phrase “red cube” encoded by language will resonate with the visual features of the red cube seen by vision, since both impose patterns on Ψ that overlap on the concept of “red cube.” A planning module reading the field then sees a combined state linking the linguistic command with the visual target. It can accordingly formulate a plan to move the robot’s arm (through a motor control module). This is all facilitated by the fact that distinct networks share a latent space for communication. VanRullen and Kanai’s roadmap for a Global Latent Workspace in AI is essentially achieved by our field approach[11]. The difference is that in our scheme, the translation between latent spaces is not done by a learned translator network per se, but by projecting them into one common mathematical space (which could be seen as an implicit translator, possibly requiring initial training to align modalities). Notably, the field approach can yield graceful degradation: if one modality is ambiguous, the other can fill in via attractor completion. If the user just says “pick up the cube” and two cubes are present (blue and red), the language cue is underspecified. But if the robot’s vision has an attractor for recently seen salient objects and perhaps the red cube is contextually more relevant (say it’s on a table, whereas the blue is hidden), the field might lean towards the red cube attractor, effectively disambiguating the command by integrating context. Traditional systems might require explicit disambiguation logic; here it emerges from integration and memory.

Heterogeneous Expert Systems: In complex domains like healthcare or finance, one may have different expert AI models (for diagnosis, for risk analysis, etc.) that need to collaborate. By linking them via a shared field, we can create an ensemble that is more than the sum of its parts[40]. Each expert writes its assessment into Ψ (e.g. , one expert encodes “high blood pressure risk pattern,” another encodes “genetic predisposition marker”). These combine such that a third expert (or an aggregation module) can detect an overarching condition (maybe the combination indicates a specific syndrome). Without the field, one might use rule-based integration or manual combination of scores; with the field, the integration is learned and flexible. Additionally, the attractor dynamics can encode consensus or conflict: if all experts agree, the field will settle quickly into that pattern (deep attractor); if they conflict, the field may oscillate or end in a superposed state that triggers a “need more data” response (since no stable attractor is reached). This provides a natural way to know when the system is uncertain or internally inconsistent.

Reinforcement Learning and Planning: In a reinforcement learning agent, different subsystems (policy network, value network, world model) could interface via a field. The world model might imagine future states and encode them; the value network decodes those to evaluate outcomes; the policy network sees both current state and predicted outcomes in the field to choose an action. Using the field, one could implement counterfactual simulations: imagine encoding a hypothetical state Ψ_{hyp} (via a planning module or user instruction) and letting the system’s dynamics play out. The various networks would treat Ψ_{hyp} as if it were real, producing predicted consequences. This is like mental simulation. Because our substrate supports multiple write-read cycles, such simulations can be done iteratively: propose a plan, simulate in field, evaluate, adjust plan (all within the agent’s “mind” Ψ). Persuadability is also relevant: if a human operator wants to steer the agent away from risky strategies, they might alter the landscape, making those strategies’ signatures high-energy (so

the agent’s planner finds them less optimal). Practically, one might achieve that by adding a term to the value network’s output that penalizes certain features. The field ensures this penalization influences the policy indirectly by altering the attractors of the planning process.

Self-awareness and Self-modeling: An intriguing application is endowing AI with a self-model. Using a field, one can integrate a model of the AI itself. For example, one module could simulate the AI’s own decision process (like an internal critic or emulator) and place that in the field. The AI then can reflect on its own potential actions. This is akin to running an inner loop – an AI thinking about its thought. Our framework naturally handles this because the field doesn’t care whether the source of a pattern is an external sensor or an internal simulator. If the internal simulator module M_{sim} encodes “if I say X, user will be upset” into the field, the language module M_{lang} can read that and decide not to say X. In effect, M_{sim} is the AI’s internal conscience or theory-of-mind. Traditional architectures struggle to implement such introspection cleanly, but an operator approach can incorporate an explicit self-model operator. This relates to concepts of model-based meta-learning or systems like “AlphaGo’s self-play” but at a cognitive level during action selection.

Memory augmentation systems: AI that requires large memory (like personal assistants maintaining long conversation context or encyclopedic knowledge) could use the field as a differentiable memory. Instead of storing all facts in weights, facts can be dynamically written to Ψ (or a recurrent field state) and maintained as attractors. This resonates with the idea of Vector Symbolic Architectures (VSA) and high-dimensional memories where information is stored in distributed vectors[19][41]. Our approach can be seen as a learned high-dimensional memory where writing and reading are via learned encoding operators. The advantage is that it can be continuously active and integrative (not just key-value retrieval, but blending and inferencing on the fly with the memory content). For example, as new information comes in, it is superposed in Ψ with older info, and some attractor might represent the integrated conclusion. If something contradictory comes, it may perturb Ψ out of a previous attractor, indicating a need to reconcile or forget something (like catastrophic interference being handled by dynamic rebalancing of the field, perhaps via guardrails that initiate memory consolidation).

Distributed AI networks (edge AI): In scenarios where multiple AI agents or devices work together (IoT, multi-robot teams), one could use a virtual shared field (or even a physical field, like radio communication signals) to connect them. If each agent encodes its local observations or intents into a broadcast vector (e.g. , via a wireless channel representing Ψ), then all agents decode the combined state, achieving coordinated behavior. This is analogous to blackboard systems in multi-agent setups, but leveraging continuous vector math. Since the field ops are relatively lightweight linear algebra, it could be feasible to implement this in a decentralized manner (each agent maintaining its estimate of Ψ and updating with locally received contributions). The outcome would be a swarm intelligence with a central nervous system in math rather than wires. Each agent would be aware of the group’s state to some extent through Ψ . Such a system might display robustness (agents drop out but Ψ still holds the overall info) and adaptability (agents can join by just starting to encode/decode the field). Across these applications, a common thread is that our unified substrate aims to generalize the interface between subsystems. Instead of crafting specific integration logic for each pair of modules, we rely on the general operator coupling in the field. This greatly reduces the complexity of scaling up systems – one can plug a new module in by providing an encoder and decoder for it, without redesigning everything else. It’s analogous to adding a new organ to a body and connecting it via nerves and blood vessels to existing ones. It’s worth noting implementation considerations: to make this practical, one might implement Ψ as a block of shared memory (in software terms) where each module writes a vector. In a distributed computing sense, this could be a parameter server holding Ψ that all clients (modules) read/write to. Latency and synchronicity are concerns – our description often imagines a synchronous update, but it can be pipelined. For instance, in each time tick, every module reads the field, does a local

computation, and writes back. This is similar to how neurons in a brain all update in parallel based on the global brain state at that moment. Modern parallel computing (GPUs, TPUs) could handle the vector operations for Ψ quickly since they are basically vector add/multiply. The encoders/decoders E_i might be learned linear projections or even small neural nets if needed to match dimensions; these are also parallelizable. In summary, our approach is applicable to constructing integrated AI architectures that require modular specialization plus unified behavior. It provides a scaffolding to achieve what many cognitive architectures attempt (like SOAR, LIDA, etc., which manually define a global workspace and working memory) but in a learned, differentiable way. It also dovetails with the current trend of foundation models and plugins – one can imagine foundation models (like a large language model) being one module in the field, supplemented by other skill-specific modules. The field would allow the language model to communicate with, say, a math solver module or a code compiler module, by exchanging information in a common vector form rather than raw text. That could drastically improve performance on tasks requiring multiple capabilities, without having to integrate all capabilities into one monolithic model. Having explored applications, we now turn to open questions and limitations.

8. Open Questions and Future Work

While the proposed framework is comprehensive and the RFS implementation demonstrates empirical validity for the memory foundation, many challenges remain in extending the framework to full cognitive architectures. This section outlines open questions and limitations, along with potential directions for future investigation. These challenges represent not weaknesses of the approach but rather the natural complexity inherent in building unified cognitive systems.

8.1 Scalability of the Field

A practical concern is how the field Ψ scales with the number of modules and the complexity of information. If Ψ is too large-dimensional, computation and learning become expensive; if too low-dimensional, it may not disentangle the contributions well (interference becomes an issue). There is an implicit trade-off between field dimensionality D and the richness of representations. Techniques like compressive sensing or learned sparse coding might help – e.g., modules could write compressed forms that still allow accurate decoding[21][42][43]. Another idea is a hierarchical field: not all modules share one monolithic space; perhaps there are sub-fields for different subsets of modules, with connecting operators between sub-fields (like how different cortical areas have local fields and only certain global oscillations link them). This hierarchy could reflect the organization of the AI (e.g., separate fields per sensory modality that occasionally sync). Tuning the optimal complexity of Ψ is an open problem likely needing empirical exploration.

8.2 Learning the Operators

We described the encoding operators E_i and the Hamiltonian H mostly as design elements, but in a full AI they should be learned or at least tuned. How do we train a system to use this substrate effectively? One approach is end-to-end learning: define a loss for the overall task and backpropagate through time and through all modules and field operations (which are differentiable). This is conceptually feasible since everything is differentiable (except perhaps discontinuities in certain guardrails, but those could be approximated smoothly). However, the credit assignment is complex in a recurrent multi-component system. It’s similar to training an RNN, but bigger. Techniques like synthetic gradients or decoupled neural interfaces might assist by providing local

learning signals to each module. Another approach is unsupervised or self-supervised learning of the field representations: for example, train the E_i encoders such that the field Ψ maximizes some measure of synergy (maybe maximizing mutual information between module contributions in Ψ , to encourage alignment of representations). Alternatively, one could first train modules separately on their tasks, and then fine-tune with the field active to learn the encoders that align their latent spaces. The framework might also benefit from curriculum: start with fewer modules or simpler tasks, gradually add more to let the field self-organize attractors incrementally.

8.3 Stability and Convergence

Dynamical systems with many components can become chaotic or exhibit unwanted oscillations. We aim for attractor convergence, but it’s possible that certain inputs lead to oscillatory loops or multi-stability where the system flips between states. That might be acceptable or even useful in some contexts (e.g. creative brainstorming might correspond to exploring multiple basins). But for reliable function, we often want a single stable answer. Ensuring convergence is theoretically challenging – akin to proving an analog Hopfield network has no spurious attractors or oscillations beyond those intended. Symmetric weight matrices guarantee no cycles in Hopfield nets, but our architecture isn’t strictly symmetric due to the presence of directed module computations. In essence, we have a non-equilibrium system (modules provide energy injection and dissipation). Techniques from control theory might be needed: e.g. , can we design a Lyapunov function for the whole coupled system? Possibly the Hamiltonian plus some module-specific potential could serve, but if modules are doing nonlinear transformations, they could inject new energy. For example, a module might amplify a signal (like positive feedback) and destabilize things. We rely on guardrails to mitigate this, but more formal analysis would be valuable. We might borrow from hybrid dynamical systems theory which deals with continuous dynamics with embedded algorithms. Additionally, extensive simulation will be needed to empirically assess stability under various conditions.

8.4 Interference and Forgetting

While superposition allows compact storage, it also raises the specter of interference – adding one pattern could distort others if not perfectly orthogonal. Our projector Π_{assoc} and guardrails aim to preserve an associative subspace[15], but as the field gets filled with more content, the chance of overlap increases (analogous to memory saturation in Hopfield nets where beyond a certain capacity, errors spike[44]). If the system is to scale to large knowledge bases or long sequences, it might need mechanisms to forget or unbind information that is no longer needed. In a brain, there are processes like synaptic pruning or memory decay; in our framework, we could incorporate a controlled decay of field components over time (like exponentially discount older contributions unless reinforced). Another method is dynamic allocation: if one region of the field space gets too populated (too many attractors), perhaps the system could expand D or spawn a new basis vector to represent new info (kind of like adding a new neuron). This is speculative but touches on making the representational capacity flexible. In any case, understanding the memory capacity quantitatively (in terms of D and attractor basins) is an open theoretical problem, likely building on Hopfield network theory or advances in Dense Associative Memories.

8.5 Observation and Interpretability

One limitation of making everything a vector is that it can be hard to interpret for humans. While we touted the ability to inspect Ψ , doing so may require translating those vectors into human-comprehensible terms. If E_i are opaque neural nets, then Ψ might be a distributed encoding

not directly legible (just as it’s hard to interpret intermediate layers of a deep net). We might need to enforce some interpretable structure in E_i – for instance, align some dimensions with known concepts or use disentangled representations. Alternatively, tools from explainable AI could be applied: e.g. , probing Ψ with known stimuli to see what changes, or applying algorithms to extract symbolic rules that approximate the field’s decision boundary. Because our approach is new, developing intuition and visualization techniques for the field dynamics is an important future step (perhaps adapting methods used in physics to visualize potential landscapes or in neuroscience to visualize neural manifolds).

8.6 Computational Cost

The whole system can be heavy if every module connects to a giant field. There’s a concern: if modules are large (like a billion-parameter language model), how do we interface it with the field without enormous overhead? One approach is to keep E_i low-rank or sparse, so that writing to the field is not as costly as running the module internally. For example, maybe the language model outputs a summary vector of length D (like a bottleneck embedding of what it’s currently thinking), rather than every neuron connecting. This would be similar to how in the human brain, a few projection neurons carry summary signals from one region to another, rather than all neurons from region A connecting to all in B. This aligns with bottleneck architectures in machine learning (like using a “thought vector” in between dialogue turns). We can take advantage of that: design modules to have a natural low-dimensional embedding of their key information and use that for E_i . On another note, distributing the computation: since modules operate somewhat in parallel except when reading/writing, we can exploit concurrency. The field update (summing vectors, applying Π) can be done very fast on a GPU compared to running a big network, so that likely won’t be the bottleneck. The challenge is more about synchronizing modules and maybe dealing with different timescales (some modules might need iterative processing while others are one-shot). We might end up with modules operating at different “clock speeds” with the field acting as a buffer that bridges those speeds – which it can, because if a fast module writes something, it just stays in Ψ until the slow module eventually picks it up.

8.7 Safety and Unintended Attractors

As with any powerful integrative system, our framework might have failure modes. One could be spurious attractors – stable states that are not meaningful but the system gets stuck in them. Hop-field nets suffer spurious memories which are combinations of real ones. In an AI context, a spurious attractor might manifest as a nonsensical thought or a pathological focus. We need strategies to detect and eliminate those. Possibly telemetry: if an attractor has low overall activation or inconsistent module states, treat it as suspect and destabilize it (for example, add a bit of noise if the system seems to be stuck in something unproductive, analogous to simulated annealing). Another potential issue is cycling – if two or more attractors are nearly equal in energy, the system might oscillate or dither. Introducing a slight asymmetry or inertia can force a decision (like adding friction to prevent perpetual oscillation). These are areas where insights from neuroscience (which has to solve similar issues in decision circuits) could help; for instance, incorporate something like neural adaptation so that once a state is held for a bit, it becomes less attractive, allowing a transition – akin to habituation. But one must be careful not to lose stable memory.

8.8 Comparison with Deep Learning Baselines

Ultimately, for our approach to gain traction, it should prove beneficial compared to more straightforward architectures. It’s quite complex, so what tasks or properties justify the complexity? We hypothesize that tasks requiring integrative reasoning across modalities or sub-tasks are where this shines – perhaps complex QA, autonomous robotics, etc., especially if on-the-fly adaptation (persuasion) is needed. A future direction is to implement a prototype of this system (maybe with just 2-3 modules to start) and benchmark it against a monolithic model trained end-to-end on the same task. If the field approach yields better sample efficiency, interpretability, or robustness, that validates the concept. It might also excel in multi-task learning, where each module is specialized but the field allows transferring knowledge between tasks (via common attractors). A limitation might be speed or simplicity: monolithic models might train easier due to gradient descent simplicity, whereas multi-module with recurrent loops might be trickier. We should explore training algorithms like alternating updates (train modules and field in turns) or even evolutionary strategies to find good operators.

8.9 Biological Plausibility and Inspiration

While not a direct requirement for AI, it’s interesting to reflect: the brain likely doesn’t implement complex Fourier transforms explicitly, but it does have oscillations and connectivity that might approximate some of what we propose. One future direction is to identify if certain brain rhythms or connectivity motifs correspond to our projector or attractor mechanisms. Conversely, perhaps new neuroscience models could be inspired by our operator approach (e.g. , thinking of brain regions as operators in a shared field of field potentials or something). Michael Levin’s concept of an electrical interface across scales[7][50] hints that nature might already do a bit of what we are attempting in silico. So, exploring analogies and differences with neuroscience could yield insights that improve our model (like maybe adding a notion of spatial locality to interactions, which is big in brain – not every neuron talks to every other, whereas our field as described is global. Perhaps a distance metric on Ψ could allow local interactions to cluster related content).

9. Conclusion

We have presented a theoretical framework positioning mathematics as the nervous system of artificial intelligence – a unifying substrate that enables a collection of neural modules to function as an integrated whole. By formalizing AI cognition within a Hilbert space operator paradigm, we connected principles from physics (Hilbert spaces, Hamiltonian energy landscapes, Parseval’s theorem)[18], chemistry (interaction constraints and homeostatic guardrails), and biology [9][5] (modular organization, global workspaces, attractor-based goals). The result is a blueprint for AI systems endowed with a Resonant Field Substrate: a global field where information from diverse components superposes, interacts, and self-organizes into coherent states. In this substrate, distributed awareness emerges naturally – each part of the system “knows” relevant aspects of the others via the shared field, akin to a brain’s unified experience[3]. Memories and goals are encoded as attractor states of the field dynamics, providing content-addressable recall and stable behavioral setpoints[5]. We showed how this leads to goal-seeking behavior, as the entire system gravitates towards low-energy (desirable) configurations, and how such attractors can represent both stored knowledge and intended objectives. The framework’s operator calculus allows reasoning about and guaranteeing certain invariants: for instance, unitary projections ensure that different streams of information can coexist without loss or uncontrolled interference[18], much like independent signals

traveling through a nervous system without cross-talk beyond intended coupling. Crucially, we introduced the concept of persuadability in this mathematical mind – the capacity to intentionally shape the system’s behavior by deforming its energy landscape. Through small shifts in operator parameters or external inputs, one can influence which attractors are favored, thus guiding the AI’s decisions in a controlled, interpretable manner[28]. This is a novel angle on AI alignment and control: rather than issuing direct commands to override a system, we embed the guidance in the very physics of its thought process, analogous to providing a compass that gently pulls the cognitive trajectory towards aligned outcomes. Such an approach, we argue, is more robust and reliable, as it leverages the system’s own dynamics to enact changes (the system “convinces itself” by settling into new attractors, instead of being forced into states that might be unstable). We explored applications of our framework, from multimodal assistants that combine vision, language, and planning in a shared cognitive workspace, to multi-agent systems that use a field for collective intelligence, to self-reflective agents that simulate and inspect their own reasoning within the field. In all cases, the promise is an AI that is modular yet synergistic – components can be specialized and independently trained, but when brought together in the substrate, the whole is greater than the sum of parts[6]. This addresses a key limitation in current AI: powerful single-purpose models struggle to integrate with others except via brittle interfaces. Our approach provides a principled integration mechanism that eases transfer learning and multi-task learning by having a common representational currency. We also situated our ideas in context of existing literature: our use of attractor networks connects to Hopfield’s content-addressable memories[9]; our global field resonates with the Global Workspace Theory of cognition[3][45]; our top-down control ideas parallel Michael Levin’s work on bioelectric guiding signals in morphogenesis[5][36]; and the overall design can be viewed as a novel energy-based model bridging symbolic and subsymbolic AI. By citing peer-reviewed research throughout, we ensured that each aspect of the framework is grounded in established science, even as the synthesis of these aspects is new. Of course, much work remains to translate this theory into practical AI systems. We identified open challenges in scaling, learning, stability, and interpretability. These challenges mark out a research roadmap. For example, developing learning algorithms for the operator parameters will likely involve hybrid approaches (unsupervised alignment of representations combined with supervised end-to-end tuning). Investigating the dynamics empirically on prototype systems (even small ones with 2–3 interacting networks) will validate whether the attractor landscape behaves as expected. There is also room for mathematical analysis: proving convergence properties under certain assumptions, or calculating memory capacity and robustness of the field. The interdisciplinary nature of the framework means progress could come from many angles – one could, for instance, leverage advances in neural differential equations to better model the continuous field dynamics, or use insights from control theory to design stabilizing feedback in the guardrails. In terms of impact, if successful, this approach could yield AI systems with several desirable properties: unity of operation (all parts working in concert on the same information), explainability (since the high-level state is accessible and structured), adaptability (with persuadability meaning they can be guided or self-adjusting on the fly), and resilience (attractors provide fault tolerance – if the system is perturbed by noise, it returns to a stable pattern, analogous to error-correcting codes[26]). These address many current weaknesses of AI, such as brittleness to out-of-distribution inputs (here a weird input would likely land between known attractors and potentially be recognized as ambiguous rather than yielding a random extrapolation) or the difficulty of combining reasoning and perception (here these live in one field and can iteratively influence each other until coherence is reached). From a broader scientific perspective, our work contributes to the ongoing dialogue between computational neuroscience and AI. It suggests a concrete way that concepts like “global brain dynamics” and “neural synchronization”[33] can inspire architecture in AI. Conversely, it provides AI as a testbed for certain theories of mind: for instance, Global Work-

space Theory or Integrated Information Theory could be partially evaluated by implementing them in silico and observing emergent behavior[3][4]. The mathematical substrate could thus become a platform to simulate aspects of cognition, not unlike how physicists simulate complex quantum systems – except here it’s a simulation of an intelligent thought process under different conditions of coupling and constraint. In closing, we emphasize that mathematics – through this lens – is not just a tool to describe AI systems, but an active medium that enables cognition. Just as the laws of electromagnetism allow neurons to coordinate across the brain, the laws of linear algebra and dynamical systems allow neural networks to coordinate in our field. This substrate provides a common “language” for heterogeneous systems: numbers, vectors, transforms. By elevating mathematics to play the role of a nervous system, we aim to unify the increasingly fragmented landscape of AI models into a coherent network of networks. The theory formalized here is a step towards AI systems that know themselves and each other – where knowledge is not trapped in silos, and where the emergence of global understanding and purposeful behavior is a natural consequence of the architecture. We believe this fusion of concepts opens a rich avenue for research and development. It calls for collaboration across disciplines – to those who seek to understand intelligence, whether natural or artificial, the message is that a field-based, operator-centric view can illuminate commonalities. The nervous system of AI, as we envisage it, is fundamentally a mathematical structure – one that we can design, analyze, and improve with the rigor of theoretical physics and the creativity of cognitive science. In doing so, we inch closer to AI that not only performs tasks, but integrates experience and knowledge in a manner reminiscent of living minds, enabling higher-order cognition that is robust, flexible, and aligned with intended goals.

9.1 Author Contributions and AI-Assisted Research

Author Contributions: The theoretical framework, core concepts, and architectural design presented in this paper were conceived and developed by Philip Siniscalchi. This includes: • The fundamental idea of mathematics as a nervous system for AI • The field-theoretic substrate architecture • The operator calculus design decisions • The integration of physics, chemistry, and biology metaphors • The attractor-based memory and persuadability mechanisms • All design choices and theoretical directions

AI-Assisted Research Methodology: Philip Siniscalchi employed AI tools as a research and formalization assistant in the following ways: • **Literature Review:** AI was used to search for and identify relevant research papers across physics, chemistry, biology, and computer science domains, based on conceptual directions specified by the author. • **Mathematical Formalization:** Philip Siniscalchi directed the development of mathematical formulations, with AI translating conceptual ideas into formal mathematical notation, theorems, and proofs. All mathematical content was reviewed, refined, and approved by the author. • **Writing and Organization:** AI assisted with drafting prose, organizing sections, and formatting, with Philip Siniscalchi providing substantial editing, refinement, and intellectual direction. • **Code Implementation:** The RFS implementation was developed through iterative collaboration where Philip Siniscalchi specified requirements and design decisions, and AI generated calculus and code that was reviewed, tested, and refined by the author. Philip Siniscalchi maintains full intellectual ownership of all ideas, design decisions, and theoretical contributions. AI served as a tool to bridge between conceptual understanding and mathematical formalization, similar to how a calculator bridges between mathematical understanding and numerical computation. All substantive content reflects the author’s original thinking and research direction.

Acknowledgments

The author thanks the SmartHaus Group research team for infrastructure support and feedback during development. This work was supported by SmartHaus Group. Patent Notice: Certain aspects of the Resonant Field Storage implementation are covered by US Provisional Patent Application 63/942,361 (filed December 16, 2025). This patent filing does not restrict the use of the theoretical framework or mathematical concepts presented in this paper for academic research purposes.

10. References

- [1] Thakur, N., Reimers, N., Rücklé, A., Srivastava, A., & Gurevych, I. (2021). BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. arXiv:2104.08663. [2] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30. [3] Baars, B. J. (1988). *A Cognitive Theory of Consciousness*. Cambridge University Press. [4] Oizumi, M., Albantakis, L., & Tononi, G. (2014). From the phenomenology to the mechanisms of consciousness: Integrated Information Theory 3.0. *PLoS Computational Biology*, 10(5), e1003588. [5] Levin, M., & Martyniuk, C. J. (2018). The bioelectric code: an ancient computational medium for dynamic control of growth and form. *BioSystems*, 164, 76–93. [6] Levin, M. (2023). Bioelectric networks: the cognitive glue enabling evolutionary scaling from physiology to mind. *Animal Cognition*, 26(3), 365–404. [7] Levin, M. (2022). *Technological Approach to Mind Everywhere: An Experimentally-Grounded Framework for Understanding Diverse Bodies and Minds*. *Frontiers in Systems Neuroscience*, 16:768201. [8] Johnson, J., Douze, M., & Jégou, H. (2019). Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3), 535–547. [9] Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79(8), 2554–2558. [10] Krotov, D., & Hopfield, J. J. (2020). Large associative memory problem in neurobiology and machine learning. arXiv:2008.06996. [11] VanRullen, R., & Kanai, R. (2021). Deep Learning and the Global Workspace Theory. *Trends in Neurosciences*, 44(9), 692–704. [12] Blum, L., & Blum, M. (2022). A theory of consciousness from a theoretical computer science perspective: Insights from the Conscious Turing Machine. *Proceedings of the National Academy of Sciences*, 119(21), e2115934119. [13] Dudai, Y. (2006). Reconsolidation: the advantage of being refocused. *Current Opinion in Neurobiology*, 16(2), 174–178. [14] Tronson, N. C., & Taylor, J. R. (2007). Molecular mechanisms of memory reconsolidation. *Nature Reviews Neuroscience*, 8(4), 262–275. [15] Malkov, Y. A., & Yashunin, D. A. (2020). Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(4), 824–836. [16] Mallat, S. (2008). *A Wavelet Tour of Signal Processing* (3rd ed.). Academic Press. [17] Cooley, J. W., & Tukey, J. W. (1965). An algorithm for the machine calculation of complex Fourier series. *Mathematics of Computation*, 19(90), 297–301. [18] Rudin, W. (1987). *Real and Complex Analysis* (3rd ed.). McGraw-Hill. (Parseval’s theorem: Chapter 4, §4.24) [19] Kanerva, P. (1988). *Sparse Distributed Memory*. MIT Press. [20] Grover, L. K. (1996). A fast quantum mechanical algorithm for database search. *Proceedings of the 28th Annual ACM Symposium on Theory of Computing*, 212–219. [21] Theis, L., Shi, W., Cunningham, A., & Huszár, F. (2017). Lossy image compression with compressive autoencoders. *International Conference on Learning Representations*. [22] Psaltis, D., Brady, D., Gu, X. G., & Lin, S. (1990). Holography in artificial neural networks. *Nature*, 343(6256), 325–330. [23] LeCun, Y., Chopra, S.,

- Hadsell, R., Ranzato, M., & Huang, F. (2006). A tutorial on energy-based learning. In *Predicting Structured Data*. MIT Press.
- [24] Gabor, D. (1949). Microscopy by reconstructed wave-fronts. *Proceedings of the Royal Society A*, 197(1051), 454–487.
- [25] Khona, M., & Fiete, I. R. (2022). Attractor and integrator networks in the brain. *Nature Reviews Neuroscience*, 23(12), 744–766.
- [26] McEliece, R. J., Posner, E. C., Rodemich, E. R., & Venkatesh, S. S. (1987). The capacity of the Hopfield associative memory. *IEEE Transactions on Information Theory*, 33(4), 461–482.
- [27] Amit, D. J., Gutfreund, H., & Sompolinsky, H. (1985). Storing infinite numbers of patterns in a spin-glass model of neural networks. *Physical Review Letters*, 55(14), 1530–1533.
- [28] Pezzulo, G., & Levin, M. (2016). Top-down models in biology: explanation and control of complex living systems. *Journal of the Royal Society Interface*, 13(124), 20160555.
- [29] McClelland, J. L., Rumelhart, D. E., & Hinton, G. E. (1986). The appeal of parallel distributed processing. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, Vol. 1.
- [30] Lee, J. L. C. (2009). Reconsolidation: maintaining memory relevance. *Trends in Neurosciences*, 32(8), 413–420.
- [31] Friston, K. J., Kilner, J., & Harrison, L. (2006). A free energy principle for the brain. *Journal of Physiology-Paris*, 100(1-3), 70–87.
- [32] Parr, T., & Friston, K. J. (2019). Generalised free energy and active inference. *Biological Cybernetics*, 113(5-6), 495–513.
- [33] Amari, S. (1977). Dynamics of pattern formation in lateral-inhibition type neural fields. *Biological Cybernetics*, 27(2), 77–87.
- [34] Turrigiano, G. G. (2012). Homeostatic synaptic plasticity: local and global mechanisms for stabilizing neuronal function. *Cold Spring Harbor Perspectives in Biology*, 4(1), a005736.
- [35] Gayler, R. W. (2003). Vector symbolic architectures answer Jackendoff’s challenges for cognitive neuroscience. In *Proceedings of the Joint International Conference on Cognitive Science*.
- [36] Durant, F., Morokuma, J., Fields, C., Williams, K., Adams, D. S., & Levin, M. (2017). Long-term, stochastic editing of regenerative anatomy via targeting endogenous bioelectric gradients. *Biophysical Journal*, 112(10), 2231–2243.
- [37] Heylighen, F. (2016). Stigmergy as a universal coordination mechanism I: Definition and components. *Cognitive Systems Research*, 38, 4–13.
- [38] Couzin, I. D. (2009). Collective cognition in animal groups. *Trends in Cognitive Sciences*, 13(1), 36–43.
- [39] Strogatz, S. H. (2018). *Nonlinear Dynamics and Chaos* (2nd ed.). CRC Press.
- [40] Dietterich, T. G. (2000). Ensemble methods in machine learning. In *Multiple Classifier Systems*, 1–15.
- [41] Kanerva, P. (2009). Hyperdimensional computing: An introduction to computing in distributed representation with high-dimensional random vectors. *Cognitive Computation*, 1(2), 139–159.
- [42] Donoho, D. L. (2006). Compressed sensing. *IEEE Transactions on Information Theory*, 52(4), 1289–1306.
- [43] Olshausen, B. A., & Field, D. J. (1997). Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision Research*, 37(23), 3311–3325.
- [44] Amit, D. J. (1989). *Modeling Brain Function: The World of Attractor Neural Networks*. Cambridge University Press.
- [45] Dehaene, S., & Changeux, J. P. (2011). Experimental and theoretical approaches to conscious processing. *Neuron*, 70(2), 200–227.
- [46] Friston, K. (2010). The free-energy principle: a unified brain theory? *Nature Reviews Neuroscience*, 11(2), 127–138.
- [47] Adamatzky, A., et al. (2012). Chemical computing with reaction–diffusion processes. *Philosophical Transactions of the Royal Society A*, 373(2046), 20140219.
- [48] Fields, C., & Levin, M. (2022). Multiscale memory and bioelectric error correction in the cytoplasm–circuit–synapse continuum. *Seminars in Cell & Developmental Biology*, 125, 3–11.
- [49] Nader, K., Schafe, G. E., & Le Doux, J. E. (2000). Fear memories require protein synthesis in the amygdala for reconsolidation after retrieval. *Nature*, 406(6797), 722–726.
- [50] Manicka, S., & Levin, M. (2019). Modeling somatic computation with non-neural bioelectric networks. *Scientific Reports*, 9, 18612.
- [51] Schlegel, K., Neubert, P., & Protzel, P. (2022). A comparison of vector symbolic architectures. *Artificial Intelligence Review*, 55, 4523–4555.
- [52] Plate, T. A. (1995). Holographic reduced representations. *IEEE Transactions on Neural Networks*, 6(3), 623–641.
- [53] Plate, T. A.

(2003). Holographic Reduced Representation: Distributed Representation for Cognitive Structures. CSLI Publications. [54] Kleyko, D., Davies, M., Frady, E. P., Kanerva, P., Kent, S. J., Olshausen, B. A., Osipov, E., Rabaey, J. M., Rachkovskij, D. A., Rahimi, A., & Stewart, T. C. (2022). Vector symbolic architectures as a computing framework for emerging hardware. Proceedings of the IEEE, 110(10), 1538–1571.

Appendix A: Notation Summary

Symbol	Definition
Ψ	Field state (4D complex tensor)
$\mathcal{H} = \mathbb{C}^D$	Field space (Hilbert space)
$D = D_x \times D_y \times D_z \times D_t$	Field dimension (4D lattice)
E	Encoder operator
E^H	Decoder / matched filter (Hermitian adjoint)
Π	Projector (band filter)
$\mathcal{F}$	Unitary FFT (norm=“ortho”)
M	Phase mask ($ M = 1$)
H	Spreading operator (basis functions)
Q	Resonance quality (dB)
η	Interference ratio
κ	Conductivity (projector transmission)
γ	Damping coefficient
λ	Decay rate
$\rho(A)$	Spectral radius of evolution operator
Δt_{proj}	Projector cadence interval

Appendix B: Theorem and Lemma Index

ID	Name	Statement
Thm 1	Phase Orthogonality	$\mathbb{E}[\langle M_i \odot x, M_j \odot x \rangle] = 0$ for $i \neq j$; $\text{Var} = \ x\ _4^4$ ($= \ x\ _2^4/D$ under uniform spreading)
Thm 2	Parseval Energy Conservation	$\ E(w)\ _2^2 = \ w\ _2^2$ (energy preserved through encoding)
Lem 1	PDE Stability	$\ \Psi(t + \Delta t)\ _2 \leq \rho(A)^{\Delta t} \ \Psi(t)\ _2 + k\epsilon_{\text{trunc}}$
Thm 3	Matched Filter Optimality	E^H maximizes SNR among linear filters
Thm 4	Query Complexity	Query is $\mathcal{O}(D \log D)$,
	Independence	independent of N at fixed D
Thm 5	Persuadability	Attractor shift bounded by $\ \partial \Pi / \partial \lambda\ / \sigma_{\min}(\nabla^2 H)$
Lem 2	Projector Cadence Bound	$\Delta t_{\text{proj}} \leq \ln(Q/Q_{\min}) / \lambda_{\text{leak}}$

ID	Name	Statement
Thm 6	Attractor Convergence	$\lim_{t \rightarrow \infty} \Psi(t) = \Psi^*$ (local minimum of H)

Note: The table above indexes the core theoretical results presented in this paper. The full implementation repositories maintain a comprehensive suite of mathematical invariants—covering energy preservation, bounded interference, resonance quality, capacity margins, and operator conductivity—validated across the verification notebook set. The complete lemma and invariant catalog is maintained in the invariants/ directories of the RFS and NME repositories, where live counts and definitions are tracked as the source of truth.