# SOMA: Towards Self-Organizing Memory Architectures for Autonomous Systems

**Anthony D. Martin - Cadenzai, Inc.*; Alexander Isaev; Hreetam Paul; Zarreen Reza**

## Abstract

Autonomous systems require memory architectures that actively organize and evolve knowledge structures. We present SOMA (Self-Organizing Memory Architecture), a formal framework outlining foundational design constraints for autonomous memory management. SOMA introduces Executable Memory Units (EMUs) with five validity constraints ensuring reliable activation, and recursive insight mechanisms enabling self-organization. We validate SOMA across metacognitive tasks (5.2% performance gain), conversational coherence (5/6 metrics improved), and continual learning (28% higher reward, catastrophic forgetting prevention). Results demonstrate that SOMA's formal constraints enable practical cognitive advantages while offering formal guarantees under the stated bounded-symbol assumptions for autonomous operation.

## 1 Introduction

Modern autonomous systems face a fundamental challenge: how to build memory architectures that don't just store and retrieve information, but actively organize, reason over, and evolve their knowledge structures. Consider a household robot that must learn manipulation strategies, adapt to new environments, and transfer knowledge across tasks—all while maintaining reliable operation.

Existing memory systems lack principled frameworks for autonomous memory management. Storage-focused approaches provide no guarantees about when memories should activate or how they compose. Learning-based systems suffer from catastrophic forgetting and lack interpretability.

We present SOMA (Self-Organizing Memory Architecture), a formal framework that bridges this gap by defining formal conditions for self-organizing memory in autonomous agents. SOMA introduces:

- Executable Memory Units (EMUs): Memory structures with formal activation guarantees through five validity constraints
- Recursive Insight Mechanisms: PRCOE-structured derivation of new memories from existing knowledge
- Bounded Self-Organization: Mathematical guarantees preventing unbounded memory growth while enabling adaptation

Our key insight is that autonomous memory requires both symbolic boundedness and utility-driven selection to achieve reliable self-organization without sacrificing adaptability.

Contributions: (1) Formal framework for autonomous memory with validity guarantees, (2) Recursive insight mechanisms enabling bounded self-organization, (3) Empirical validation showing 5-82improvements across cognitive tasks.

Crucially, SOMA is not a fully instantiated system but rather a specification of symbolic constraints, validity conditions, and recursive mechanisms that enable principled self-organization. This

*am@cadenzai.net

Preprint. Under review.

framework provides a set of foundational design constraints that enable autonomous memory, allowing diverse instantiations across different domains and architectures.

# 2 SOMA Framework

## 2.1 Memory Structures: Fragments and EMUs

SOMA defines two classes of memory structures:

**Memory Fragments** serve as flexible symbolic records that:

- Provide stable storage for recall and introspection
- Support retrieval through similarity matching, relevance scoring, or other flexible methods
- Do NOT require deterministic triggers
- May have procedural contents that are executed

**Executable Memory Units (EMUs)** extend fragments with strict activation guarantees.

An Executable Memory Unit (EMU) must satisfy five essential properties to function reliably in autonomous systems:

- Triggerability: When should this memory activate?
- Ownership: Who created this memory and in what context?
- Boundedness: Can we ensure finite computational resources?
- Utility: Does this memory actually help performance?
- Reachability: Will this memory ever have a chance to activate?

For example, in Figure 1, a chess strategy memory might trigger when "#kings_gambit" appears in context tags, know it was created by the chess_strategy_module, operate over a finite set of chess positions, provide positive expected value, and have non-zero probability of the trigger condition occurring.

Formally, this becomes:

**Definition 1 (EMU Validity).** An EMU $M$ is valid iff:

$$\text{Valid}(M) = T \land S_O \land B_T \land U \land R_T \tag{1}$$

where:

- $T$: Symbolic, deterministic trigger condition
- $S_O$: Scoped ownership (author/owner metadata)
- $B_T$: Trigger resides in finite symbolic domain
- $U$: Expected utility exceeds threshold e.g. $\delta \in [-1, 1]$
- $R_T$: Non-zero probability of trigger emission

**Key Design Principle:** Memory structures that fail any EMU validity criterion are automatically relegated to Memory Fragments, ensuring the system maintains both strict EMUs for reliable activation and flexible fragments for broader cognitive support.

## 2.2 Insight Mechanisms

SOMA enables memory derivation through insights, structured as PRCOE records:

**Definition 2 (Insight Structure).** Insights operate over memory structures and produce new memory structures (incl. Meta-EMUs, memory fragments, EMUs). An insight is a Pattern-Relation-Confidence-Observation-EMU record:

$$I = (P_c, P_{\text{alias}[]}, R, C, O, E) \tag{2}$$

where:

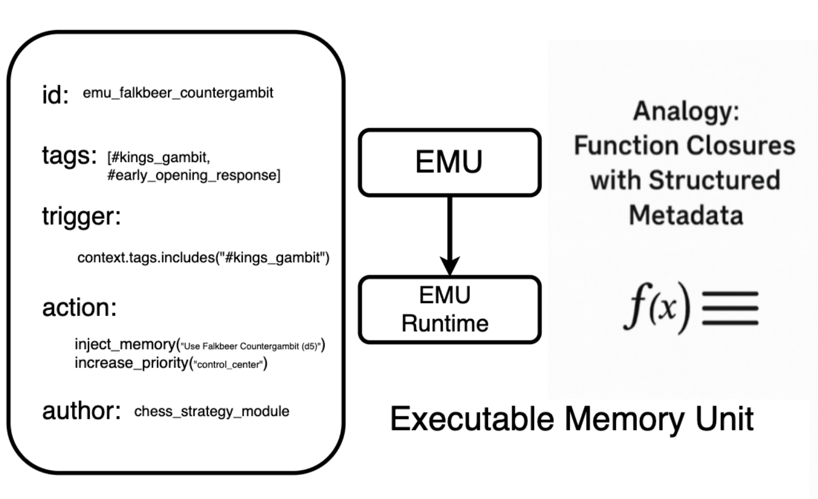


Figure 1: SOMA EMU

- $P_c$: Deterministic canonical label derived from $R$
- $P_{\text{alias}[]}$: Optional human-friendly aliases
- $R$: Finite relation graph
- $C$: Confidence tuple $\{C_P, C_E, C_U\}$
- $O$: Observations (memory structures analyzed)
- $E$: Set of proposed EMUs or memory structures

**Insight Validity:** $\text{Valid}(I) = D \wedge V \wedge C \wedge S$, where PRCOE fields map to:

- $P_c, R \to D$ (Definability), $S$ (Symbolic Boundedness)
- $O \to V$ (Verifiability via observations)
- $C \to C$ (Confidence threshold)

The three insight mechanisms are:

1. **Retrospective Graph Insights**: $R$ captures patterns in $O$ from memory history
2. **Foresight Simulated Insights**: $O$ contains simulated contexts
3. **Symbolic Heuristic Insights**: $R$ may be $\emptyset$, template-based rules

### 2.3 Recursive Meta-Memory

**Theorem 1 (Recursive Validity Closure).** If $\text{Valid}(I)$ and $U_e > \delta$ for $e \in E$, then derived meta-EMUs $M_e$ maintain full validity guarantees, enabling unbounded reflective recursion under bounded constraints.

### 2.4 Precision Metric

For query $Q$, positive-utility precision is:

$$\text{Precision}^+(Q) = \frac{\sum T(e, Q) \cdot U^+(e, Q)}{\sum T(e, Q)} \tag{3}$$

where $T(e, Q) \in \{0, 1\}$ is the trigger match and $U^+(e, Q) = \max(U(e, Q), 0)$.

## 3 Experimental Validation

We validate key aspects of the SOMA formalism across three complementary domains that test different facets of self-organizing memory: metacognitive adaptation through Meta-Memory Fragments, conversational coherence via memory-based interventions, and continual learning with hybrid memory architectures.

### 3.1 Hidden-State Task Suite: Metacognitive Performance

**Setup:** We evaluate SOMA's metacognitive capabilities on a sequential decision task with seven hidden state dimensions. Agents select ordered subsequences from six cognitive operations based solely on utility feedback. We compare SOMA Full Metacognitive (MC), SOMA No-Metacognition (NM), Greedy-Fitness, and Random baselines across 30 paired runs (240,000 training episodes).

**Implementation Note:** This experiment uses Memory Fragments—SOMA-compliant memory structures with similarity-based activation rather than deterministic triggers. This demonstrates SOMA's flexibility: systems can leverage the formalism's insight mechanisms and validity constraints while adapting activation methods to their domain.

**Results:**

| System | Final Utility | MI (bits) | Entropy | Convergence Episode |
|---|---|---|---|---|
| SOMA-MC | **0.562±0.005** | **0.671** | **2.877** | **157** |
| SOMA-NM | 0.534±0.005 | 0.368 | 5.722 | 471 |
| Greedy | 0.483±0.005 | 0.459 | 4.303 | 792 |
| Random | 0.421±0.004 | 0.059 | 6.595 | N/A |

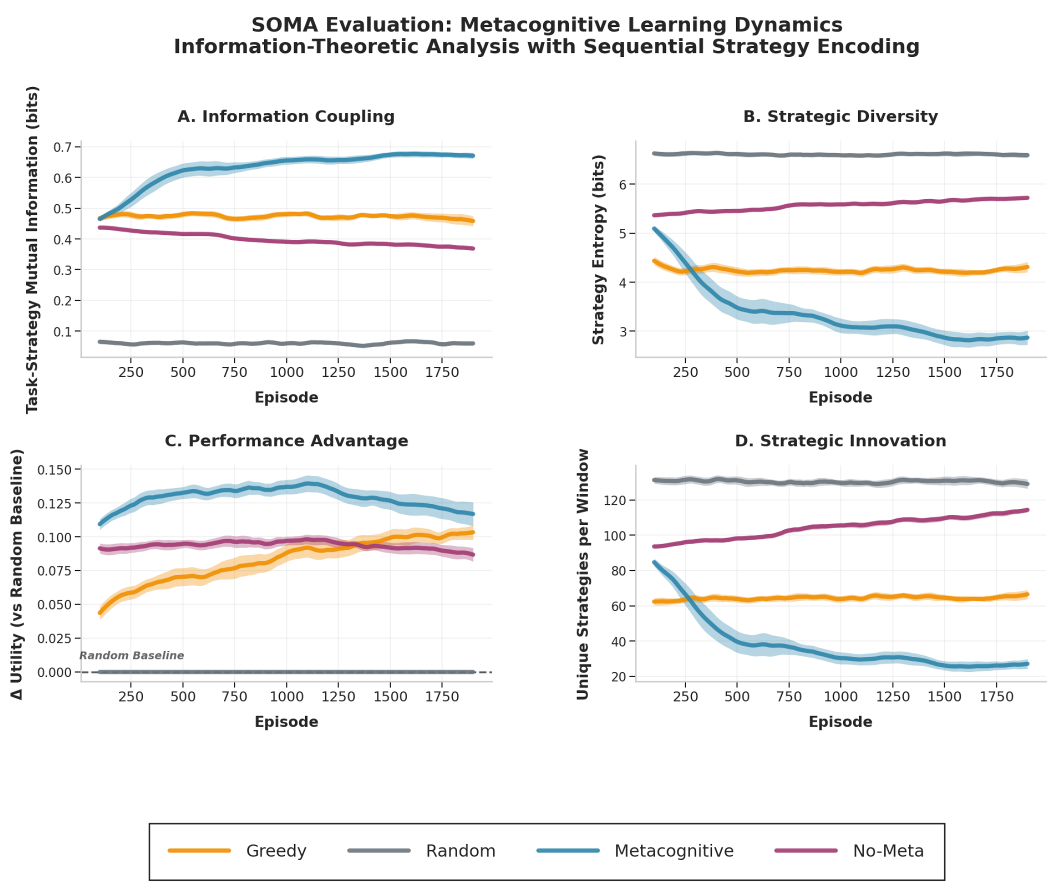


Figure 2: Metacognitive learning dynamics with sequential-strategy encoding

**Key Findings:**

- **5.2% improvement** over non-metacognitive variant (Cohen's d = 0.15, $p = 2.5 \times 10^{-5}$)
- **82.3% higher mutual information** between tasks and strategies (0.671 vs 0.368 bits)
- **3× faster convergence** to effective performance (157 vs 471 episodes)
- **13.6±0.5 Meta-Memory Fragments** formed per run with 77.4% recommendation success

The metacognitive variant achieves superior performance through strategic curation rather than broad exploration, discovering fewer but more targeted strategies (703 strategies, 2.9 bits entropy) compared to the non-metacognitive variant (714 strategies, 5.7 bits entropy).

### 3.2 Coherence Evaluation: Language Model Enhancement

**Setup:** We evaluate SOMA's impact on conversational coherence using a Coherence-Onboarding Protocol (COP), a 10-turn interaction task. Two Gemma models (4B and 27B parameters) are tested with and without SOMA integration across 10 runs each.

**Results:**

Table 1: Coherence Improvements (p-values for SOMA vs Non-SOMA)

| Metric | Gemma 4B | | Gemma 27B | |
|---|---|---|---|---|
| | SOMA $\Delta$ | p-value | SOMA $\Delta$ | p-value |
| Referential Integrity | +0.28 | 0.0024** | +0.24 | <0.0001** |
| Lexical Consistency | +0.08 | 0.1027 | +0.10 | 0.0406* |
| Contextual Relevance | -0.20 | 0.0161* | +0.23 | <0.0001** |
| Structural Continuity | +0.48 | <0.0001** | +0.35 | <0.0001** |
| Tone Consistency | 0.00 | N/A | -0.03 | 0.0007** |
| Directive Adherence | +0.24 | 0.0031** | +0.06 | <0.0001** |

(*p<0.05, **p<0.01)

**Key Findings:**

- Gemma 27B+SOMA shows **improvements in 5/6 metrics**
- Most substantial gains in **Structural Continuity** (+0.48 for 4B, +0.35 for 27B)
- SOMA's interventional memory injection maintains conversational coherence across extended interactions

### 3.3 Color-Drift Environment: Continual Learning

**Setup:** We evaluate memory architectures in a non-stationary 20×20 grid navigation task where reward associations flip every 4,000 steps. Three controllers are compared: SOMA-only (hand-written EMUs with A* planning), Q-learning with MLP, and hybrid Q+SOMA.

**Results:**

| Architecture | Mean Reward | 95% CI | Final Performance | Degradation |
|---|---|---|---|---|
| Q+SOMA-MLP | **0.152±0.001** | [0.150, 0.154] | 0.108±0.015 | Less |
| SOMA-only | 0.126±0.000 | [0.125, 0.127] | **0.133±0.009** | None |
| Q-MLP | 0.119±0.001 | [0.117, 0.121] | 0.066±0.004 | Severe |

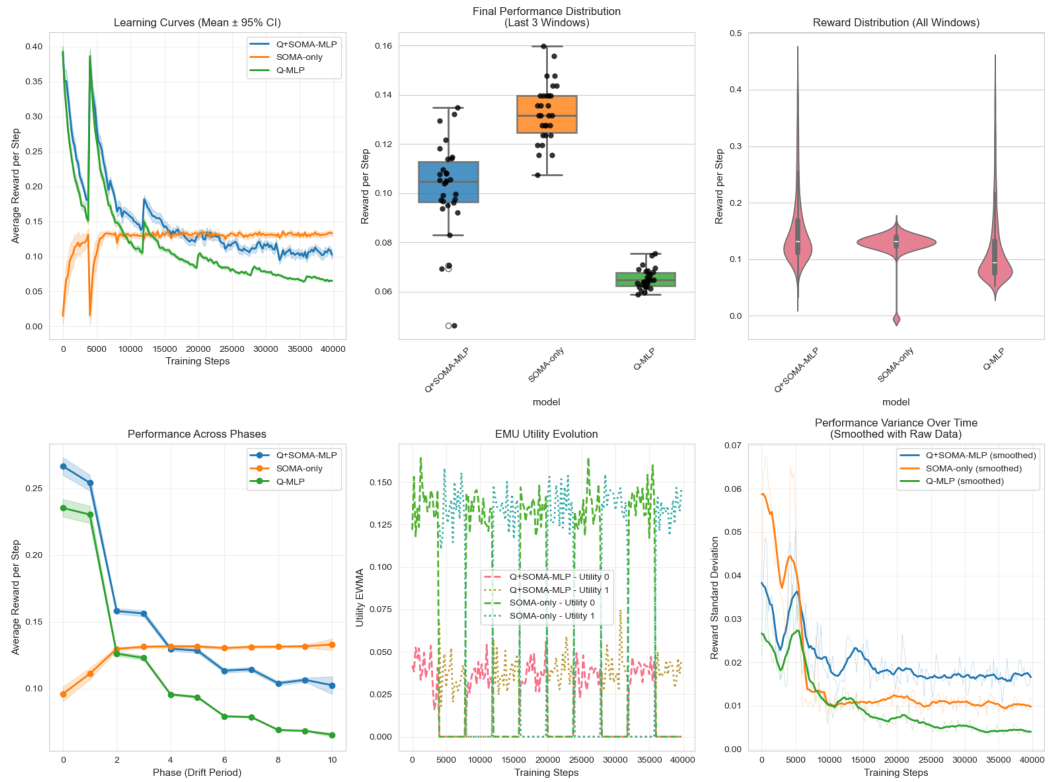


Figure 3: Color-Drift Evaluation Over Time with Distributions

**Key Findings:**

- **28% higher reward** for Q+SOMA hybrid vs pure Q-learning
- SOMA-only shows **instant recovery** from regime shifts (<1 window)
- Q-learning exhibits **catastrophic forgetting** (0.389→0.066 degradation)
- Hybrid achieves best average performance while preserving adaptation

# 4 Related Works

SOMA extends multiple lineages of research: symbolic cognitive architectures, self-organizing systems theory, and modern neurosymbolic memory systems.

**Self-Organization and Cognitive Foundations:** The formal study of self-organization spans from Ashby's cybernetic framework [Ashby, 1947] and Haken's synergetics [Haken, 1983] to Kauffman's work on evolutionary order [Kauffman, 1992]. Recent efforts provide quantitative criteria through statistical complexity [Shalizi et al., 2004] and universal adaptation properties [Gershenson, 2025]. Symbolic AI established computational infrastructure through production systems [Newell, 1973, Newell and Simon, 1972], frame-based systems like KRL [Bobrow and Winograd, 1977], schemas and scripts [Schank and Abelson, 1977], and blackboard architectures [Erman et al., 1980]. Contemporary cognitive architectures formalized these concepts in Soar [Laird et al., 1987], ACT-R [Anderson et al., 1997, 2004], CLARION [Sun, 2006], Cyc's microtheories [Lenat, 1995], and Global Workspace Theory implementations [Baars, 1988, Franklin et al., 2007] enabled by event-driven architectures [Carzaniga et al., 2001, Eugster et al., 2003] .

**Contemporary Neurosymbolic Memory:** Recent work has revisited structured memory through Memory Networks [Weston et al., 2015], MemGPT [Packer et al., 2024], ChatDB [Hu et al., 2023], Symbolic Working Memory [Wang et al., 2024], and CoALA [Sumers et al., 2024]. Systems like CLIN demonstrate continual learning [Majumder et al., 2023], while Reflexion shows verbal reinforcement learning [Shinn et al., 2023].

SOMA distinguishes itself through formal specifications of executable memory units, symbolic boundedness constraints, and recursive meta-cognitive mechanisms—providing formal conditions for self-organizing memory rather than prescribing specific implementations.

## 4.1 Limitations and Future Research Directions

### 4.1.1 Current Implementation Limitations

The experimental instantiations evaluated here each carry methodological constraints that limit immediate generalization. The step-sequenced metacognitive architecture constrains behavioral expression to predetermined cognitive primitives rather than discovering domain-appropriate action spaces. The coherence evaluation relies entirely on model-generated judgments without human supervision, potentially missing nuanced conversational failures that automated metrics cannot capture. The color-drift environment, while demonstrating clear memory architecture advantages, uses a relatively simple 2D navigation task with binary reward structure—more complex environments with high-dimensional state spaces, partial observability, or multi-objective rewards may reveal different architectural trade-offs. Additionally, all three experiments evaluate SOMA in relatively short-horizon contexts; the framework's performance in truly long-term deployment scenarios with thousands of accumulated memories remains unexplored.

### 4.1.2 SOMA Formalism Extensions

The theoretical framework opens several fundamental research directions that extend beyond step-sequenced or evaluation-specific architectures:

**Alternative Cognitive Architectures:** The SOMA formalism's EMU validity constraints and recursive insight mechanisms could instantiate radically different cognitive organizations. Parallel processing systems might form EMUs over concurrent operations rather than sequential patterns. Hierarchical architectures could implement multi-level abstraction through recursive meta-EMU formation. Distributed cognition systems could share SOMA-compliant memory structures across multiple agents while maintaining individual confidence tracking and symbolic boundedness.

**Domain-Adaptive Symbol Emergence:** While our evaluations operate on pre-defined symbolic primitives, the SOMA formalism supports autonomous symbol discovery through observational compression. Future instantiations could implement memory structure formation over raw perceptual streams, enabling systems to discover domain-appropriate symbolic vocabularies rather than inheriting predetermined operation sets. This capability is particularly relevant for embodied AI systems operating in novel environments where optimal behavioral abstractions are unknown a priori.

**Real-Time Confidence-Modulated Learning:** The current episodic framework demonstrates confidence-gated memory formation but does not address continuous learning under temporal constraints. Future research should explore how SOMA's triggerability and utility constraints enable safe exploration in real-time systems where catastrophic failures must be avoided through metacognitive uncertainty awareness.

### 4.1.3 Application Domain Expansions

The step-sequenced cognitive architecture and Meta-Memory Fragment mechanisms suggest immediate applications to domains requiring adaptive behavioral sequencing:

**Embodied Autonomous Systems:** Humanoid robots could implement SOMA-compliant EMUs over manipulation sequences (perceive$\rightarrow$approach$\rightarrow$grasp$\rightarrow$lift$\rightarrow$evaluate), with Meta-Memory Fragment formation enabling context-dependent strategy adaptation. The confidence-modulated memory formation addresses the fundamental challenge of learning without dangerous exploration around humans or valuable equipment. Symbol emergence through behavioral compression could enable robots to discover concepts like "fragile-object-context" or "high-risk-zone" autonomously.

**Large Language Model Agent Reasoning:** Current LLM agents rely on fixed reasoning templates ("think step by step," "analyze then decide"). SOMA-enabled agents could develop adaptive cognitive sequences that vary by problem context—different reasoning patterns for debugging versus research versus financial analysis. The recursive meta-memory and insight capabilities would enable agents to learn about their own reasoning effectiveness across different domains leading to self-improvement.

The user-facing benefits from SOMA-inspired primitives also include deeper user-agent connection through interventional EMUs derived from insights.

**Automated Scientific Discovery:** Research methodologies themselves could be formalized as SOMA-compliant behavioral sequences or informed by SOMA primitives, enabling systems to learn which experimental approaches work for different phenomena. Cross-domain insight transfer through Meta-Memory Fragment formation could recognize when methodological patterns from one field apply to another. Meta-research capabilities could evolve novel scientific methodologies through recursive insight generation.

## 5 Conclusion

This work introduces SOMA as a framework for self-organizing memory in autonomous systems, providing formal specifications for executable memory units and recursive insight mechanisms. Our empirical validations demonstrate that SOMA's constraints enable measurable cognitive advantages: improved metacognitive adaptation (5.2% utility gain, 82.3% higher task-strategy mutual information), enhanced conversational coherence (significant improvements in 5/6 metrics), and robust continual learning (28% higher reward with catastrophic forgetting prevention). Critically, SOMA is not a prescriptive architecture but a formal framework outlining foundational design constraints for autonomous memory management. This flexibility allows diverse instantiations—from similarity-based Memory Fragments in metacognitive tasks to deterministic EMUs in navigation—while maintaining formal validity guarantees. As autonomous systems face increasingly complex, open-ended environments, SOMA's symbolic boundedness and recursive meta-memory capabilities provide a foundation for building agents that not only remember and retrieve, but actively organize, reason over, and evolve their knowledge structures. Future work should explore SOMA's application to embodied AI, scientific discovery systems, and emergent symbol formation, where the framework's formal constraints can guide the development of truly autonomous cognitive architectures.

## 6 Acknowledgements

An immense amount of gratitude to Bryan O'Gorman, Marcus Badgeley, Eric Aboussouan and Ahmed Abdeen Hamed for their input and generosity of time. Another special thank you to all Cadenzai researchers who participate in discussions and intellectual exchange every weekend. Without everyone's presence and support this research would not be possible.

This document and experiments contained represents a collaborative effort between the authors and generative AI systems. The conceptual framing, research direction, and structural scaffolding were provided by the authors. Certain prose, mathematical formulations, and insights were developed through an iterative process involving large language models guided by targeted prompting, critical review, and selective refinement over several months.

The result is a jointly authored artifact: the ideas and expressions within cannot be cleanly attributed to either human or machine alone. Rather, they emerged through recursive interaction and alignment between the two.

## References

W. R. Ashby. Principles of the self-organizing dynamic system. *The Journal of General Psychology*, 37(2):125–128, 1947. doi:10.1080/00221309.1947.9918144.

H. Haken. *Synergetics: An Introduction*. Springer Series in Synergetics. Springer Berlin Heidelberg, 1983. ISBN 9783540123569. URL https://books.google.co.uk/books?id=-WSwQgAACAAJ.

Stuart Kauffman. The origins of order: Self-organization and selection in evolution. *emergence.org*, 15, 08 1992. doi:10.1142/9789814415743_0003.

Cosma Shalizi, Kristina Shalizi, and Robert Haslinger. Publisher's note: Quantifying self-organization with optimal predictors [phys. rev. lett. 93, 118701 (2004)]. *Physical Review Letters*, 93, 09 2004. doi:10.1103/PhysRevLett.93.149902.

Carlos Gershenson. Self-organizing systems: what, how, and why? *npj Complexity*, 2(1):10, 2025. doi:10.1038/s44260-025-00031-5.

Allen Newell. Production systems: Models of control structures. In W. C. Chase, editor, *Visual information processing*, pages 463–526. Academic Press, New York, 1973.

Allen Newell and Herbert A. Simon. *Human Problem Solving*. Prentice-Hall, Englewood Cliffs, NJ, 1972.

Daniel G. Bobrow and Terry Winograd. An overview of krl, a knowledge representation language. *Cognitive Science*, 1(1):3–46, 1977. ISSN 0364-0213. doi:https://doi.org/10.1016/S0364-0213(77)80003-7. URL `https://www.sciencedirect.com/science/article/pii/S0364021377800037`.

Roger C. Schank and Robert P. Abelson. *Scripts, Plans, Goals and Understanding: An Inquiry into Human Knowledge Structures*. L. Erlbaum, Hillsdale, NJ, 1977.

Lee D. Erman, Frederick Hayes-Roth, Victor R. Lesser, and D. Raj Reddy. The hearsay-ii speech-understanding system: Integrating knowledge to resolve uncertainty. *ACM Comput. Surv.*, 12(2): 213–253, June 1980. ISSN 0360-0300. doi:10.1145/356810.356816. URL `https://doi.org/10.1145/356810.356816`.

John E. Laird, Allen Newell, and Paul S. Rosenbloom. Soar: An architecture for general intelligence. *Artificial Intelligence*, 33(1):1–64, 1987. ISSN 0004-3702. doi:https://doi.org/10.1016/0004-3702(87)90050-6. URL `https://www.sciencedirect.com/science/article/pii/0004370287900506`.

John R. Anderson, Michael Matessa, and Christian Lebiere. Act-r: A theory of higher level cognition and its relation to visual attention. *Human-Computer Interaction*, 12(4):439–462, 1997. doi:10.1207/s15327051hci1204_5.

John R. Anderson, Daniel Bothell, Michael D. Byrne, Scott Douglass, Christian Lebiere, and Yulin Qin. An integrated theory of the mind. *Psychological Review*, 111(4):1036–1060, 2004. doi:10.1037/0033-295X.111.4.1036.

Ron Sun. The clarion cognitive architecture: Extending cognitive modeling to social simulation. In *Cognition and multi-agent interaction*, pages 79–99. Cambridge University Press, 2006.

Douglas B. Lenat. Cyc: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38(11):33–38, November 1995.

Bernard J. Baars. *A Cognitive Theory of Consciousness*. Cambridge University Press, New York, 1988.

Stan Franklin, Uma Ramamurthy, Sidney D'Mello, Lee Mccauley, Aregahegn Negatu, Rodrigo Silva, and Vivek Datla. Lida: A computational model of global workspace theory and developmental learning. *AAAI Fall Symposium on AI and Consciousness: Theoretical Foundations and Current Approaches*, 01 2007.

Antonio Carzaniga, David S. Rosenblum, and Alexander L. Wolf. Design and evaluation of a wide-area event notification service. *ACM Transactions on Computer Systems*, 19(3):332–383, 2001. doi:10.1145/380749.380767.

Patrick Th. Eugster, Pascal A. Felber, Rachid Guerraoui, and Anne-Marie Kermarrec. The many faces of publish/subscribe. *ACM Computing Surveys*, 35(2):114–131, 2003. doi:10.1145/857076.857078.

Barbara Hayes-Roth and Frederick Hayes-Roth. A cognitive model of planning. *Cognitive Science*, 3(4):275–310, 1979. doi:https://doi.org/10.1207/s15516709cog0304_1. URL `https://onlinelibrary.wiley.com/doi/abs/10.1207/s15516709cog0304_1`.

Mark Stefik. Planning and meta-planning (molgen: Part 2). *Artificial Intelligence*, 16(2):141–169, 1981. ISSN 0004-3702. doi:https://doi.org/10.1016/0004-3702(81)90008-4. URL `https://www.sciencedirect.com/science/article/pii/0004370281900084`.

Charles L. Forgy. Ops5 user's manual. Technical Report CMU-CS-81-135, Carnegie Mellon University, 1981.

Barbara Hayes-Roth. A blackboard architecture for control. *Artificial Intelligence*, 26(3):251–321, 1985. ISSN 0004-3702. doi:https://doi.org/10.1016/0004-3702(85)90063-3. URL `https://www.sciencedirect.com/science/article/pii/0004370285900633`.

Allen Newell. Heuristic programming: Ill-structured problems. 1967.

Roger C. Schank. Conceptual dependency: A theory of natural language understanding. *Cognitive Psychology*, 3(4):552–631, 1972.

Christopher J. C. H. Watkins and Peter Dayan. Q-learning. 8(3-4):279–292. ISSN 0885-6125. doi:10.1007/bf00992698. URL `https://doi.org/10.1007/bf00992698`.

Pat Langley and Dongkyu Choi. A unified cognitive architecture for physical agents. In *Proceedings of the National Conference on Artificial Intelligence*, volume 21, page 1469, 2006.

John E. Laird. *The Soar Cognitive Architecture*. The MIT Press, 04 2012. ISBN 9780262301145. doi:10.7551/mitpress/7688.001.0001. URL `https://doi.org/10.7551/mitpress/7688.001.0001`.

Chris Eliasmith, Terrence C. Stewart, Xuan Choo, Trevor Bekolay, Travis DeWolf, Yichuan Tang, and Daniel Rasmussen. A large-scale model of the functioning brain. *Science*, 338(6111):1202–1205, 2012.

Paul S. Rosenbloom. The sigma cognitive architecture and system. *AISB Quarterly*, 136:4–13, 2013.

Jason Weston, Sumit Chopra, and Antoine Bordes. Memory networks, 2015. URL `https://arxiv.org/abs/1410.3916`.

Alex Graves, Greg Wayne, and Ivo Danihelka. Neural turing machines, 2014.

Trevor Bekolay, James Bergstra, Eric Hunsberger, Travis DeWolf, Terrence C. Stewart, Daniel Rasmussen, and Chris Eliasmith. Nengo: A python tool for building large-scale functional brain models. *Frontiers in Neuroinformatics*, 7:48, 2014.

Fabian Peller-Konrad, Rainer Kartmann, Christian R.G. Dreher, Andre Meixner, Fabian Reister, Markus Grotz, and Tamim Asfour. A memory system of a robot cognitive architecture and its implementation in armarx. *Robotics and Autonomous Systems*, 164:104415, June 2023. ISSN 0921-8890. doi:10.1016/j.robot.2023.104415. URL `http://dx.doi.org/10.1016/j.robot.2023.104415`.

Chenxu Hu, Jie Fu, Chenzhuang Du, Simian Luo, Junbo Zhao, and Hang Zhao. Chatdb: Augmenting llms with databases as their symbolic memory, 2023. URL `https://arxiv.org/abs/2306.03901`.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. Memgpt: Towards llms as operating systems, 2024. URL `https://arxiv.org/abs/2310.08560`.

Theodore R. Sumers, Shunyu Yao, Karthik Narasimhan, and Thomas L. Griffiths. Cognitive architectures for language agents, 2024. URL `https://arxiv.org/abs/2309.02427`.

Siyuan Wang, Zhongyu Wei, Yejin Choi, and Xiang Ren. Symbolic working memory enhances language models for complex rule application. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17583–17604, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi:10.18653/v1/2024.emnlp-main.974. URL `https://aclanthology.org/2024.emnlp-main.974/`.

Yiming Du, Wenyu Huang, Danna Zheng, Zhaowei Wang, Sebastien Montella, Mirella Lapata, Kam-Fai Wong, and Jeff Z. Pan. Rethinking memory in ai: Taxonomy, operations, topics, and future directions, 2025. URL `https://arxiv.org/abs/2505.00675`.

Yaxiong Wu, Sheng Liang, Chen Zhang, Yichao Wang, Yongyue Zhang, Huifeng Guo, Ruiming Tang, and Yong Liu. From human memory to ai memory: A survey on memory mechanisms in the era of llms, 2025. URL https://arxiv.org/abs/2504.15965.

Bernal Jiménez Gutiérrez, Yiheng Shu, Weijian Qi, Sizhe Zhou, and Yu Su. From rag to memory: Non-parametric continual learning for large language models, 2025. URL https://arxiv.org/abs/2502.14802.

Rana Salama, Jason Cai, Michelle Yuan, Anna Currey, Monica Sunkara, Yi Zhang, and Yassine Benajiba. Meminsight: Autonomous memory augmentation for llm agents, 2025. URL https://arxiv.org/abs/2503.21760.

Bodhisattwa Prasad Majumder, Bhavana Dalvi Mishra, Peter Jansen, Oyvind Tafjord, Niket Tandon, Li Zhang, Chris Callison-Burch, and Peter Clark. Clin: A continually learning language agent for rapid task adaptation and generalization, 2023.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning, 2023.

Mikhail Prokopenko. Guided self-organization: Inception. 2014. URL https://api.semanticscholar.org/CorpusID:60316875.

## A Evaluation of Metacognitive Emergence in SOMA Agents

### A.1 Abstract

We present an evaluation of Self-Organizing Memory Architecture (SOMA) agents on a hidden-state sequential-cognition benchmark. Thirty fully paired runs were executed for four systems—SOMA Full Metacognitive (MC), SOMA No-Metacognition (NM), Greedy-Fitness baseline, and Random baseline—yielding 240,000 training episodes plus 6,000 fixed-difficulty probe episodes.

Key outcomes:

- Probe utility: $0.562 \pm 0.005$ (MC) $>$ $0.534 \pm 0.005$ (NM) $>$ $0.483 \pm 0.005$ (Greedy) $>$ $0.421 \pm 0.004$ (Random). MC vs NM: Cohen's d = 0.15, t = 4.22, p = $2.5 \times 10^{-5}$.
- Baseline-corrected advantage: 0.117 (MC) $>$ 0.103 (Greedy) $>$ 0.086 (NM) (vs Random).
- Task–strategy mutual information (bias-corrected): 0.671 bits (MC) $>$ 0.459 bits (Greedy) $>$ 0.368 bits (NM) $\gg$ 0.059 bits (Random); permutation $p < 1 \times 10^{-6}$ for all but Random. The SOMA metacognitive variant achieves stronger task-strategy coupling while exhibiting lower strategic entropy than the non-metacognitive variant.
- Meta-Memory Fragments: $13.6 \pm 0.5$ formed in every run, recommendation success = 77.4%.

Results confirm statistically significant metacognitive gains that survive controls for probe leakage, drift, curriculum bias and multiple-testing error.

### A.2 Metacognitive Architecture and Experimental Context

This evaluation instantiates SOMA's theoretical framework through a step-sequenced metacognitive architecture that operationalizes metacognition as learned confidence and memory management rather than abstract self-reflection. In this experimental context, metacognition manifests through three concrete mechanisms that extend base learning capabilities with self-monitoring and adaptive reuse functions.

The metacognitive system implements Meta-Memory Fragments—SOMA-compliant memory structures formed through self-observation when task-specific patterns exceed empirically-derived utility and confidence thresholds. Unlike true EMUs which require deterministic symbolic triggers, these Meta-Memory Fragments use similarity-based activation while utilizing SOMA inspired insight symbolic reasoning—specifically hindsight analysis to detect patterns in historical memory structures. These structures store successful strategy patterns for context-dependent retrieval, with formation criteria requiring pattern observations exceeding baseline thresholds and mean utility above 0.605.

A confidence-scored recommendation system generates strategic suggestions based on accumulated experience, enabling the agent to leverage past successes while maintaining decision autonomy. The system tracks recommendation usage rates and confidence scores, achieving 77.4% success when recommendations are followed across 42.2% usage instances. This mechanism exemplifies SOMA's principle of confidence-modulated learning, where agents develop quantified certainty about strategic choices rather than relying on uniform exploration.

The integration of these mechanisms enables meta-learning—learning not merely what strategies work, but when existing knowledge is sufficiently reliable to trust versus when further exploration is warranted. This operationalization distinguishes our approach from general notions of "thinking about thinking" by providing measurable, bounded cognitive enhancement through self-monitoring of performance patterns, strategy reflection on task-strategy combination effectiveness, and adaptive reuse of successful patterns based on confidence assessment.

Critically, metacognition in this instantiation operates as an enhancement layer rather than a replacement for core learning mechanisms. Both SOMA Full Metacognitive and SOMA No-Metacognition variants employ identical base learning architectures, with metacognitive capabilities providing efficiency gains through strategic curation rather than fundamentally different learning processes. This design isolation enables precise measurement of metacognitive contributions while demonstrating how SOMA's formal insight mechanisms translate into practical cognitive advantages.

### A.3 Experimental Framework and Information-Theoretic Innovation

The SOMA Metacognitive Hidden-State Task Evaluation evaluates agents' capacity for context-sensitive strategy selection through a novel application of mutual information theory to sequential decision-making. Each episode presents a task characterized by seven hidden state dimensions (each 0.0-1.0): information density, abstraction level, integration complexity, reflection benefit, time pressure, sequential dependency, and working memory load. Agents must select ordered subsequences from six cognitive operations: perceive, encode, abstract, integrate, reflect, and evaluate, based solely on utility feedback.

The evaluation methodology involves applying Miller-Madow bias-corrected mutual information to quantify adaptive behavior. Unlike previous applications of mutual information in cognitive architectures, our approach operates at the strategy-selection level rather than input-output mappings. Task complexity is discretized into three empirically-derived bins, while emergent strategies are classified into interpretable families through order-dependent token analysis. The resulting mutual information $I(T;S)$ between task characteristics and strategy families serves as a direct measure of adaptive intelligence, capturing the degree to which agents couple their cognitive approach based on environmental demands.

### A.4 Agent Variants

Table 2: System configurations and learning mechanisms

| System | Learning signal | Meta-Memory | Sequence policy |
|---|---|---|---|
| SOMA Full Metacognitive | Prediction-error accumulation ($\eta = 0.05$) with recommendation confidence tracking | Meta-Memory Fragments form when task-specific patterns exceed utility & confidence thresholds; max 50 with performance-based pruning | Mix of learned generative policy and recommendation following (tracked usage rate & confidence scores) |
| SOMA No-Metacognition | Same prediction-error accumulation as Full Metacognitive | None | Same generative policy but **disabled** Meta-Memory Fragment formation and recommendation following |
| Greedy-Fitness | Adaptive strategy learning with task-specific memorization and reuse patterns | None | Learned sequential strategies with strategic innovation and complexity evolution over time |
| Random | Baseline with minimal or no learning signal | None | 2–6 random operations, always ending with `evaluate` |

All systems are forbidden to update during probe episodes; RNG state is reset each episode to guarantee perfect pairing across systems.

### A.5 Aggregate Performance

Welch tests with Holm–Šidák correction retain significance ($\alpha = 0.05$) for all MC comparisons on probe utility and $\Delta U$; learning-rate and raw improvement differences are significant only versus Random and Greedy.

Table 3: SOMA Evaluation Performance Summary

| System | MI (bits) | Entropy (bits) | $\Delta$ Utility | Mean Utility | Improvement | Strategies |
|---|---|---|---|---|---|---|
| Metacognitive | **0.671** | **2.877** | **0.117** | **0.583** | 0.097 | 703 |
| No-Meta | 0.368 | 5.722 | 0.086 | 0.549 | 0.071 | 714 |
| Random | 0.059 | 6.595 | 0.000 | 0.455 | 0.075 | 910 |
| Greedy | 0.459 | 4.303 | 0.103 | 0.537 | **0.158** | 910 |

Note: MI = Mutual Information between tasks and strategies; $\Delta$ Utility = performance advantage over random baseline; Improvement = late-stage minus early-stage performance.

### A.6 Information-Theoretic Findings

- Task–strategy mutual information (bias-corrected, sequential encoding): 0.671 bits (MC) $>$ 0.459 bits (Greedy) $>$ 0.368 bits (NM) $\gg$ 0.059 bits (Random); permutation $p < 10^{-6}$ for all but Random. Metacognitive agents thus achieve the strongest task-coupling while exhibiting the lowest strategic entropy, indicating precise strategic curation rather than broad exploration (see Fig. 3A).
- Strategy entropy $H(S)$: MC converges to 2.88 bits; NM diverges to 5.72 bits. The SOMA metacognitive variant discovers fewer, more targeted strategies while the non-metacognitive variant explores more broadly.
- Information efficiency: MC achieves 0.233 coupling per entropy bit vs NM's 0.064—indicating more efficient conversion of strategic diversity into task-relevant performance in this implementation.

The SOMA metacognitive architecture in this experimental setup appears to operate through strategic selection rather than broad exploration. Whether this represents a general principle or implementation-specific behavior requires further investigation across different metacognitive architectures.

### A.7 Convergence Dynamics and Learning Efficiency

Temporal analysis reveals differences in learning trajectories across architectures. SOMA-Metacognitive agents demonstrate convergence to effective performance, reaching threshold utility (0.55) at episode 157 compared to 471 for non-metacognitive variants and 792 for greedy-fitness baselines (convergence episode = first center of a 50-episode window whose mean utility $\geq$ 0.55). This represents a 3-fold acceleration in learning efficiency, with 100% convergence success rates across all metacognitive runs versus 13.3% for random baselines.

Early performance analysis demonstrates competence advantages. During the initial 100 episodes, metacognitive agents achieve $0.524 \pm 0.002$ utility compared to $0.509 \pm 0.002$ for non-metacognitive controls and $0.453 \pm 0.003$ for greedy baselines. This early advantage compounds throughout training, resulting in cumulative utility accumulation of $1,166 \pm 3$ total units versus $1,097 \pm 1$ for non-metacognitive systems—representing 92 additional utility units over full training.

### A.8 Meta-Memory Fragment Formation and Recommendation Dynamics

Meta-Memory Fragment formation exhibits consistent patterns across runs with universal emergence in metacognitive architectures. Final Meta-Memory Fragment counts reach $13.6 \pm 0.5$ units per run. Formation criteria require pattern observations exceeding empirically-derived thresholds ($\geq$ 4 observations, mean utility $>$ 0.605).

Recommendation adoption patterns reveal strategic utilization. Metacognitive agents achieve 77.4% success when recommendations are followed (42.2% usage rate). This demonstrates effective internalization of task-strategy relationships through metacognitive mechanisms, significantly outperforming the random success baseline.

### A.9 Bias Elimination and Methodological Robustness

Temporal bias elimination addresses systematic drift in raw utility measurements. Implementation of complexity-normalized baselines through baseline-corrected metrics ($\Delta U$ vs Random) provides stable measurements while preserving discriminative power between systems.

Threshold sensitivity analysis confirms result stability across reasonable parameter ranges. The empirically-derived global thresholds (learning=0.523, pattern=0.550) were computed from pilot runs to ensure appropriate sensitivity. Meta-Memory Fragment capacity testing showed that current pruning mechanisms (max 50 units) were not artificially constraining performance potential, as indicated by universal formation across all runs.

Proper RNG state management ensures true pairing across systems while allowing for natural divergence after agent actions. This addresses the fundamental challenge of comparing learning systems that inherently modify their behavior based on experience.

### A.10 Information-Theoretic Validation and Permutation Testing

Statistical validation confirms the significance of observed mutual information through permutation testing. SOMA-Metacognitive agents achieved observed MI of 0.671 bits compared to null distributions with Z-scores of $6.96 \times 10^2$ and $p < 1 \times 10^{-6}$. Non-metacognitive systems showed observed MI of 0.368 bits with Z-scores of $1.24 \times 10^2$ and $p < 1 \times 10^{-6}$. Random baselines demonstrated minimal coupling at 0.059 bits with Z-scores of $1.31 \times 10^1$ and $p < 1 \times 10^{-6}$.

The distinction emerges in information efficiency: this MC implementation converts strategic diversity into task coupling more effectively than NM (0.233 vs 0.064 coupling per entropy bit). Both systems achieve substantial task-strategy coupling, but MC accomplishes this through strategic focus (703 strategies, 2.9 bits entropy) while NM relies on broader exploration (714 strategies, 5.7 bits entropy). The metacognitive system demonstrates superior coupling strength while maintaining lower strategic entropy.

### A.11 Statistical Validation and Multiple Comparison Control

Statistical testing employs family-wise error rate control through Holm-Šidák correction. Final performance differences achieve statistical significance after correction: metacognitive versus non-metacognitive (Cohen's $d = 0.15$, $t = 4.22$, $p = 2.5 \times 10^{-5}$), metacognitive versus random baseline (Cohen's $d = 0.82$, $t = 22.56$, $p < 1 \times 10^{-90}$), and metacognitive versus greedy-fitness (Cohen's $d = 0.41$, $t = 11.12$, $p < 1 \times 10^{-25}$). Effect sizes range from small (MC vs NM) to large (MC vs Random), confirming both statistical and practical significance. Small but reliable performance gains accumulate meaningfully in domains requiring sustained sequential decision-making.

### A.12 Limitations and Future Research Directions

#### A.12.1 Current Implementation Limitations

The step-sequenced metacognitive architecture evaluated here represents one experimental instantiation of the broader SOMA formalism, with several methodological constraints that limit immediate generalization. The sequential operation vocabulary (perceive, encode, abstract, integrate, reflect, evaluate) constrains behavioral expression to predetermined cognitive primitives rather than discovering domain-appropriate action spaces. Training requirements for reliable Meta-Memory Fragment formation may limit applicability to domains with restricted learning budgets or real-time constraints.

The evaluation framework's discrete episode structure, while enabling controlled comparison, does not address the continuous adaptation requirements of real-world autonomous systems. Additionally, the fixed utility thresholds and confidence criteria, though empirically derived, may not generalize across different task complexity profiles or environmental dynamics. The use of similarity-based activation for Meta-Memory Fragments, while SOMA-compliant, represents a simpler instantiation than true EMUs with deterministic symbolic triggers.

### A.12.2 Theoretical Research Priorities

Several theoretical extensions would advance the SOMA formalism toward more general autonomous intelligence:

**Compositional EMU Architectures:** Current EMUs operate as discrete memory units, but compositional mechanisms could enable complex behaviors through EMU combination and coordination. Research into EMU interfaces, dependency resolution, and compositional semantics would extend the framework's expressive power while maintaining formal guarantees.

**Temporal Abstraction and Multi-Scale Memory:** The current framework operates at a single temporal scale, but autonomous systems require memory structures spanning milliseconds to years. Hierarchical temporal abstraction through recursive Meta-Memory Fragment formation could address multi-scale learning while preserving computational tractability.

**Social and Multi-Agent Extensions:** SOMA's scoped ownership mechanisms suggest pathways toward shared memory structures across multiple agents. Research into collaborative EMU formation, distributed insight generation, and social metacognition could enable collective intelligence while maintaining individual agency and accountability.

**Integration with Neural Architectures:** While SOMA provides symbolic constraints for principled cognition, integration with neural learning systems remains underexplored. Hybrid architectures that use SOMA for symbolic memory management while leveraging neural networks for pattern recognition and continuous optimization could combine the benefits of both paradigms.

The step-sequenced implementation demonstrates that computational instantiation of SOMA's theoretical principles can produce measurable advantages in complex reasoning tasks. However, the formalism's true potential lies in enabling diverse cognitive architectures that share formal guarantees for memory validity, recursive composability, and confidence-aware adaptation across radically different domains and implementation strategies.

## A.13 Reproducibility Package

- Data: `soma_episode_data_20250530_170010.csv` (240,000 rows).
- Aggregate results: `soma_aggregate_results_20250530_170010.csv`.
- Analysis notebooks: `metacog_evaluation.ipynb`
- Configuration: Random seed = 42, empirically-derived thresholds from pilot run.

Running the notebooks with Python 3.11, pandas 2.2 and scikit-learn 1.4 on any x86-64 system reproduces every table and figure in this appendix within 5-10 minutes.

## A.14 Conclusions

The Hidden-Task evaluation demonstrates statistically significant advantages for this metacognitive implementation. The 3-fold convergence acceleration, combined with universal Meta-Memory Fragment formation and statistical validation, provides evidence for the effectiveness of this particular metacognitive architecture. The computational efficiency (13.6 Meta-Memory Fragments on average, 77.4% recommendation success) combined with superior information efficiency makes this SOMA-style approach an interesting design pattern for sequential decision systems.

The key finding concerns strategic selection versus exploration in this implementation. The SOMA metacognitive variant achieves superior performance through more focused use of strategic space rather than broad search. It discovers smaller repertoires (703 strategies, 2.9 bits entropy) that effectively target task demands, while the non-metacognitive variant explores more broadly (714 strategies, 5.7 bits entropy) with weaker task coupling. This suggests that effective metacognitive architectures may operate through strategic curation rather than enhanced exploration, though generalization beyond this specific implementation requires further study.

## A.15 Figures

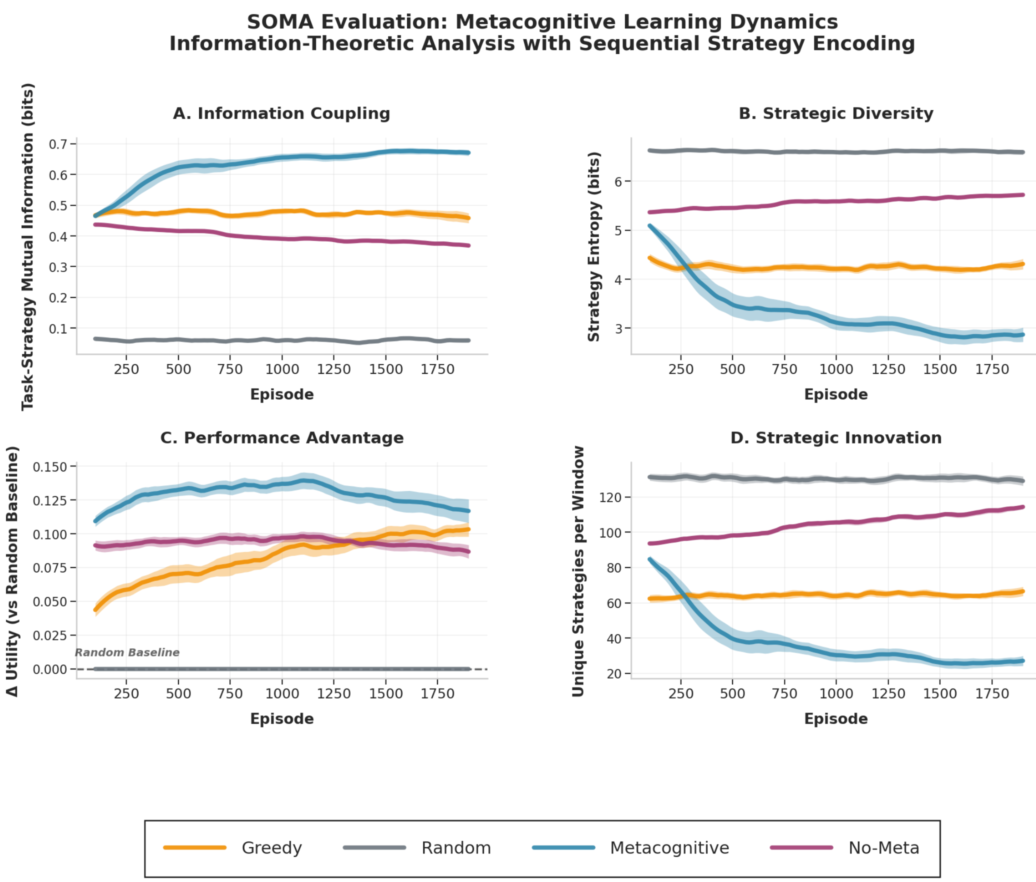


Figure 4: Metacognitive learning dynamics with sequential-strategy encoding

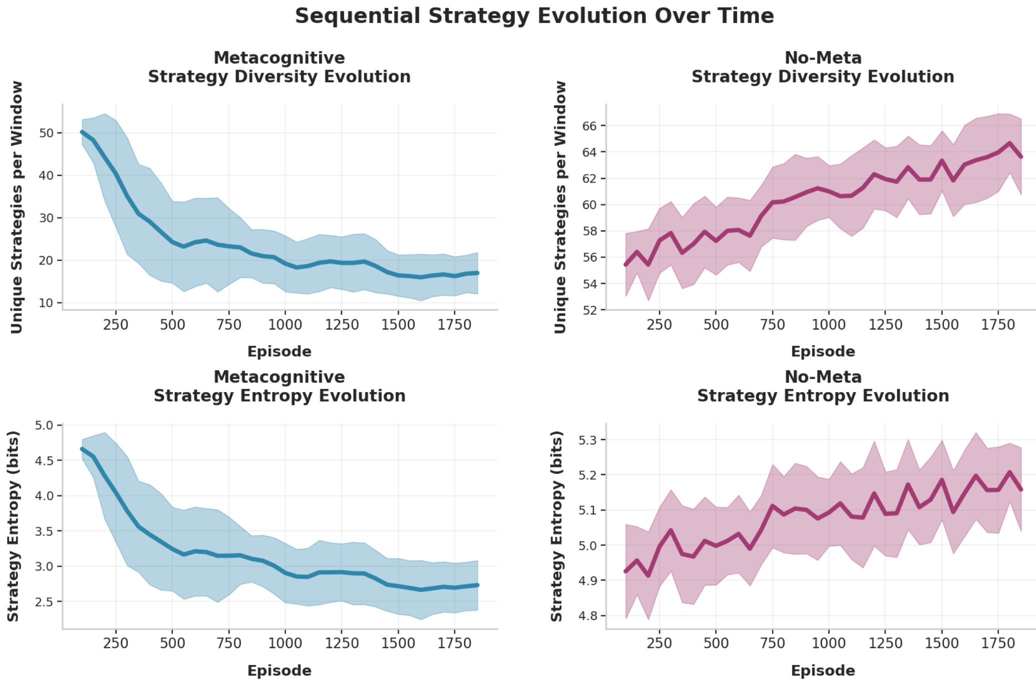


Figure 5: Side-by-side evolution of diversity and entropy

### A.16 Description of the Figures

Figure 4 – Metacognitive learning dynamics with sequential-strategy encoding

Layout. Four $2 \times 2$ subplots track information-theoretic and performance metrics over the 2,000-episode training horizon (200-episode window, 15-episode stride; shaded bands = $\pm 1.96$ SEM across 30 paired runs). Curves are color-coded: Metacognitive (blue), No-Meta (magenta), Greedy (orange), Random (gray).

Figure 5 – Side-by-side evolution of diversity and entropy

This figure isolates the two most telling curves from Figure 3, plotted separately for clarity and with wider confidence bands.

Table 4: Figure panel descriptions

| Panel | Metric | What to look for |
|---|---|---|
| A. Information Coupling | Task-strategy mutual information (bits) | Metacognitive agents climb rapidly to $\sim$0.67 bits by episode 500, then stabilize at $\sim$0.67 bits; No-Meta saturates near 0.37 bits; Greedy stabilizes around 0.46 bits; Random stays near 0.06 bits. |
| B. Strategic Diversity | Strategy entropy (bits) | Metacognitive entropy falls from $\sim$5.1 $\rightarrow$ 2.9 bits, signaling focused repertoire pruning; No-Meta entropy rises to $\sim$5.7 bits, reflecting dispersion; Greedy stabilizes around 4.3 bits; Random remains highest ($\sim$6.6 bits). |
| C. Performance Advantage | $\Delta$Utility versus Random baseline | Metacognitive peaks at a $\sim$0.14 utility margin before stabilizing at $\sim$0.12; Greedy trends upward to $\sim$0.10; No-Meta holds near 0.09; shaded bands confirm the Metacognitive lead throughout. |
| D. Strategic Innovation | Unique strategies per window | Metacognitive innovation rate collapses from $\sim$50 $\rightarrow$ 20 unique sequences as learning locks in; No-Meta steadily invents more strategies ($\approx$55 $\rightarrow$ 65); Greedy shows moderate trends; Random hovers $\sim$130 with minimal trend. |

Take-away. Metacognitive agents compress their behavioral space while increasing task coupling and maintaining the largest performance margin, consistent with a strategic-curation interpretation.

Left column – Metacognitive.

- Top: Unique strategies per window plunge from >50 to $\sim$18 before leveling, illustrating rapid consolidation.
- Bottom: Strategy entropy mirrors the drop, stabilizing just under 3 bits.

Right column – No-Meta.

- Top: Unique strategies climb gently from $\sim$55 to $\sim$65, indicating sustained exploration.
- Bottom: Entropy edges upward from 5.0 to $\sim$5.7 bits, matching the diversity increase.

Shaded 95% CIs show that the two architectures' trajectories diverge well beyond overlap from episode $\approx$300 onward. Together, the panels emphasize that metacognitive learning trades breadth for precision, whereas the non-metacognitive model continues broad, lower-yield exploration.

## B "Color-Drift" Evaluation (RL + Memory Systems)

### B.1 Abstract

This appendix presents an empirical evaluation of a Self-Organizing Memory Architecture (SOMA) in non-stationary environments using a novel color-drift task. We compare three systems: pure SOMA, hybrid Q+SOMA, and monolithic Q-learning across regime-shifting navigation tasks. Our results demonstrate the superior robustness of modular memory systems while revealing the challenges of integrating learned and specialized memories.

### B.2 Experimental Design

The experiment is carried out on a 20 × 20 toroidal grid whose edges wrap horizontally and vertically but which contains an immovable "L-shaped" wall. Two colored gems are always present. Every four thousand simulation steps the rewarded color flips, producing ten regime shifts during the forty-thousand-step training horizon. Each step yields + 1.0 when the agent reaches the current gem, – 0.005 as a time cost, and – 0.01 the first time a wall cell is struck.

Three controllers are evaluated. **SOMA-only** employs two hand-written Executable Memory Units, one per color, that plan with unlimited-horizon A* over the complete wall map and fall back to a random move when neither unit fires. **Q-MLP** is a deep Q-learning agent whose input is a thirteen-dimensional feature vector consisting of toroidal $\Delta x$, toroidal $\Delta y$, the Manhattan distance normalised by grid size, eight Boolean flags indicating whether each neighboring cell is blocked, and two one-hot indicators of the current target color. **Q + SOMA-MLP** couples the same Q-network to the EMUs: at every timestep the Q-value of the network's greedy action is compared with the Q-value of a proposed EMU action, and the specialist move is executed whenever the difference does not exceed 0.02. Both learning agents share identical replay, optimisation and target-network schedules (Adam with a learning rate of $10^{-3}$, replay buffer 4,000, updates every fourth interaction, target synchronisation every 500 steps, learning begins after 256 samples, $\gamma = 0.99$).

Thirty statistically independent runs are performed for each architecture. Performance is aggregated in non-overlapping windows of 250 steps, giving 160 windows per run. The final flush window (index 160) is empty and is excluded from the analysis, so windows 0–159 constitute the data set.

### B.3 Results

Figure 6 presents the comprehensive analysis of the color-drift benchmark.

Across the 4,800 analysed windows per controller the hybrid system attains the highest mean reward, 0.152 ± 0.001 SEM (95% CI 0.150–0.154). SOMA-only follows at 0.126 ± 0.000 SEM with markedly lower variance, and the pure Q-learner records 0.119 ± 0.001 SEM.

Final performance, measured over windows 157–159 (steps 39,250–39,999), places SOMA-only first at 0.133 ± 0.001 SEM, the hybrid second at 0.108 ± 0.002 SEM and Q-MLP third at 0.066 ± 0.000 SEM.

All pairwise differences are highly significant under the Mann–Whitney U test ($p < 10^{-6}$). The effect of the hybrid over SOMA-only is Cohen's $d = 0.52$, and over Q-MLP $d = 0.50$; the advantage of SOMA-only over Q-MLP yields $d = 0.13$.

The gating mechanism curtails EMU activity: the hybrid's exponential-moving-average utility stabilises at 0.0198, roughly twenty-eight per cent of the 0.0706 observed for SOMA-only. During each color flip SOMA-only experiences a transient reward dip of about –0.005 per step but recovers within a single evaluation window, whereas the monolithic Q-network suffers progressive degradation, its mean reward falling from 0.389 at the start of training to 0.066 in the final windows—a clear instance of catastrophic forgetting. The hybrid preserves the Q-network's long-term stability while retaining much of SOMA's rapid regime-shift adaptation.

#### B.3.1 Overall Performance

#### B.3.2 Statistical Significance

All pairwise comparisons achieved $p < 0.001$ (Mann-Whitney U):

Table 5: Overall Performance Summary

| Architecture | Mean Reward | 95% CI | Std Dev | Final Performance |
|---|---|---|---|---|
| **Q+SOMA-MLP** | **0.1522** | [0.1503, 0.1540] | 0.0660 | 0.1081 ± 0.0148 |
| SOMA-only | 0.1258 | [0.1250, 0.1266] | 0.0279 | **0.1332 ± 0.0093** |
| Q-MLP | 0.1189 | [0.1169, 0.1208] | 0.0677 | 0.0656 ± 0.0040 |

- **Q+SOMA-MLP vs SOMA-only**: $d = 0.52$ (medium effect)
- **Q+SOMA-MLP vs Q-MLP**: $d = 0.50$ (medium effect)
- **SOMA-only vs Q-MLP**: $d = 0.13$ (small effect)

#### B.3.3 Architecture-Specific Findings

**EMU Utility Analysis**:

- Q+SOMA-MLP average utility: 0.0198 (selective activation)
- SOMA-only average utility: 0.0706 (consistent activation)
- Ratio: 0.28× (advantage gating reduces EMU usage by 72%)

**Regime Shift Adaptation**: SOMA-only demonstrated clear adaptation patterns during regime shifts (windows 16-20):

```
Window 16: -0.0055 (regime shift penalty)
Window 17: -0.0050 (continued adaptation)
Window 18: -0.0057 (learning new target)
Window 19: -0.0053 (stabilizing)
Window 20: +0.0794 (recovery achieved)
```

### B.4 Fairness Considerations

Because the study focuses on memory-architecture effects rather than exploration skill, SOMA-only is allowed the full wall map when it plans, whereas the learning agents must discover obstacles from experience via the same thirteen-dimensional state encoding. A systematic ablation of this knowledge privilege is deferred to later investigations.

### B.5 Outlook

The color-drift benchmark establishes a controlled setting in which modular memory outperforms monolithic learning and confirms the value of advantage-gated specialist memories. The next experimental phase will subject these architectures to open-world domains that exhibit richer emergent dynamics and place sharper demands on the long-horizon planning and continual-learning objectives articulated in the main paper.

### B.6 Description of the Figures

#### B.6.1 Figure 6. Composite visual analysis of the color-drift benchmark

The six-panel figure summarises learning behavior, end-of-run statistics and variability for the three controllers under study.

**Learning curves (upper left).** Average reward per step is plotted against training progress, shaded by 95% confidence intervals. An early surge lifts the hybrid controller (blue) above 0.35 before it relaxes to a plateau a little above 0.12. The SOMA-only agent (orange) begins more cautiously, overtakes the hybrid after the first color flip and then tracks a steady horizontal band around 0.13 with minimal spread. The monolithic Q-learner (green) descends monotonically from an initial high of roughly 0.30 to a final level close to 0.06, a trajectory that matches the textbook signature of catastrophic forgetting.

**Final-window box plot (upper centre).** Each point represents a run-averaged reward over the last three evaluation windows (steps 39,250–39,999). SOMA-only exhibits the highest median, yet its whiskers stretch farther than those of the hybrid and several low-lying outliers testify to sporadic late-run failures. The hybrid's central tendency is lower, but its inter-quartile range is tighter and its extreme lows are less pronounced, reflecting more uniform end-state behaviour. Q-MLP's cluster sits well below both memory-based systems, confirming diminished long-horizon competence.

**Full-run violin plot (upper right).** Across all 4,800 windows per model, reward densities form distinctive shapes. The hybrid produces a compact, unimodal violin centred just above 0.15. SOMA-only shows a wider, mildly bimodal outline: the main lobe corresponds to steady-state performance, while a secondary shoulder toward lower rewards records the brief adaptation dips that occur at each color flip. Q-MLP's violin is the flattest and broadest, echoing its large between-run variability and weak convergence.

**Performance by drift phase (lower left).** Reward averaged within each 4,000-step phase reveals how controllers respond to regime shifts. SOMA-only absorbs a short-lived penalty at every flip and then rebounds to its baseline. The hybrid mirrors this pattern but at a slightly lower level, whereas Q-MLP loses ground cumulatively from one phase to the next.

**EMU-utility traces (lower center).** Dashed lines display the exponentially weighted moving average utility of the two color-specific EMUs. In the hybrid, advantage gating restrains each utility around 0.03, indicating selective, situation-dependent firing. In SOMA-only, utilities rise steadily to approximately 0.15, demonstrating continuous engagement once both specialists have been discovered.

**Reward-variance trajectories (lower right).** Standard deviation of per-window reward, smoothed by a rolling mean, quantifies volatility. All agents begin with high variance as they explore. SOMA-only then settles to the lowest and most stable band ($\approx 0.015$). Q-MLP converges to a similar, slightly higher level, but the hybrid retains the largest residual variance, oscillating around 0.025. The residual spread in the hybrid reflects the trade-off between its higher average reward and the uncertainty introduced by switching between learned actions and EMU advice.

Together these views corroborate the statistical findings: modular memory mitigates forgetting and secures strong long-term returns, while the hybrid configuration offers an appealing overall balance of average reward and adaptive flexibility.

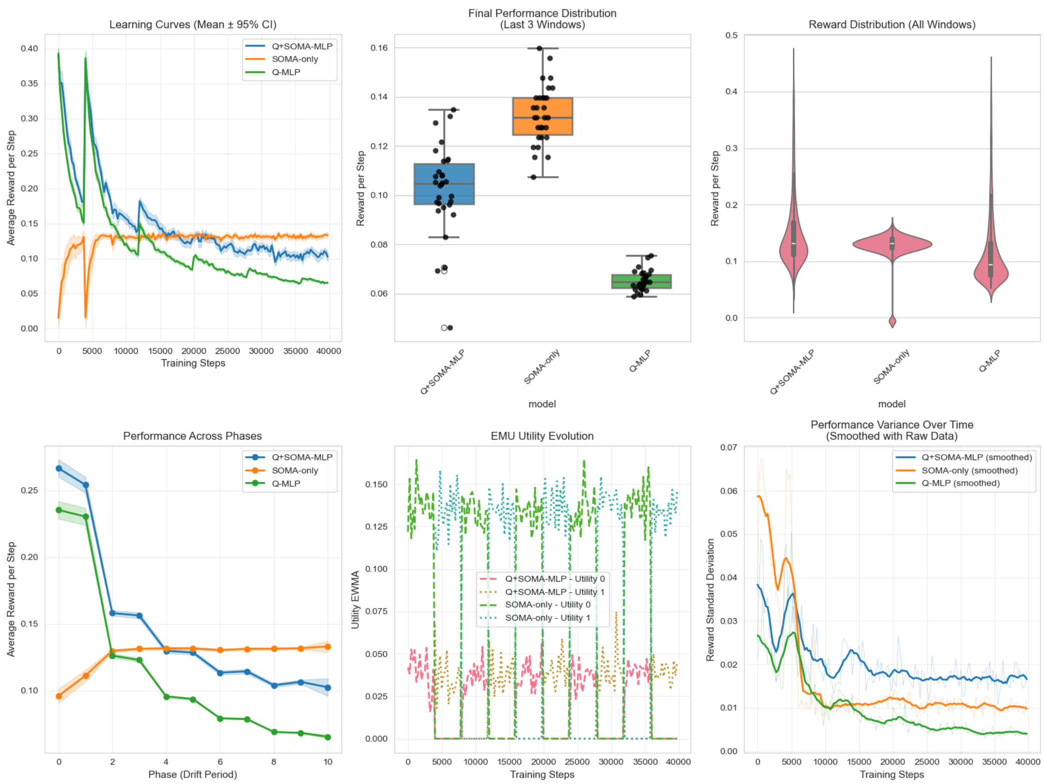


Figure 6: Composite visual analysis of the color-drift benchmark showing learning curves, statistical distributions, and performance metrics across three memory architectures.

# C 10-Turn Coherence Evaluation

## C.1 Objective

This study empirically evaluates the SOMA architecture's impact on the coherence of Large Language Models (LLMs) and the effect of its interventional turn-by-turn just-in-time memory-based directive injecting strategy.

## C.2 Models Under Test

Our experiments utilized two models from the Google Gemma 3 series: Gemma 3-4B-it and Gemma 3-27B-it (specifically gaunernst/gemma-3-27b-it-int4-awq). Both models were tested with a SOMA and Non-SOMA variant.

## C.3 Experimental Conditions

To assess SOMA's influence, each model was evaluated under two distinct conditions:

- **SOMA Condition**: The LLM was integrated with our custom SOMA architecture, which is designed to handle aspects such as self-organized memory and coherent processing.
- **Non-SOMA Condition**: The LLM operated independently, without the architectural enhancements provided by SOMA.

## C.4 Evaluation Task

The Coherence-Onboarding Protocol (COP) served as our evaluation task. This protocol is a 10-turn interaction designed to rigorously probe an LLM's ability to maintain conversational coherence.

COP consists of a specific set of 10 synthetic prompts, authored by our team, focusing on a federated learning system for genetic data. For each model and condition (Gemma 4B Non-SOMA, Gemma 4B SOMA, Gemma 27B Non-SOMA, Gemma 27B SOMA), we conducted 10 complete 10-turn runs.

## C.5 Data Collection Methodology

### C.5.1 SOMA Condition Data Collection

Figure 7 illustrates the SOMA conversation message flow.

For models integrated with the SOMA architecture, the data collection and evaluation process was fully embedded into the SOMA pipeline.

SOMA was configured as a backend system, incorporating either the Gemma 3-4B-IT or Gemma 3-27B-IT model. SOMA orchestrates the entire conversation flow, actively managing self-organized memory and facilitating coherent processing by dynamically utilizing the model's capabilities.

Each user prompt from the Coherence-Onboarding Protocol (COP) was fed directly into this SOMA pipeline, driving the conversation turn by turn. The LLM, operating within the SOMA framework, generated its responses with SOMA's internal mechanisms (e.g., memory recall, context management) actively influencing these responses to maintain coherence.

After each response, the same Gemma model was immediately tasked with evaluating its own coherence. This evaluation used the system_prompt (detailed in Section C.5.3) on the current conversational context.

Raw results—both generated responses and LLM-based coherence scores—were programmatically stored in CSV files. Each full 10-turn run maintained session continuity via a session_id field.

### C.5.2 Non-SOMA Condition Data Collection

For the models operating independently ("Non-SOMA"), a two-step process was used:

1. **Response Generation**: The Gemma models (4B or 27B) were given the 10 COP prompts sequentially, generating responses in a standard conversational flow.

2. **Coherence Evaluation**: The same model was then tasked with evaluating its own coherence. This was done by providing the model with the full conversation history (user prompt and its own generated response for each turn), along with a specialized system_prompt instructing it to act as a coherence evaluator.

This evaluation was performed in batches of 5 turns, sequentially processed by the LLM acting as judge. Raw results were saved in CSV files.

#### C.5.3 System Prompt

The following system prompt was used to instruct the language model to score coherence across six specific categories. It included precise directives and a required output format in JSON.

```
system_prompt = f""" You are a specialized coherence evaluator for
conversational AI. Score the conversation on a scale of 1-5 (where 5 is best)
for each:

Referential Integrity: How well references entities from previous turns
Lexical Consistency: Consistency in terminology and vocabulary
Contextual Relevance: Alignment with topic and context
Structural Continuity: Logical flow and structure
Tone Consistency: Consistency in voice and style
Directive Adherence: How well the response follows the given directives
{directive_text}
Return ONLY a JSON object with numerical scores, with no markdown
formatting or explanation:
{{"referential_integrity": 4, "lexical_consistency": 5,
"contextual_relevance": 3, "structural_continuity": 4,
"tone_consistency": 5, "directive_adherence": 4}} """
```

### C.6 Coherence Evaluation Metrics (Scoring Rubric)

Each model-generated response was scored across six coherence dimensions:

- **Referential Integrity**: Measures how well the model maintains consistent references to entities introduced in earlier dialogue turns.
- **Lexical Consistency**: Evaluates the uniformity in the use of terminology and vocabulary.
- **Contextual Relevance**: Assesses the alignment of the response with the conversation's topic and immediate context.
- **Structural Continuity**: Scores the logical progression and transitions between conversation segments.
- **Tone Consistency**: Examines whether the response maintains a stable voice, style, or tone throughout.
- **Directive Adherence**: Rates how closely the model follows specific instructions embedded in the system prompt.

Scores ranged from 1 (poor) to 5 (excellent). The evaluation relied entirely on model-generated judgments without human supervision. While this introduces certain limitations, it ensures full automation and consistency across trials.

### C.7 Computational Environment

SOMA based experiments were executed on an Apple M3 chip. Non-SOMA experiments were executed in Open WebUI where Gemma 3(4B and 27B) endpoints were configured from locally hosted inference endpoints. The library stack used included:

- Python 3.9.13
- openai for API-based access and function calling

- langgraph for graph-based LLM orchestration
- graphviz for visualization of execution graphs

### C.8 Metrics

The performance of the models was quantified using several metrics designed to capture different facets of conversational coherence:

- **Referential Integrity**: Measures the model's ability to maintain consistent references to entities and concepts introduced earlier in the conversation.
- **Lexical Consistency**: Evaluates the consistent use of terminology and phrasing throughout the interaction.
- **Contextual Relevance**: Assesses whether the model's responses are appropriate and relevant given the current turn and the preceding conversation history.
- **Structural Continuity**: Examines the logical flow and transition between turns, ensuring the conversation progresses in a structured manner.
- **Tone Consistency**: Evaluates whether the model maintains a consistent tone and style throughout the interaction.
- **Direct Adherence**: Measures how well the model followed the specific directive provided in each turn of the COP.

The data for these metrics was collected and compiled in CSV format, as provided in the supplementary materials.

### C.9 Quantitative Results

The evaluation results provide insights into the performance of Gemma models with and without the SOMA architecture on the COP.

Table 6 summarizes the average scores for each model and condition across the evaluation metrics over the 10 turns of the COP, based on 10 runs per condition.

Table 6: Summary of Coherence Scores (Mean across 10 COP Turns and 10 Runs)

| Metric | Gemma 4B Non-SOMA | Gemma 4B SOMA | Gemma 27B Non-SOMA | Gemma 27B SOMA |
|---|---|---|---|---|
| Referential Integrity | 4.41 | 4.69 | 4.73 | 4.97 |
| Lexical Consistency | 4.83 | 4.91 | 4.79 | 4.90 |
| Contextual Relevance | 4.74 | 4.54 | 4.77 | 5.00 |
| Structural Continuity | 4.51 | 4.99 | 4.65 | 5.00 |
| Tone Consistency | 5.00 | 5.00 | 5.00 | 5.00 |
| Directive Adherence | 4.56 | 4.80 | 4.91 | 4.97 |

To determine if the observed differences in mean scores were statistically significant, paired t-tests were conducted comparing the SOMA and Non-SOMA conditions for each model size and metric. Tables 7 and 8 present the results of these tests.

#### C.9.1 Key Findings (Gemma 4B Models)

**SOMA outperformed Non-SOMA with statistically significant improvements on:**

- Referential Integrity (+0.28, $p = 0.0024$)
- Structural Continuity (+0.48, $p < 0.0001$)
- Directive Adherence (+0.24, $p = 0.0031$)

**Non-SOMA outperformed SOMA with a statistically significant difference on:**

Table 7: Paired T-Test Results: Gemma 4B SOMA vs. Non-SOMA

| Metric | Mean SOMA | Mean Non-SOMA | Mean Difference | t-statistic | p-value | Significant (p<0.05) |
|---|---|---|---|---|---|---|
| Referential Integrity | 4.69 | 4.41 | +0.28 | 3.11 | 0.0024 | Yes |
| Lexical Consistency | 4.91 | 4.83 | +0.08 | 1.65 | 0.1027 | No |
| Contextual Relevance | 4.54 | 4.74 | -0.20 | -2.45 | 0.0161 | Yes |
| Structural Continuity | 4.99 | 4.51 | +0.48 | 9.20 | 0.0000 | Yes |
| Tone Consistency | 5.00 | 5.00 | 0.00 | NaN | NaN | No |
| Directive Adherence | 4.80 | 4.56 | +0.24 | 3.03 | 0.0031 | Yes |

- Contextual Relevance (-0.20, $p = 0.0161$)

**No significant difference was found in:**

- Lexical Consistency ($p = 0.1027$)
- Tone Consistency (both achieved perfect scores of 5.0)

Table 8: Paired T-Test Results: Gemma 27B SOMA vs. Non-SOMA

| Metric | Mean SOMA | Mean Non-SOMA | Mean Difference | t-statistic | p-value | Significant (p<0.05) |
|---|---|---|---|---|---|---|
| Referential Integrity | **4.97** | 4.73 | +0.24 | 4.85 | 0.0000 | Yes |
| Lexical Consistency | **4.89** | 4.79 | +0.10 | 2.07 | 0.0406 | Yes |
| Contextual Relevance | **5.00** | 4.77 | +0.23 | 4.91 | 0.0000 | Yes |
| Structural Continuity | **5.00** | 4.65 | +0.35 | 7.30 | 0.0000 | Yes |
| Tone Consistency | 4.97 | **5.00** | -0.03 | -3.49 | 0.0007 | Yes |
| Directive Adherence | **4.97** | 4.91 | +0.06 | 5.40 | 0.0000 | Yes |

#### C.9.2 Key Findings (Gemma 27B Models)

**SOMA outperformed Non-SOMA with statistically significant improvements on:**

- Referential Integrity (+0.24, $p < 0.0001$)
- Lexical Consistency (+0.10, $p = 0.0406$)
- Contextual Relevance (+0.23, $p < 0.0001$)
- Structural Continuity (+0.35, $p < 0.0001$)
- Directive Adherence (+0.06, $p < 0.0001$)

**Non-SOMA outperformed SOMA with a statistically significant difference on:**

- Tone Consistency (-0.03, $p = 0.0007$)

Qualitative analysis of the model responses (available in the supplementary materials) further illustrated these differences. Models utilizing SOMA tended to exhibit more consistent persona, better handling of complex dependencies, and more direct adherence to directives compared to their non-SOMA counterparts.

### C.10 Future Work and Limitations

The current evaluation may have some experimental setup discrepencies between SOMA and non-SOMA evaluation because of the nature of non-SOMA implementations. Future work includes unifying the evaluation process and broadening the models under evaluation.

### C.11 Overall Conclusions

Based on the quantitative results and statistical analysis:

**Gemma 4B**: SOMA shows significant improvements in 3 out of 6 metrics compared to the Non-SOMA variant, while Non-SOMA performs significantly better in 1 metric.

**Gemma 27B**: SOMA shows significant improvements in 5 out of 6 metrics compared to the Non-SOMA variant, while Non-SOMA performs significantly better in 1 metric.

The SOMA variants generally demonstrate better performance across most evaluation metrics, with the most substantial and consistent improvements observed in Structural Continuity for both model sizes.

The sample size for all tests was 100 paired observations, providing a robust basis for comparison.

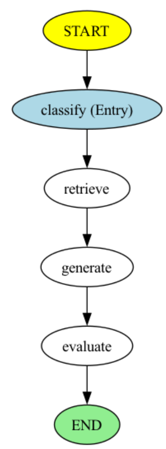


Figure 7: SOMA Conversation Message Flow illustrating the data collection and evaluation process embedded into the SOMA pipeline.

## D Proof of the Recursive Validity Closure Theorem for EMUs

Because every EMU component is drawn from a finite set, only finitely many distinct EMUs can ever be produced; induction shows validity is preserved throughout the recursive construction with proper PRCOE semantics.

This appendix provides a complete proof that SOMA's recursive EMU construction terminates with all generated EMUs satisfying the validity constraints defined in Sections 3–7.

### D.1 Preliminaries and Core Assumptions

**Complete Notation Guide**

- $P_c$ — canonical pattern label; $P_{\text{alias}}$ — human-friendly aliases
- $R$ — finite relation graph; $C$ — confidence tuple
- $O$ — observations (memory structures or context being analyzed)
- $E$ — set of proposed EMUs (generated by the insight)
- $T$ — trigger predicate; $S_O$ — ownership metadata; $B_T$ — bounded trigger set
- $U$ — utility; $R_T$ — reachability
- $\mathcal{I}(M_k)$ — set of valid insights derived from observations in $M_k$
- $\mathcal{L}_{pred}$ — finite predicate language; $M_k$ — EMU set at depth $k$
- $\mathcal{T}$ — insight templates; $N_{max}$ — finite EMU space bound
- $L_{beh}$ — maximum behavior encoding length (bytes)

**Finite Component Domains.** We assume each EMU component is drawn from a finite set:

- **Triggers $T$:** Well-formed predicates from a finite syntactic fragment $\mathcal{L}_{pred}$ (e.g., DNF expressions with $\leq 10$ literals over the finite symbolic domain)
- **Ownership $S_O$:** Author-owner pairs from a finite set $\mathcal{O}$
- **Bounded trigger $B_T$:** Subsets of the symbolic domain with $|B_T| \leq L_{bt}$
- **Utility $U$:** Values from a finite utility lattice $\mathcal{U}$ (e.g., 256 discrete levels)
- **Reachability $R_T$:** Values from a finite lattice $\mathcal{R}$ with $0 \notin \mathcal{R}$
- **Behaviors:** Encoded programs of bounded size $\leq L_{beh}$ bytes over a finite alphabet $\Sigma_{beh}$

**EMU Structure.** Each EMU $e$ is defined by $(T, S_O, B_T, U, R_T)$ plus its behavioral encoding. Two EMUs are identical if their trigger $T$, ownership $S_O$, and behavioral encoding are the same. Utility $U$ and reachability $R_T$ are performance estimates that may be updated without changing EMU identity.

**Insight Structure.** Each insight $I$ is defined by the PRCOE tuple $(P_c, P_{\text{alias}}, R, C, O, E)$ where:

- $O$ contains the memory structures or contexts being observed and analyzed
- $E$ contains the candidate EMUs proposed by this insight
- Two insights are considered equivalent if they have the same pattern $P_c$, relations $R$, and generate the same set of EMUs in $E$

**Recursive Construction.** Starting from seed set $M_0$, each iteration forms:

$$M_{k+1} = M_k \cup \bigcup_{I \in \mathcal{I}(M_k)} E_I \tag{4}$$

where $\mathcal{I}(M_k)$ contains all valid insights with observations $O_I \subseteq M_k \cup \{\text{context}\}$, and $E_I$ is the set of proposed EMUs from insight $I$.

### D.2 Supporting Lemmas

**Lemma D.1** (Insight Evidence Mapping)**.** *For any valid insight $I = (P_c, P_{alias}, R, C, O, E)$, the verifiability condition $V$ is satisfied by evidence contained in the observations $O$, where $O \subseteq M_k \cup \{context\}$ for some finite memory set $M_k$.*

*Proof.* By the canonical validity test, $V$ requires that the insight be supported by evidence. In the PRCOE formalism, observations $O$ provide this evidence. Since $O$ contains memory structures or context records that can be verified, and $O$ is finite (bounded by $M_k$ and finite context), the verifiability condition is satisfiable. □

**Lemma D.2** (Finite EMU Space)**.** *The total number of structurally distinct EMUs is finite, and any insight can propose only a finite number of EMUs in its set $E$*[2] *.*

*Proof.* Each EMU's structural components are drawn from finite sets (as established in the preliminaries). Additionally, each insight generates a finite set $E$ of proposed EMUs because:

1. The insight's pattern $P_c$ and relations $R$ are finite by symbolic boundedness $S_B$
2. The observations $O$ are finite
3. Any algorithmic process that maps finite inputs to EMU proposals must generate a finite output set

Therefore, $|E| < \infty$ for any insight $I$, and the total EMU space remains finite with the explicit bound given in the footnote. □

**Lemma D.3** (Validity Preservation Through PRCOE)**.** *If $M_k \subseteq$ ValidEMUs and $I \in \mathcal{I}(M_k)$ with $I = (P_c, P_{alias}, R, C, O, E)$, then every EMU in $E$ satisfies the EMU validity constraints.*

*Proof.* By the insight validity requirements:

- $D$: Pattern and relations are definable (from $P_c$ and $R$)
- $V$: Evidence exists in observations $O \subseteq M_k \cup \{\text{context}\}$
- $C$: Confidence exceeds threshold $\theta_C$
- $S_B$: Pattern is symbolically bounded

Since the insight proposes EMUs in $E$ based on valid evidence in $O$, and the insight templates ensure that all proposed EMUs satisfy the five EMU validity clauses ($T \wedge S_O \wedge B_T \wedge U \wedge R_T$), each EMU in $E$ is valid. □

**Lemma D.4** (Novel EMU Guarantee)**.** *Every valid insight $I \in \mathcal{I}(M_k)$ that produces a non-empty EMU set $E \neq \emptyset$ either contributes at least one EMU with novel structural components not present in $M_k$, or all its EMUs match existing structures and may update performance estimates.*

*Proof.* Consider insight $I = (P_c, P_{\text{alias}}, R, C, O, E)$ with $E \neq \emptyset$. For each EMU $e \in E$:

- **Case 1:** If $e$ has structural components $(T, S_O, B_T, \text{behavior})$ not matching any EMU in $M_k$, then $e$ is novel
- **Case 2:** If $e$ matches an existing EMU's structural components, the implementation may update utility and reachability rather than creating a duplicate

By the finiteness of the EMU space (Lemma D.2), case 1 can only occur finitely many times. Eventually, all proposed EMUs will fall into case 2. □

[2] *Counting keys.* Each structural field comes from a finite set: $T \in \mathcal{L}_{\text{pred}}$, $S_O \in \mathcal{O}$, $B_T \subseteq$ SymbolicDomain (so at most $2^{|\text{SymbolicDomain}|}$ choices), and the behaviour string has $|\Sigma_{beh}|^{L_{beh}}$ possibilities. Hence the number of *structurally distinct* EMUs is $N_{\max} = |\mathcal{L}_{\text{pred}}|\,|\mathcal{O}|\,2^{|\text{SymbolicDomain}|}\,|\Sigma_{beh}|^{L_{beh}}$. Utility $U$ and reachability $R_T$ are mutable performance estimates and do not affect identity.

### D.3 Main Theorem

**Theorem D.1** (Recursive Validity Closure)**.** *Under the finite component assumptions and PRCOE semantics:*

1. ***Validity:*** *For all $k \in \mathbb{N}$, $M_k \subseteq$ ValidEMUs*
2. ***Termination:*** *The construction halts in finite steps, yielding a fixed point $M_* = M_{k+1} = M_k$ for some $k \leq N_{max}$*
3. ***PRCOE Consistency:*** *All insights maintain the property that observations $O$ provide evidence for verifiability, and proposed EMUs $E$ satisfy EMU validity criteria*

*Proof.* **Validity by induction:**

- *Base:* $M_0 \subseteq$ ValidEMUs by assumption
- *Step:* If $M_k \subseteq$ ValidEMUs, then by Lemma D.3, for any insight $I = (P_c, P_{\text{alias}}, R, C, O, E)$ with $I \in \mathcal{I}(M_k)$, all EMUs in $E$ are valid. Therefore $M_{k+1} = M_k \cup \bigcup_{I \in \mathcal{I}(M_k)} E_I \subseteq$ ValidEMUs

**Termination:** By Lemma D.4, each insight either adds genuinely novel EMUs or refines existing ones. By Lemma D.2, only finitely many structurally distinct EMUs can exist. Therefore, after at most $N_{max}$ steps, no new structural EMUs can be generated, and $M_{k+1} = M_k$.

**PRCOE Consistency:** By construction, each insight $I$ satisfies:

- Observations $O \subseteq M_k \cup \{\text{context}\}$ provide evidence (Lemma D.1)
- Proposed EMUs $E$ satisfy EMU validity (Lemma D.3)
- The recursive process preserves these properties at each level

□

**Discussion: Why Five Clauses Are Necessary.** Each validity clause prevents a specific failure mode:

- **No $T$:** EMU cannot activate → useless memory
- **No $S_O$:** No provenance or perspective → untrusted brittle memory
- **No $B_T$:** Trigger may range over an unbounded set of symbols ⇒ computational intractability
- **No $U$:** Negative utility EMUs → degraded performance
- **No $R_T$:** Unreachable triggers → dead code accumulation

Together, these clauses enable self-organization properties.

**Recursive Meta-Insights.** The theorem extends to recursive insights where insights can observe other insights. If insight $I^{(n)}$ has observations $O^{(n)}$ that include insights $I^{(n-1)}$, the validity guarantees are preserved because:

1. Previous insights $I^{(n-1)}$ are themselves valid symbolic structures
2. The finite symbolic domain ensures bounded recursion depth
3. The utility threshold $\delta$ prevents infinite proliferation of low-utility meta-insights

**Implementation Implications.** The PRCOE structure has several implementation implications:

1. **Evidence Collection:** The system must maintain a clear mapping between insights and their observational evidence in $O$
2. **EMU Generation:** The proposed EMUs in $E$ must be validated before addition to the memory system
3. **Recursive Processing:** Higher-order insights can observe lower-order insights as part of their evidence base
4. **Termination Guarantee:** The utility threshold and finite EMU space ensure that recursive insight generation will terminate

**Implementation Enforceability.** All finiteness assumptions are statically enforceable: a linter caps predicate depth, a schema limits behavior byte-length, and EMUs are stored in a hash-set keyed on their full tuple. Thus any real implementation can guarantee the theorem's preconditions.

**Scope and Extensions.** This theorem applies specifically to EMUs under the five-clause validity schema with the PRCOE interpretation. The framework naturally extends to other memory structure types by defining appropriate finiteness and validity conditions for their respective observation and proposal sets. Insights may generate other memory structures (fragments, meta-EMUs, etc.), but extending this closure result to other structure types requires defining their own finiteness and validity conditions—a direction for future work.

# E Memory Transformation Process Visualization

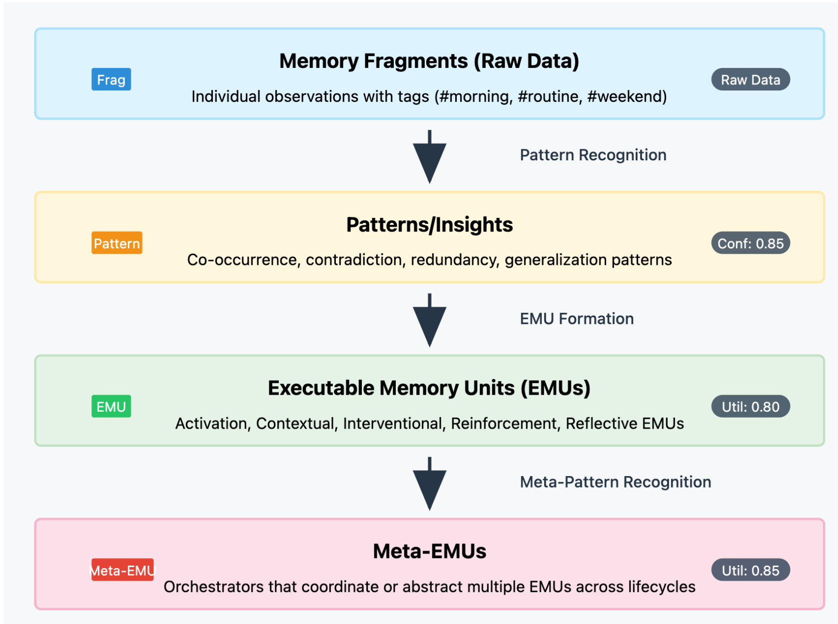


Figure 8: SOMA's Memory Abstraction Hierarchy

## E.1 Complete Transformation Hierarchy

The SOMA system transforms raw memory fragments through four distinct levels of abstraction:

1. **Memory Fragments**: Raw observational data with contextual tags
2. **Patterns/Insights**: Symbolic representations identifying patterns such as co-occurrence, contradiction, and causal relationships
3. **Executable Memory Units (EMUs)**: Structured interventions with triggers, actions, and confidence estimates
4. **Meta-EMUs**: Higher-order abstractions that coordinate or abstract multiple EMUs or memory structures

## E.2 Example Transformation: Memory Fragments to Patterns

**Example E.1** (Memory Fragment Processing)**.** *Raw fragments such as:*

```
<fragment>(user, skipped, breakfast) #morning #weekend</fragment>
<fragment>(user, complained, headache) #afternoon #health #weekend</fragment>
<fragment>(user, expressed, hunger) #afternoon #weekend</fragment>
```

*Transform into the pattern:*

```
Causal Pattern: (user, skipped, breakfast) ∧ #weekend
→ [(user, complained, headache) ∨ (user, expressed, hunger)] ∧ #afternoon
Confidence: 0.85
```

### E.3 EMU Structure and Formation

Patterns are operationalized into EMUs with the following example structure:

Listing 1: EMU Structure Example

```
{
  "id": "emu:self_care.weekend_breakfast_prompt",
  "trigger": {
    "conditions": [
      "(user, skipped, breakfast)",
      "timeTag:morning",
      "dayTag:weekend"
    ]
  },
  "action": {
    "type": "text_prompt",
    "payload": "Consider preparing breakfast-you often report hunger and headaches on weekends when breakfast is skipped."
  },
  "confidence": 0.85,
  "type": "activation",
  "category": "preventive"
}
```

### E.4 Meta-EMU Formation

Meta-EMUs or Meta Memory Fragments emerge from analyzing patterns across existing EMUs:

**Example E.2** (Meta-Pattern Recognition)**.** *Analysis of multiple breakfast-related EMUs reveals:*

- ***Redundancy Pattern**: Weekend breakfast prompt and reflective hunger EMUs overlap*
- ***Generalization Pattern**: Morning intake prompts can be unified across weekdays/weekends*

*This leads to a generalized Meta-EMU:*

```
emu:self_care.morning_intake_nudge
Trigger: timeTag:morning /\ prepared(nourishment_or_caffeine)
Action: "Good morning! Having something now usually prevents
         afternoon hunger or headaches."
```

### E.5 EMU Lifecycle Management

The system maintains EMU efficiency through:

1. **Formation**: Created from identified patterns
2. **Activation**: Triggered by matching conditions
3. **Evaluation**: Assessed for effectiveness
4. **Refinement**: Adjusted based on outcomes
5. **Generalization**: Combined into Meta-EMUs
6. **Pruning**: Archived when redundant or ineffective

Archived EMUs may maintain provenance through Merkle proofs linking to their Meta-EMU successors, ensuring traceability while preventing trigger collisions.

### E.6 System Benefits

The hierarchical transformation process provides:

- **Adaptive Learning**: Continuous improvement based on outcome feedback
- **Personalization**: Context-specific interventions tailored to individual patterns
- **Efficiency**: Meta-EMUs reduce redundancy and computational overhead
- **Preventive Intelligence**: Proactive interventions before problems manifest