Generative Intelligence Network (GIN):
A Cognitive Architecture of Persistent States and Entropic Triggers for Human-Level Cognition

Igor C. Lacerda
Independent Researcher

December 2025

Abstract

Most LLM-based agents remain fundamentally stateless: intelligence collapses at the end of the context window, leading to fragile planning, low task persistence, and no internal criterion for knowing when to stop or rethink. This paper presents the *Generative Intelligence Network (GIN)*, a system architecture for autonomous agents based on state navigation, entropy-driven control, and persistent memory. Rather than proposing a new foundation model, GIN focuses on cognitive orchestration: how agents decide, hesitate, simulate, and revise actions over time. The architecture integrates three orthogonal layers—the Social Layer (A2A), the Execution Layer (MCP), and the Cognitive Layer (MinimalAdS)—creating a system that evolves through Sleep-Time Compute and is evaluated using metrics such as the Cognitive Compression Ratio (CCR).

1 Introduction: The Crisis of Static Intelligence

Modern AI faces a paradox: models with encyclopedic IQ fail at basic sequential tasks due to a lack of persistence. Large Language Models (LLMs) operate as stateless functions ($P(y|x)$), where intelligence evaporates at the end of the context window. They possess no continuous "self" nor an intrinsic ability to know when to stop. To reach AGI, the focus must shift from model training to cognitive system engineering. The proposed architecture integrates three foundational layers:

1. **Social Layer**: Interoperability via the Agent2Agent (A2A) protocol
2. **Physical Layer**: Tool execution via the Model Context Protocol (MCP)
3. **Cognitive Layer**: MinimalAdS, a state-navigation system governed by epistemic uncertainty and hierarchical memory

2 Interoperability Infrastructure: A2A and MCP

2.1 Agent2Agent Protocol (A2A)

The Agent2Agent (A2A) protocol serves as the communication backbone, using *Agent Cards* for dynamic discovery. It allows a Supervisor agent to discover and contract a Specialist agent without pre-programming. Communication is structured in typed JSON schemas, ensuring that intent is preserved across agents.

2.2 Model Context Protocol (MCP)

While A2A manages conversation, the Model Context Protocol (MCP) standardizes action. It decouples reasoning from tools, allowing agents to connect to any data source or actuator through a standardized handshake, maintaining system security and modularity.

3 The Cognitive Core: MinimalAdS

The core innovation is the *Minimal Algorithm of Following (MinimalAdS)*, which reframes reasoning not as text generation, but as navigation through a state graph:

$$S_{t+1} = f(S_t, a_t, m_t)$$

Where:

- S_t = internal cognitive state
- a_t = candidate action
- m_t = memory retrieval

The agent does not act immediately. Instead, it enters a deliberative loop composed of:

- **Perception**: Data ingestion and RAG
- **Planning**: Action hypothesis generation
- **Simulation**: Imagination of outcomes before acting
- **Action**: Execution via MCP

4 Adaptive Teleology: Entropy-Learned Triggers

True autonomy emerges from probabilistic triggers based on *Semantic Entropy*. The agent computes entropy not only over tokens, but over meanings (semantic clusters).

4.1 Uncertainty as a Control Signal

Given a set of simulated futures $\{F_1, F_2, \ldots, F_n\}$, the agent computes semantic divergence:

$$H = -\sum_i p(F_i) \log p(F_i)$$

The decision rule is:

$$\text{Action} = \begin{cases} \text{Abort / Replan / Explore} & \text{if } H > \tau \\ \text{Execute Action} & \text{if } H \leq \tau \end{cases}$$

4.2 Process Reward Models (PRMs)

To train these triggers, we employ Process Reward Models (PRMs). Unlike RLHF, which rewards only the final outcome, PRMs reward each step of logical reasoning. The agent learns to maximize the process reward (R_{proc}), which is inversely proportional to its confusion (entropy).

5 Hierarchical Memory and Sleep-Time Compute

Identity persistence is guaranteed by an architecture inspired by operating systems:

- **Context (RAM)**: Working memory
- **Recall (Disk)**: Vector / episodic storage
- **Archive**: Compressed semantic rules

During idle periods, the agent performs *Sleep-Time Compute*:

- Replay of failed experiences
- Distillation of heuristic rules
- Refinement of entropy thresholds τ

6 Validation Metrics: Cognitive Compression Ratio (CCR)

We propose the *Cognitive Compression Ratio (CCR)* to measure cognitive efficiency:

$$\text{CCR} = 1 - \frac{|T_{plan}|}{|T_{input}|}$$

An increasing CCR indicates that the agent is internalizing episodic experience into semantic wisdom, requiring fewer "thought tokens" to solve complex problems over time.

7 Comparison with Existing Paradigms

Paradigm	**Approach**	**Limitations**
ReAct	Interleaves reasoning and action	No learned abort criterion
AutoGPT / CrewAI	External signals / fixed heuristics	No internal uncertainty control
GIN	Semantic entropy control	Computational complexity

8 Practical Applications

8.1 High-Risk Agents

Finance, healthcare, and legal domains where entropy-based control allows action abortion under high uncertainty.

8.2 Autonomous Software Engineering

Code development and refactoring with pre-execution impact simulation and long-term context retention.

8.3 Corporate Automation with Compliance

Bureaucratic processes and audits with rigorous schema validation and compliance guarantees.

8.4 Evolving Personal Assistants

Assistants that learn during idle periods (Sleep-Time Compute) and continuously refine user understanding.

9 Technical Implementation

9.1 LangGraph Architecture

MinimalAdS is implemented as a state graph:

- Nodes represent cognitive states
- Edges are entropy-conditioned transitions
- Checkpoints persist episodic memory

9.2 Schemas with Pydantic

Strict state validation using Pydantic models ensures structural integrity.

9.3 Memory System with ChromaDB

Persistent vector storage for semantic retrieval and experience consolidation.

10 Conclusion and Future Directions

GIN represents the convergence of robust software engineering (A2A/MCP) with computational cognitive science (MinimalAdS/Entropy). We are not merely building better chatbots—we are building *Cognitive Operating Systems*. The path to AGI does not require "new physics," but disciplined integration of persistent states, uncertainty-based triggers, and interoperable protocols.

Future work includes:

- Refinement of semantic entropy estimators
- Stability analysis of threshold τ across domains
- Exploration of the relationship between CCR and generalization
- Integration with Vision-Language-Action (VLA) models for robotics

References

1. Google Developers. Agent2Agent (A2A) Protocol.
2. Anthropic. Model Context Protocol (MCP).
3. Minimal Algorithm of Following (MinimalAdS) Implementation.
4. Semantic Entropy in LLMs.
5. Process-Supervised Reward Models (PRMs).

6. LangGraph & Stateful Orchestration.
7. MemGPT: Towards LLMs as Operating Systems.
8. Voyager: An Open-Ended Embodied Agent.