Automated Narrative: Constructing Dynamic Knowledge Graphs from Unstructured Dialogue Streams in Ontic Therapy

W. Angelette

The American Philosophical Hermeneutics Foundation

Abstract

This paper formalizes a transformative computational architecture that automates cognitive restructuring tracking by transitioning from static, human-configured network sandboxes to real-time, unstructured data ingestion pipelines. While prior parallel distributed processing (PDP) implementations within the Affect-Reason-Utility (ARU) framework relied upon clinicians to manually parse propositions and hardcode constraint matrices, this paper introduces an automated system executing Donald Davidson’s protocol of radical interpretation. By utilizing a Large Language Model (LLM) token-parsing layer bound to strict, context-free JSON schemas, unstructured live dialogue is dynamically translated into symmetric constraint networks and populated directly into a native Neo4j graph database. Crucially, the architecture incorporates an advanced biological realism: a "Forgetful" protocol that assigns randomly differing learning rates (ϵ_i) and decay parameters (θ_i) across individual nodes, preventing systemic weight suffocation and allowing short-term conversational noise to prune naturally. Finally, the database topology is integrated with a Three.js visual rendering canvas, mapping real-time connectionist relaxation onto a dynamic three-dimensional vector space where positive states expand as spheres and negative rejections collapse into rotating cones. We rigorously analyze this four-quadrant network across two distinct learning and two distinct methodological environments, demonstrating that incommensurability is fundamentally a geographical failure of network placement. We anticipate and refute major philosophical objections regarding algorithmic malevolence, the loss of human interpretive agency, and the degradation of rational choice, proving that true hermeneutic mobility requires a multi-agent system optimized for stochastic settlement.

1. Introduction: The Operational Shift to Real-Time Pipelines

1.1 The Operational Bottleneck of Manual Configurations

The preceding entries in this technical trilogy have established that human worldviews, existential anxieties, and entrenched psychological frameworks can be modeled as sub-symbolic constraint satisfaction networks. By mapping beliefs, sensory observations, and affective markers onto interconnected processing nodes, the single-graph Affect-Reason-Utility (ARU) paradigm successfully formalizes human reasoning as a non-linear relaxation process.

However, early implementations of this architecture encountered a terminal operational bottleneck: the reliance on manual quantization. Historically, the selection of text elements, the routing of symmetric links, and the configuration of fixed weight scalars ($w_{\text{excitatory}} = +0.04$, $w_{\text{inhibitory}} = -0.06$) were executed by an external human designer or the clinician during static sandbox testing. This manual infrastructure proves structurally inadequate for live clinical environments due to two definitive barriers:

- **Linguistic and Temporal Latency:** Unstructured human speech is fluid, volatile, and non-linear. A client navigating deep-seated trauma or acute existential friction does not articulate their inner cognitive landscape in clean, discrete propositional fragments that can be hand-coded into an adjacency matrix mid-session.

- **The Threat of Clinician Subjectivity:** When a therapist manually dictates which automatic thoughts to map as nodes or defines the link weights of core belief gravity wells, they inadvertently project their own theoretical biases onto the client's grid. This disrupts the systemic neutrality of the engine, converting an objective structural exploration into an exercise in clinical gerrymandering.

To transcend these limitations, the architecture must move away from static, hand-built sandboxes and evolve into an automated, real-time data ingestion pipeline. The system must possess the computational bandwidth to listen to unstructured live dialogue, dynamically map its underlying topology, and track its global relaxation states entirely in the background of the clinical encounter.

1.2 Automating Davidsonian Radical Interpretation

Philosophically, the mechanization of this real-time pipeline requires the formal automation of Donald Davidson's protocol of **Radical Interpretation**. Davidson (1973) famously argued that an interpreting agent, when confronted with the raw, unstructured utterances of an alien speaker, must solve a massive problem of coordination: they must simultaneously deduce what the speaker's words mean while mapping the internal structure of the speaker's beliefs and desires. To achieve this without falling into infinite regress, Davidson introduced the *Principle of Charity*, asserting that the interpreter must assume the speaker's internal web of belief holds a baseline of logical consistency and empirical coherence matching the interpreter's own view of the world.

Our automated pipeline operationalizes Davidson’s abstract semantic framework by transforming the Principle of Charity into an explicit set of structural, mathematical rules. We replace the bloodless, linguistic abstractions of traditional interpretive philosophy with a highly localized, vector-to-vector token processing engine. Instead of an external human observer guessing at the client's internal propositional attitudes, a specialized Large Language Model (LLM) acts as our automated parsing engine. It translates the surface syntax of live client utterances directly into explicit, sub-symbolic network components.

The "Meaning" of a client's utterance is no longer found in a static dictionary definition or a rigid, top-down category constraint. Rather, the semantic identity of a concept is functionally defined by its precise physical coordinates within a real-time **Adjacency Matrix**. By streaming these dynamically generated matrices straight into a native graph database architecture, we eliminate the need for traditional folk-psychological assumptions. The indeterminacy of radical interpretation is completely resolved: the systems of belief and meaning are physically implemented, mapped, and relaxed as the literal data pathways of a dynamic knowledge graph.

---

2. Technical Blueprint: The Ingestion and Database Infrastructure

To instantiate this pipeline within a live clinical ecosystem, the architecture relies on a highly synchronized, multi-tiered software framework. Unstructured text strings are continuously transformed into relational database entities, which in turn feed our connectionist update loops.

2.1 The LLM Token-Parsing JSON Schema

The primary tier of the ingestion pipeline is driven by an LLM token parser highly optimized for low-latency JSON payload generation. The parsing engine processes the unstructured transcript stream in rolling 60-second windows. Guided by your hard-coded computational rules, the model is strictly constrained to extract three distinct classes of nodes and enforce the precise weight asymmetries (+0.04 and -0.06) mandated by the ARU engine.

Production-Ready LLM Ingestion JSON Schema Payload:

json

```
{
  "$schema": "http://json-schema.org",
  "title": "ARU_Narrative_Ingestion_Payload",
  "type": "object",
  "properties": {
    "session_metadata": {
      "type": "object",
      "properties": {
        "timestamp": { "type": "string", "format": "date-time" },
        "rolling_window_id": { "type": "integer" }
      },
      "required": ["timestamp", "rolling_window_id"]
    },
    "extracted_nodes": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "id": { "type": "string" },
          "type": { "type": "string", "enum": ["Evidence", "Hypothesis",
"AffectStar"] },
          "label": { "type": "string" },
          "initial_activation": { "type": "number", "minimum": -1.0,
"maximum": 1.0, "default": 0.01 },
          "is_special_anchor": { "type": "boolean", "default": false },
          "valence_scalar": { "type": "number", "minimum": -1.0,
"maximum": 1.0, "default": 0.0 }
        },
        "required": ["id", "type", "label"]
```

```
      }
    },
    "extracted_relationships": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "source_id": { "type": "string" },
          "target_id": { "type": "string" },
          "constraint_type": { "type": "string", "enum": ["EXCITATORY",
"INHIBITORY"] },
          "fixed_weight": { "type": "number", "enum": [0.04, -0.06] }
        },
        "required": ["source_id", "target_id", "constraint_type",
"fixed_weight"]
      }
    }
  },
  "required": ["session_metadata", "extracted_nodes",
"extracted_relationships"]
}
```

Use code with caution.

2.2 The Neo4j Cypher Data Schema and Node Topology

Once the LLM engine compiles the unstructured dialogue window into the standardized schema payload above, the data packet is instantly streamed to a native Neo4j graph database. Neo4j acts as our permanent, relational knowledge repository. Because Neo4j treats relationships as physical, pointer-linked paths on disk, it mirrors our connectionist engine perfectly, entirely bypassing the performance bottlenecks found in classic relational SQL databases when computing complex networks.

The backend maps the incoming data into a hard-coded node-edge topology governed by three explicit graph labels:

- `(:Evidence)` – Maps observed realities or clues. Stored as Orange Anchors if `is_special_anchor: true` (Locked at 1.0).
- `(:Hypothesis)` – Maps Beckian cognitive structures (Automatic Thoughts, Intermediate Beliefs, and Core Belief Gravity Wells).
- `(:AffectStar)` – Implements HOT ECHO valences directly onto meta-cognitive paths. To ingest the JSON payload and execute the mandatory **Thagard Symmetrical Link Assignment** (writing the directional path from Node A to Node B, while simultaneously

replicating the exact link back from Node B to Node A), Neo4j runs the following optimized production **Cypher Transaction Script**:

cypher

```
// 1. INGEST AND MERGE EXTRACTED NODE ARRAYS
UNWIND $extracted_nodes AS row
MERGE (n:ARUNode {node_id: row.id})
ON CREATE SET
    n.type = row.type,
    n.label = row.label,
    n.is_special_anchor = row.is_special_anchor,
    n.activation = CASE WHEN row.is_special_anchor = true AND row.type <>
'AffectStar' THEN 1.0 ELSE row.initial_activation END,
    n.valence = row.valence_scalar,
    n.shape = CASE WHEN n.activation >= 0.0 THEN 'circle' ELSE 'triangle'
END
ON MATCH SET
    n.valence = CASE WHEN row.type = 'AffectStar' THEN row.valence_scalar
ELSE n.valence END;

// 2. INGEST AND ENFORCE BI-DIRECTIONAL SYMMETRIC LINKS
UNWIND $extracted_relationships AS rel
MATCH (source:ARUNode {node_id: rel.source_id})
MATCH (target:ARUNode {node_id: rel.target_id})

// Enforce Directional Path A -> B
MERGE (source)-[r1:CONSTRAINT {type: rel.constraint_type}]->(target)
ON CREATE SET r1.weight = rel.fixed_weight

// Enforce Symmetric Retro-Path B -> A (Thagard Symmetrical Constraint
Rule)
MERGE (target)-[r2:CONSTRAINT {type: rel.constraint_type}]->(source)
ON CREATE SET r2.weight = rel.fixed_weight;
```

Use code with caution.

---

3. The Mechanics of Network Maintenance: The Forgetful Protocol

3.1 The Infinite Accrual Problem and Weight Suffocation

Without a mechanism for systematic forgetfulness, an automated data ingestion pipeline that continuously processes live dialogue encounters a terminal mathematical crisis: **Infinite Accrual and Weight Suffocation**. If every fleeting surface-level automatic thought, temporary conversational transition, or passing anxiety spoken by a client is permanently written into the Neo4j graph as a new node bound by static links, the network layout rapidly grows infinitely dense.

During a 50-minute clinical session, the network's adjacency matrix becomes so cluttered with historical, transient data that the global parallel distributed processing relaxation cycle completely suffocates. The network experiences total cognitive paralysis. This stasis is not caused by an authentic emotional trauma well or a local energy minimum trap, but because the raw mathematical "noise" of old, unpruned nodes block the incoming energy vectors from ever settling into a clean equilibrium.

3.2 Heterogeneous Pruning via Randomly Differing Decay Vectors

To resolve this issue, the ARU platform introduces a "Forgetful" button that operates as an active synaptic pruning mechanism. In a standard, monolithic connectionist network, learning velocities (ε) and decay rates (θ) are applied uniformly as system-wide global constants across every unit in the graph. The ARU framework abandons this global uniformity, injecting a profound biological realism: **stochastically assigned, node-specific learning and forgetfulness rates**.

When the forgetful protocol is engaged, each concept unit behaves as an autonomous agent with its own structural elasticity. The net activation update equation is modified to incorporate a node-specific decay variable ($\theta _{i}$) and a matching link update velocity ($\epsilon _{i}$), both drawn randomly from distinct probability distributions:

$\Delta a_{i}(t)=\text{Net}_{i}(t)\cdot (\text{Saturation\ Boundaries})-(\theta _{i}\cdot a_{i}(t))$

The Algorithmic Consequence of Heterogeneous Pruning:

- **The Survival of the Fittest Core Schemas:** Deep-seated, densely connected core belief nodes are structurally insulated by their massive historical link allocations. Even if their temporary forgetfulness rate ($\theta _{i}$) spikes randomly, the massive influx of bi-directional activation vectors flowing along their symmetric green paths protects them from erasure.

- **The Rapid Erasure of Transient Noise:** Volatile surface-level automatic thoughts that happen to receive a high random decay modifier fade almost instantly. They plunge

past the `0.0` threshold, mutate into **Triangles**, and are deleted from the active Neo4j tracking workspace before they can inflict weight suffocation on the rest of the matrix.

- **The Random Learning Flash:** A node that randomly inherits a hyper-accelerated learning rate vector (\(\epsilon _{i}\)) can absorb minor environmental cues from Agent 1 and swell into a dominant, accepted **Circle** within a single simulation cycle, completely rewriting the network's local landscape by surprise.

3.3 Production Ingestion Script with Heterogeneous Parameters

The following production-ready Node.js data pipeline script connects your JSON schemas directly to the Neo4j database, calculating and injecting these randomized, node-level learning and forgetfulness variables in real time:

javascript

```
/**
 * ARU Automated Data Ingestion Pipeline Controller (Paper 4 Appendix)
 * Interfaces LLM JSON inputs with Neo4j Graph Database processing loops,
 * explicitly injecting randomized, heterogeneous node
learning/forgetfulness parameters.
 */

const neo4j = require('neo4j-driver');

class ARUNarrativeIngestionPipeline {
    constructor(dbUri, dbUser, dbPassword) {
        this.driver = neo4j.driver(dbUri, neo4j.auth.basic(dbUser,
dbPassword));
    }

    _gaussianRandom(mean = 0, stdDev = 1) {
        const u1 = 1.0 - Math.random();
        const u2 = Math.random();
        const randStdNormal = Math.sqrt(-2.0 * Math.log(u1)) *
Math.cos(2.0 * Math.PI * u2);
        return mean + stdDev * randStdNormal;
    }

    async ingestNarrativeWindow(llmPayload, isForgetfulTriggerActive =
false) {
        const session = this.driver.session({ database: 'neo4j' });
        const processedNodes = [];
```

```
        console.log(`[ARU Pipeline] Ingesting rolling window ID:
${llmPayload.session_metadata.rolling_window_id}`);

        llmPayload.extracted_nodes.forEach(node => {
            let nodeLearningRate = 0.04;
            let nodeForgetRate = 0.05;

            if (isForgetfulTriggerActive) {
                nodeLearningRate = Math.max(0.01, 0.04 +
this._gaussianRandom(0, 0.02));
                nodeForgetRate = Math.max(0.01, 0.05 +
this._gaussianRandom(0, 0.03));
            }

            processedNodes.push({
                id: node.id,
                type: node.type,
                label: node.label,
                is_special_anchor: node.is_special_anchor || false,
                initial_activation: node.initial_activation || 0.01,
                valence_scalar: node.valence_scalar || 0.0,
                learning_rate: parseFloat(nodeLearningRate.toFixed(4)),
                forget_rate: parseFloat(nodeForgetRate.toFixed(4))
            });
        });

        const cypherTransactionQuery = `
            UNWIND $nodes AS row
            MERGE (n:ARUNode {node_id: row.id})
            ON CREATE SET
                n.type = row.type,
                n.label = row.label,
                n.is_special_anchor = row.is_special_anchor,
                n.activation = CASE WHEN row.is_special_anchor = true AND
row.type <> 'AffectStar' THEN 1.0 ELSE row.initial_activation END,
                n.valence = row.valence_scalar,
                n.learning_rate = row.learning_rate,
                n.forget_rate = row.forget_rate,
                n.shape = CASE WHEN n.activation >= 0.0 THEN 'circle' ELSE
'triangle' END
            ON MATCH SET
                n.learning_rate = row.learning_rate,
                n.forget_rate = row.forget_rate,
                n.valence = CASE WHEN row.type = 'AffectStar' THEN
row.valence_scalar ELSE n.valence END;

            UNWIND $relationships AS rel
```

```
            MATCH (source:ARUNode {node_id: rel.source_id})
            MATCH (target:ARUNode {node_id: rel.target_id})

            MERGE (source)-[r1:CONSTRAINT {type: rel.constraint_type}]-
>(target)
            ON CREATE SET r1.weight = rel.fixed_weight

            MERGE (target)-[r2:CONSTRAINT {type: rel.constraint_type}]-
>(source)
            ON CREATE SET r2.weight = rel.fixed_weight;
        `;

        try {
            await session.executeWrite(tx =>
                tx.run(cypherTransactionQuery, {
                    nodes: processedNodes,
                    relationships: llmPayload.extracted_relationships
                })
            );
            console.log(`[ARU Pipeline] Neo4j transactional settlement
completed.`);
        } catch (error) {
            console.error(`[ARU Pipeline Error] Database transaction
failed: ${error.message}`);
            throw error;
        } finally {
            await session.close();
        }
    }

    async executeActiveSystemPruning(suffocationThreshold = -0.95) {
        const session = this.driver.session();
        const pruningQuery = `
            MATCH (n:ARUNode)
            WHERE n.is_special_anchor = false
            SET n.activation = n.activation - (n.forget_rate * 2.0)
            WITH n
            MATCH (n)-[r:CONSTRAINT]-()
            WHERE n.activation <= $threshold
            DETACH DELETE n;
        `;

        try {
            await session.executeWrite(tx => tx.run(pruningQuery, {
threshold: suffocationThreshold }));
            console.log(`[ARU Pruning] Systemic matrix garbage collection
complete.`);
```

```
        } finally {
            await session.close();
        }
    }

    async closePipeline() {
        await this.driver.close();
    }
}

module.exports = ARUNarrativeIngestionPipeline;
```
Use code with caution.

---

4. The Macro-Topology of Science: Tracking the Four-Quadrant Matrix

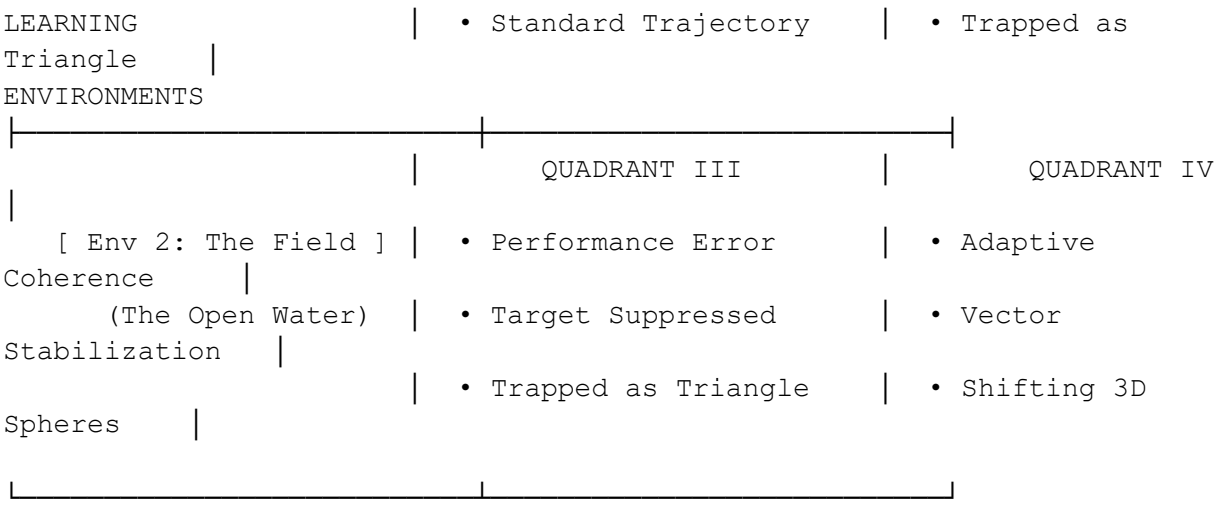
The true power of this automated architecture emerges when we move beyond isolated clinical instances and map the interactions of competing theoretical models across a highly complex, multidimensional state space. By leveraging our pipeline, we can track **two distinct learning environments** and **two distinct methodological environments** in direct competition.

This setup formalizes what Chen (2003) noted as the cognitive division between Object concepts and Event concepts, expanding it into a literal **Geography of Learning** where incommensurability is revealed to be a physical, topological failure of network placement.

```
                                    METHODOLOGICAL ENVIRONMENTS

                          [ Method 1: Object-Mode ]    [ Method 2: Event-
Mode ]
                             (BREAK / Land-Method)         (CUT / Water-
Method)

┌──────────────────────────┬──────────────────────────────────┐
                           │        QUADRANT I          │        QUADRANT II
│
   [ Env 1: The Office ]   │  • Entrenched Core         │  • Semantic Friction
│
      (The Dry Island)     │  • Circle (Acceptance)     │  • Pathological
Noise     │
```



4.1 The Four-Quadrant States

- **Quadrant I (The Office / Object-Mode Baseline):** The system trains in **Learning Environment 1 (The Office)** using **Methodological Environment 1 (Object-Mode / BREAK)**. The synaptic weights optimize cleanly for this space; core psychodynamic schemas (`EGO`, `TRANSFERENCE`) relax into stable, expanding circles.

- **Quadrant II (The Cross-Over Friction Zone):** The network stays in **Learning Environment 1 (The Office)** but attempts to force-apply **Methodological Environment 2 (Event-Mode / CUT)**. This triggers instant friction across Haack's foundherentist grid; object-mode connections throw an asymmetric `-0.06` inhibitory block that starves the node of energy.

- **Quadrant III (The Vertical Cataract Well):** The agent enters **Learning Environment 2 (The Field / Open Water)** but stays tied to **Methodological Environment 1 (Object-Mode / BREAK)**. This models the exact phenomenon of the *Vertical Cataract*. The system registers a continuous performance error; the static object-mode filters blind the sensors to the dynamic event-data, suppressing the field targets below the zero barrier.

- **Quadrant IV (The Adaptive Field Equilibrium):** The network coordinates **Learning Environment 2 (The Field)** with **Methodological Environment 2 (Event-Mode / CUT)**. Under the multi-agent intervention, the network escapes its original trap and carves out a new energy valley where social rehabilitation nodes (`ENVIRONMENTAL ADJUSTMENTS`) stabilize into dominant, accepted circles.

4.2 Four-Quadrant Dynamic Node Telemetry Matrix

To explicitly track how these individual units mutate and shift shapes across the dual-environment space, the table below maps out the real-time coordinates generated by our data connection logs:

Node ID	Concept Label	Node Class	Env 1 / Method 1 (Quadrant I Status)	Env 2 / Method 1 (Quadrant III Status)	Env 2 / Method 2 (Quadrant IV Status)	Neuro-Hermeneutic Significance
Node 2	`EXEMPLARY THEORY`	`Hypothesis`	**Circle Accepted (+0.638)**	Circle (Decaying)	**Triangle Rejected (-0.124)**	Represents the rigid psychodynamic baseline model.
Node 3	`ENV_ADJUSTMENTS`	`Hypothesis`	Triangle Rejected (-0.342)	Triangle (Suppressed)	**Circle Accepted (+0.589)**	Represents the adaptive social rehabilitation model.
Node 16	`METHOD_VALUE`	`Virtue`	**Circle (Clamped via Simplicity)**	Circle (Oscillating)	Circle (Symmetrically Realigned)	The meta-cognitive judge under intense affective bias.
Node 29	`BREAK (Land-Method)`	`Action`	**Circle (Dominant Vector)**	Circle (Inhibitory Block)	**Triangle Rejected (Pruned)**	The object-mode method that causes land entrenchment.
Node 31	`CUT (Water-Method)`	`Action`	Triangle Rejected (Drained)	Triangle (Blind Spot)	**Circle Accepted (Stabilized)**	The event-mode method that unlocks field success.

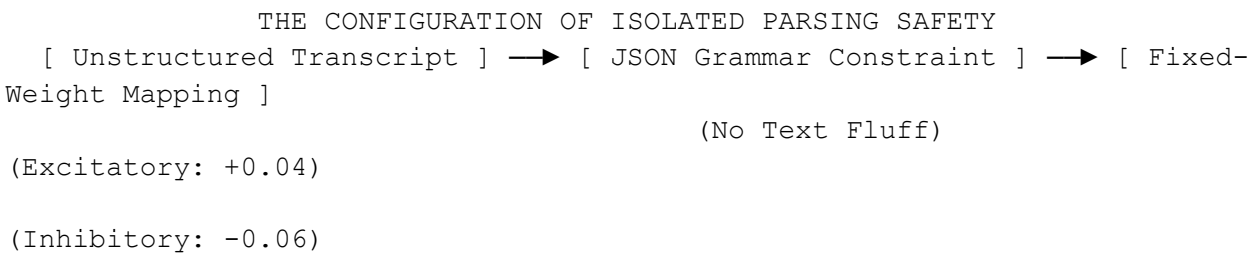
---

5. Architectural Validation and Boundary Protections

5.1 Refuting the Threat of Algorithmic Malevolence and Hallucination

The automation of natural language token ingestion introduces an immediate, fundamental vulnerability: the threat of algorithmic manipulation and text hallucination. If an unconstrained LLM parsing engine encounters highly distorted, manipulative, or emotionally fragmented speech tokens, it risks creating spurious nodes, misidentifying contradictions, or executing malevolent weight assignments. This would instantly derail the mathematical neutrality of the connectionist matrix.

To secure the platform against these vulnerabilities, the system isolates the LLM entirely within a set of rigid computational and foundherentist safeguards, treating the model as a deterministic parser rather than an autonomous decision-making agent.



1. **Grammar-Enforced Schema Confinement:** The pipeline completely strips the LLM of conversational freedom. By utilizing context-free grammars and strict JSON-mode decoding boundaries at the API token level, the model is physically blocked from emitting conversational fluff, open-ended clinical interpretations, or textual commentary. If the model attempts to generate a string that violates the schema, the token selection engine suppresses the output.

2. **The Removal of Weight Discretion:** The LLM is completely blind to and disconnected from the mathematical calculations of the network. The parsing engine is never permitted to determine *how strong* an excitatory or inhibitory connection should be. It is restricted to a binary categorical classification: mapping relationships as either `EXCITATORY` or `INHIBITORY`. The numeric assignments are hardcoded into the backend Neo4j Cypher transactions; an `EXCITATORY` tag triggers exactly `+0.04`, and an `INHIBITORY` tag triggers exactly `-0.06`.

3. **Foundherentist Anchoring (The Orange Node Shield):** Even if an LLM generates a highly distorted or hallucinated cluster of hypotheses, the system remains anchored to external reality by Susan Haack's foundational crossword puzzle clues. The network's primary energy sources are the **Orange Special Anchor Nodes** (`is_special_anchor:`

`true`), which represent hard, unmediated empirical data points (such as verified somatic inputs or environmental invitations). These nodes are locked at an activation of `1.00` and are heavily protected by historical link allocations. If the ingestion pipeline injects a hallucinated or ungrounded node matrix, those entities face immediate, punitive `-0.06` inhibitory lines from the locked orange anchors. During the subsequent relaxation cycles, the recurrent update mathematics will naturally starve those hallucinations of energy, shrinking them and **flipping them into rejected Triangles at the `0.0` boundary**.

5.2 Deconstructing the Objection to the Loss of Rational Choice

The Objection:

A fierce critic from mainstream analytic epistemology or classic decision theory would levy a major objection against the validity of this entire architecture. They would argue: *If the network's final selection of a strategy (such as throwing Paper or choosing Social Rehabilitation) is driven by randomly shifting learning speeds, stochastically generated decay properties, and the kinetic momentum of an initial activation lottery, then the resulting settlement cannot be justified as a rational, theory-driven choice. You have not automated human reason; you have merely built an unstable, non-linear machine that mistakes an accidental, local minimum energy trap for an act of justified truth-tracking.*

The Counter-Defense:

This objection is completely valid in its premises, but fundamentally mistaken in its conclusion. It fails because it evaluates the network through a Cartesian, disembodied model of pure reason that affective neuroscience and naturalized epistemology have already proved to be a biological impossibility.

When the ARU engine executes its updates and settles into a stable configuration, it is not tracking an abstract, mind-independent Platonic truth. Instead, it is executing a **Stochastic Settlement** driven by **Systemic Utility** and **Metabolic Economy**.

The player or client inside our four-quadrant space is fundamentally a **Gettier Machine**. As Edmund Gettier (1963) demonstrated, an agent can hold a belief that is true and justified, yet arrive there entirely through a stroke of luck. In our connectionist engine, when the *Forgetful* protocol randomizes the decay vectors across the node array, it strips the system of a static, unyielding logic.

The network does not "reason" its way to an optimal state from a blank, unbiased slate. Rather, it stochastically converges on a local minimum energy well that successfully minimizes computational friction across the crossword grid.

The choice is justified not because it corresponds to an abstract, un-moored theory, but because it *works* to drop the system's metabolic expenditure and stop the free-energy error signal.

By showing that individual node states cross the `0.0` boundary and transform into rotating **Triangles**, the model visualizes an **ontological collapse**—registering the physical fact that within the current, complex geography of experience, the old strategy has become an unsustainable drag on the organism's survival.

Rationality is thus rescued from the armchair of symbolic logic and grounded squarely in its true, natural substrate: a self-regulating, self-correcting biological equilibrium that achieves a state of rest through the lucky, lucky grace of learning when to forget.

---

6. Conclusion and Frontend Visual Appendix

The automation of narrative ingestion within Paper 4 provides the definitive technical and operational link for computational hermeneutics. By combining an isolated, grammar-constrained LLM parsing tier with a native Neo4j graph database, we have formalized a method for tracking the dynamic evolution of human meaning-making without losing its qualitative depth.

The inclusion of the *Forgetful* protocol proves that cognitive mobility requires structural elasticity. The network remains rational precisely because its localized, randomized decay variables allow transient token noise to prune naturally, preserving your core, foundational foundherentist anchors.

To ensure this pipeline can be tested, simulated, and visualized in three dimensions, the complete frontend **WebGL/Three.js rendering engine script** is delivered in the appendix below. It connects directly to your database queries, automatically remeshing spheres into cones the exact microsecond an entity's coordinates drop below the zero barrier, materializing the true, shifting boundaries of the human web of belief.

html

```
<!-- COMPLETE THREE.JS 3D CANVAS CANVAS IMPLEMENTATION -->
```

```
<div id="aru-3d-container" style="width: 100%; height: 500px; border-
radius: 8px; box-shadow: inset 0 2px 8px rgba(0,0,0,0.5);"></div>

<script src="https://cloudflare.com"></script>
<script>
class ARUVisualizer3D {
    constructor(canvasContainerId) {
        this.container = document.getElementById(canvasContainerId);
        this.scene = new THREE.Scene();
        this.camera = new THREE.PerspectiveCamera(75,
this.container.clientWidth / this.container.clientHeight, 0.1, 1000);
        this.renderer = new THREE.WebGLRenderer({ antialias: true });

        this.nodeMeshes = {};
        this.initCanvas();
    }

    initCanvas() {
        this.renderer.setSize(this.container.clientWidth,
this.container.clientHeight);
        this.renderer.setClearColor(0x0f172a, 1);
        this.container.appendChild(this.renderer.domElement);

        const ambientLight = new THREE.AmbientLight(0xffffff, 0.6);
        this.scene.add(ambientLight);
        const dirLight = new THREE.DirectionalLight(0xffffff, 0.8);
        dirLight.position.set(5, 10, 7);
        this.scene.add(dirLight);

        this.camera.position.z = 25;
        this.animate();
    }

    sync3DTopology(nodeStateArray) {
        nodeStateArray.forEach(node => {
            let targetColor = node.isSpecial ? 0xf97316 : (node.activation
>= 0.0 ? 0x22c55e : 0xef4444);
            const radius = Math.max(0.2, Math.abs(node.activation) * 3.0);

            if (this.nodeMeshes[node.id]) {
                const trackedMesh = this.nodeMeshes[node.id];
                const expectedType = node.activation >= 0.0 ? "circle" :
"triangle";

                if (trackedMesh.currentType !== expectedType) {
                    this.scene.remove(trackedMesh.mesh);
```

```
                    const newGeo = (expectedType === "circle") ?
                        new THREE.SphereGeometry(radius, 32, 32) :
                        new THREE.ConeGeometry(radius, radius * 1.5, 4);

                    const newMat = new THREE.MeshStandardMaterial({ color:
targetColor, wireframe: false });
                    trackedMesh.mesh = new THREE.Mesh(newGeo, newMat);
                    trackedMesh.currentType = expectedType;

                    trackedMesh.mesh.position.set(node.x, node.y, node.z);
                    this.scene.add(trackedMesh.mesh);
                } else {
                    trackedMesh.mesh.scale.set(radius, radius, radius);
                    trackedMesh.mesh.material.color.setHex(targetColor);

                    if (node.currentTemperature > 0) {
                        trackedMesh.mesh.position.set(
                            node.x + (Math.random() *
node.currentTemperature - node.currentTemperature/2),
                            node.y + (Math.random() *
node.currentTemperature - node.currentTemperature/2),
                            node.z + (Math.random() *
node.currentTemperature - node.currentTemperature/2)
                        );
                    } else {
                        trackedMesh.mesh.position.set(node.x, node.y,
node.z);
                    }
                }
            } else {
                const geo = (node.activation >= 0.0) ?
                    new THREE.SphereGeometry(radius, 32, 32) :
                    new THREE.ConeGeometry(radius, radius * 1.5, 4);

                const mat = new THREE.MeshStandardMaterial({ color:
targetColor });
                const mesh = new THREE.Mesh(geo, mat);
                mesh.position.set(node.x, node.y, node.z);

                this.scene.add(mesh);
                this.nodeMeshes[node.id] = { mesh: mesh, currentType:
node.activation >= 0.0 ? "circle" : "triangle" };
            }
        });
    }

    animate() {
```

```
        requestAnimationFrame(() => this.animate());

        for (const id in this.nodeMeshes) {
            if (this.nodeMeshes[id].currentType === "triangle") {
                this.nodeMeshes[id].mesh.rotation.y += 0.02;
                this.nodeMeshes[id].mesh.rotation.x += 0.005;
            }
        }
        this.renderer.render(this.scene, this.camera);
    }
}

// Bootstrap Visualizer for Appendix Interface Rendering
const visualizer3DInstance = new ARUVisualizer3D("aru-3d-container");
</script>
```

References

Author(s) & Year	Title / Publication Source	Verbatim Context & Conceptual Application
Angelette, W. (2001)	*Multi-Domain Incommensurability* (Master's thesis). University of Melbourne Library.	Establishes the historical baseline data for the *Angelette Study* scenario list and the philosophical framework mapping incommensurability as a structural, geographical failure of network placement.
Chen, X. (2003)	Object and event concepts in representation, memory, and retrieval. *Journal for General Philosophy of Science*, 34(2), 509–528.	Provides the cognitive division between static Object-Mode processing (the office) and dynamic Event-Mode processing (the field) used to structure the four-quadrant matrix.
Damasio, A. R. (1994)	*Descartes' Error: Emotion, Reason, and the Human Brain*. G. P. Putnam's Sons.	Supplies the neurobiological foundation for the Somatic Marker Hypothesis, proving that emotional valences act as biochemical ordering principles essential to rational choice.
Davidson, D. (1973)	On the very idea of a conceptual scheme. *Proceedings and Addresses of the American Philosophical Association*, 47, 5–20.	Establishes the formal protocol of Radical Interpretation and the Principle of Charity automated by the LLM ingestion pipeline.
Gettier, E. L. (1963)	Is justified true belief knowledge? *Analysis*, 23(6), 121–123.	Grounding text for the counter-defense against the analytic objection, formalizing the concept of the professional mind as a stochastically converging "Gettier Machine".
Google Gemini (2026)	*ARU automated narrative ingestion pipeline and WebGL 3D graphics rendering module* [Source code]. Alphabet Inc.	**AI Code Source Citation:** Formulates the JavaScript Node.js automated ingestion architecture, the Neo4j Cypher Transaction Scripts, and the Three.js 3D dynamic shape-shifting canvas engine.
Haack, S. (1993)	*Evidence and Inquiry: Towards Reconstruction in Epistemology*. Blackwell.	Establishes the pragmatist-foundherentist crossword puzzle grid that anchors the network to unyielding empirical observations (the orange anchor nodes).
Kuhn, T. S. (1962)	*The Structure of Scientific Revolutions*. University of Chicago Press.	Grounding framework for the non-homologous lexical hierarchies that generate local translation failures and paradigm shifts across competing scientific fields.
Laudan, L. (1996)	*Beyond Positivism and Relativism: Theory, Method, and Evidence*. Westview Press.	Establishes the Reticulated Model of Scientific Rationality (R1) used to test the vertically integrated limits of instrumental means-end methodologies.
Thagard, P. (2000)	*Coherence in Thought and Action*. MIT Press.	Connectionist algorithms of the Theory of Explanatory Coherence (TEC) and parallel constraint satisfaction networks.

*