

# Autonomous DOM Neural Unit: Core Mathematical Workflow

December 21, 2025

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>System Architecture Overview</b>	<b>2</b>
2.1	Core Components . . . . .	2
2.2	Dimensional Space Definition . . . . .	2
<b>3</b>	<b>Pattern Definitions and Strategic Design</b>	<b>2</b>
3.1	Pattern Taxonomy . . . . .	2
<b>4</b>	<b>Core Transformations</b>	<b>2</b>
4.1	$T$ Transformation: Relational Encoding . . . . .	2
4.1.1	Dimension Composition . . . . .	3
4.1.2	Question Logic . . . . .	3
4.1.3	Example: $T$ Transformation . . . . .	3
4.2	AttributeExpression System . . . . .	5
4.2.1	Numeric Encoding . . . . .	5
<b>5</b>	<b>Neuron Processing Workflow</b>	<b>5</b>
5.1	Complete Cycle Flowchart . . . . .	5
5.2	Phase 0: Initialization (One-Time Setup) . . . . .	6
5.3	Phase 1: Self Observation with $\alpha$ Update . . . . .	6
5.3.1	Mathematical Operations . . . . .	6
5.3.2	UNKNOWN $\gamma$ Update (Special Case) . . . . .	6
5.4	Phase 2: Competitive Assignment . . . . .	7
5.4.1	Hierarchical Selection Operator $H(B)$ . . . . .	7
5.4.2	Permutation Transform $Y$ . . . . .	7
5.5	Phase 3: Neighbor Observation with Void Handling . . . . .	7
5.5.1	Void Detection and Rerouting . . . . .	8
5.6	Phase 4: Matrix Updates with $\beta$ Update . . . . .	8
5.6.1	Mathematical Operations . . . . .	8
5.7	Phase 5: Confidence Decision and Tensor Fallback . . . . .	8

5.7.1	Confidence Decision Logic . . . . .	8
5.7.2	Tensor Fallback with $\zeta$ Update (Phase 5b) . . . . .	9
5.8	Eigen Update Sequences by Pattern Type . . . . .	10
<b>6</b>	<b>Matrix and Tensor Dimensions</b>	<b>10</b>
6.1	Core Data Structures . . . . .	10
6.2	Tensor Structures . . . . .	10
<b>7</b>	<b>AxonNetwork Architecture</b>	<b>10</b>
7.1	Queue System Structure . . . . .	10
7.2	Axon Type Definitions . . . . .	10
<b>8</b>	<b>EnhancedGrandClass Attribute Space</b>	<b>10</b>
8.1	Base 9 Dimensions . . . . .	10
8.2	Dual Combinations (AND Relationships) . . . . .	10
8.3	Triple Combinations (Robust Signatures) . . . . .	10
<b>9</b>	<b>Implementation Details</b>	<b>10</b>
9.1	Normalization Operations . . . . .	10
9.1.1	Vector Normalization . . . . .	10
9.1.2	Matrix Row Normalization . . . . .	10
9.2	Numerical Stability Considerations . . . . .	10
9.3	Error Handling Strategy . . . . .	11
<b>10</b>	<b>Example: Complete Processing Cycle</b>	<b>11</b>
10.1	Input Parameters . . . . .	11
10.2	Step-by-Step Execution . . . . .	11
10.3	Expected Output . . . . .	12
<b>11</b>	<b>Computational Properties</b>	<b>12</b>
11.1	Time Complexity Analysis . . . . .	12
11.2	Memory Usage . . . . .	12
<b>12</b>	<b>Conclusion</b>	<b>12</b>

# 1 Introduction

This document presents the **exact mathematical workflow** of the DOM neural unit. The system processes DOM elements through a 6-phase cycle using matrix operations, eigen decompositions, and relational transformations.

## 2 Core Mathematical Framework

### 2.1 Dimensional Spaces

- **25D Attribute Space:**  $\mathbb{A} = \mathbb{R}^{25}$
- **87D Relational Space:**  $\mathbb{R}^{87}$  (binary encoding)
- **5 Patterns:** DATA\\_INPUT, ACTION\\_ELEMENT, CONTEXT\\_ELEMENT, STRUCTURAL, UNKNOWN
- **6 Positions:** self, parent, up, down, left, right

### 2.2 Key Matrices and Tensors

Symbol	Dimensions	Description
$X$	$5 \times 25$	Self expectations for all patterns
$P_i$	$5 \times 25$	Neighbor expectations for pattern $i$
$B$	$5 \times 5$	Position bias matrix
$b$	$5 \times 1$	Pattern probability vector
$T(V)$	$n \times 87$	$T$ -transform of $n$ vectors
$E$	$5 \times 6 \times 25$	Complete expectation tensor
$T_{\text{exp}}$	$5 \times 6 \times 87$	$T$ -transformed expectations

Table 1: Core mathematical objects

## 3 The $T$ -Transformation: Relational Encoding

### 3.1 Definition

$$T : \mathbb{R}^{n \times 25} \rightarrow \mathbb{R}^{n \times 87}$$

### 3.2 Composition

$$87 = \underbrace{15}_{\text{combinations}} + \underbrace{9 \times 8}_{\text{base} \times \text{questions}}$$

### 3.3 Example: 3 Vectors Transformation

Given  $V \in \mathbb{R}^{3 \times 25}$ :

$$V = \begin{bmatrix} v_{1,0} & v_{1,1} & \cdots & v_{1,24} \\ v_{2,0} & v_{2,1} & \cdots & v_{2,24} \\ v_{3,0} & v_{3,1} & \cdots & v_{3,24} \end{bmatrix}$$

For dimension  $d = 0$ :

1. Count vectors with value  $v_{1,0}$ : e.g., 2 vectors
2. Set sharing flags:  $Q3_d = 1$  for those 2 vectors
3. Set uniqueness flag:  $Q4_d = 1$  for remaining vector
4. If  $v_{1,0} = 0.5$ , count optionality: e.g., 1 vector
5. Set optionality flag:  $Q8_d = 1$  for that vector

## 4 Complete 6-Phase Mathematical Workflow

### 4.1 Workflow Diagram

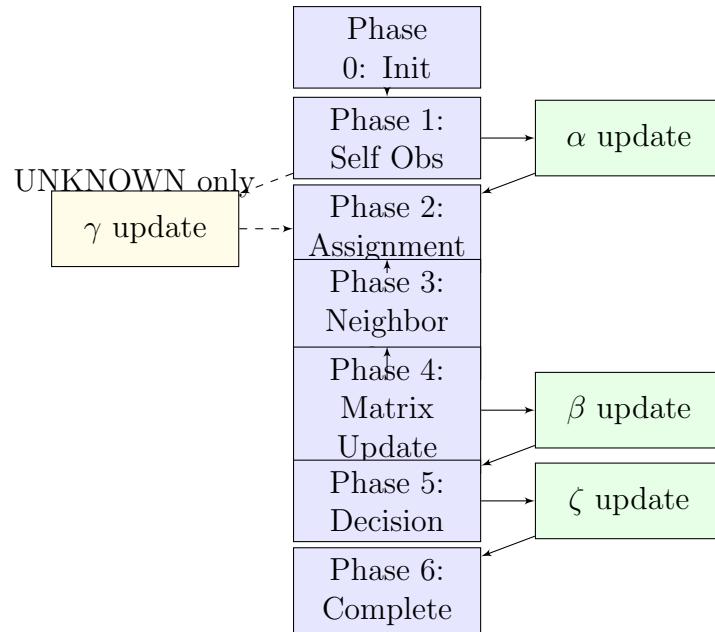


Figure 1: 6-phase workflow with eigen updates

Operation	Mathematical Result
Create ROSE instance	$\text{ROSE}(p_0, c)$
Extract $E$ tensor	$E \in \mathbb{R}^{5 \times 6 \times 25}$
Pre-compute $T_{\text{exp}}$	$T_{\text{exp}} \in \mathbb{R}^{5 \times 6 \times 87}$
Initialize $b$	$b_{\text{initial}} = [0.2, 0.2, 0.2, 0.2, 0.2]^T$
Initialize $B$	$B = \begin{cases} \text{pattern-specific} & \text{standard} \\ \frac{1}{5}J_{5 \times 5} & \text{UNKNOWN} \end{cases}$

Table 2: Phase 0 initialization

## 4.2 Phase 0: Initialization

## 4.3 Phase 1: Self Observation with $\alpha$ Update

### 4.3.1 Mathematical Operations

1. **Self observation:**  $V_{\text{self}} \in \mathbb{R}^{5 \times 25}$

$$V_{\text{self}}[i, :] = \text{filter}(X[i, :], \text{DOM observation})$$

2. **87D transformation:**

$$\begin{aligned} T_{\text{self}} &= T(V_{\text{self}}) \in \mathbb{R}^{5 \times 87} \\ S &= T(X) \in \mathbb{R}^{5 \times 87} \end{aligned}$$

3. **Covariance:**

$$S^* = S \cdot T_{\text{self}}^T \in \mathbb{R}^{5 \times 5}$$

4. **Eigen decomposition:**

$$(\alpha, v_\alpha) = \text{eig}(S^*)$$

5. **Bias update:**

$$b_{\text{initial}} \leftarrow \text{normalize}(\alpha \cdot (v_\alpha v_\alpha^T) \cdot b_{\text{initial}})$$

### 4.3.2 Example: $\alpha$ Update

Suppose after observation:

$$S^* = \begin{bmatrix} 0.8 & 0.1 & 0.05 & 0.03 & 0.02 \\ 0.1 & 0.7 & 0.1 & 0.05 & 0.05 \\ 0.05 & 0.1 & 0.6 & 0.15 & 0.1 \\ 0.03 & 0.05 & 0.15 & 0.5 & 0.27 \\ 0.02 & 0.05 & 0.1 & 0.27 & 0.56 \end{bmatrix}$$

Eigen decomposition yields:

$$\alpha = 0.85, \quad v_\alpha = [0.6, 0.4, 0.2, -0.1, -0.1]^T$$

Update:

$$b_{\text{initial}} = \text{normalize}(0.85 \times (v_\alpha v_\alpha^T) \times [0.2, 0.2, 0.2, 0.2, 0.2]^T)$$

## 4.4 UNKNOWN-specific $\gamma$ Update

---

**Algorithm 1**  $\gamma$  Update for UNKNOWN Pattern

---

```

1: procedure GAMMAUPDATE
2:   Build  $T_\gamma$  tensor ( $5 \times 5 \times 87$ )            $\triangleright$  With void reroutes
3:   Flatten:  $G_\gamma \in \mathbb{R}^{5 \times 435}$ 
4:   Covariance:  $C_\gamma = G_\gamma \cdot G_\gamma^T \in \mathbb{R}^{5 \times 5}$ 
5:   Eigen:  $(\gamma, v_\gamma) = \text{eig}(C_\gamma)$ 
6:   Update:  $B \leftarrow \text{normalize}(\gamma \cdot (v_\gamma v_\gamma^T) \cdot B)$ 
7: end procedure
```

---

## 4.5 Phase 2: Competitive Assignment $H(B)$

### 4.5.1 Hierarchical Selection

For each column  $j \in \{0, \dots, 4\}$ :

1. Find  $i = \arg \max_i B[i, j]$
  2. If ties:  $i = \arg \max_{i \in \text{candidates}} (X[i] \cdot V_{\text{self}})$
  3. Remove selected  $i$  from available rows
- Result: indices  $I = [i_0, i_1, i_2, i_3, i_4]$

### 4.5.2 Permutation Transform $Y$

$$P_{i,k} = \text{PermutationMatrix}(I) \cdot P_i$$

where  $\text{PermutationMatrix}(I)[j, i_j] = 1$ .

### 4.5.3 Example: Competitive Assignment

Given  $B$  matrix:

$$B = \begin{bmatrix} 0.3 & 0.1 & 0.2 & 0.2 & 0.2 \\ 0.2 & 0.4 & 0.1 & 0.2 & 0.1 \\ 0.1 & 0.2 & 0.5 & 0.1 & 0.1 \\ 0.2 & 0.1 & 0.1 & 0.4 & 0.2 \\ 0.2 & 0.2 & 0.1 & 0.1 & 0.4 \end{bmatrix}$$

$H(B)$  selection:

- Column 0: row 0 (0.3)
- Column 1: row 1 (0.4)
- Column 2: row 2 (0.5)
- Column 3: row 3 (0.4)
- Column 4: row 4 (0.4)

Result:  $I = [0, 1, 2, 3, 4]$

## 4.6 Phase 3: Neighbor Observation with Void Handling

### 4.6.1 Observation with Membrane Rerouting

For each position  $p \in \{\text{parent, up, down, left, right}\}$ :

- 1: **if** position has active reroute **then**
- 2:     observe(reroute\_coordinate)
- 3: **else if** waiting in membrane\_waiting **then**
- 4:     check membrane system
- 5: **else**
- 6:     observe(original\_coordinate)
- 7: **end if**

### 4.6.2 Example: Void Handling Scenario

Parameter	Value
Neuron coordinate	(0,1,2)
Void at position	"up"
Void coordinate	(0,1,1)
Candidates found	(0,1,0), (0,0,1), (0,2,1), (0,1,3)
$T$ -similarities	0.85, 0.72, 0.45, 0.63
Selected reroute	(0,1,0)

## 4.7 Phase 4: Matrix Updates with $\beta$ Update

### 4.7.1 Matrix Operations Sequence

Step	Mathematical Operation
1	$P_{i,k}^{87D} = T(P_{i,k}) \in \mathbb{R}^{5 \times 87}$
2	$W_p^{87D} = T(O) \in \mathbb{R}^{5 \times 87}$
3	$D = P_{i,k}^{87D} \cdot (W_p^{87D})^T \in \mathbb{R}^{5 \times 5}$
4	$\hat{B} = D \cdot B$
5	$B^* = \text{row\_normalize}(\hat{B})$
6	$(\beta, v_\beta) = \text{eig}(B^*)$
7	$b_{\text{final}} = \text{normalize}(\beta \cdot (v_\beta v_\beta^T) \cdot b_{\text{initial}})$

### 4.7.2 Example: Matrix Update Computation

Given observations:

$$O = \begin{bmatrix} 0.8 & 0.2 & \cdots & 0.1 \\ 0.1 & 0.9 & \cdots & 0.2 \\ 0.3 & 0.1 & \cdots & 0.8 \\ 0.2 & 0.7 & \cdots & 0.1 \\ 0.6 & 0.3 & \cdots & 0.4 \end{bmatrix}$$

After  $T$ -transform:

$$D = \begin{bmatrix} 0.7 & 0.1 & 0.05 & 0.1 & 0.05 \\ 0.05 & 0.8 & 0.05 & 0.05 & 0.05 \\ 0.1 & 0.1 & 0.6 & 0.1 & 0.1 \\ 0.05 & 0.05 & 0.2 & 0.6 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 & 0.6 \end{bmatrix}$$

$\beta$  update:  $\beta = 0.78$ , update  $b_{\text{final}}$

## 4.8 Phase 5: Confidence Decision

Let  $p_{\text{current}} = \arg \max(b_{\text{final}})$ .

$$\text{decision} = \begin{cases} \text{RECYCLING} & \text{if } p_{\text{current}} = \text{current pattern} \\ \text{TENSOR\_FALLBACK} & \text{otherwise} \end{cases}$$

### 4.8.1 Tensor Fallback with $\zeta$ Update

1. Build tensors:  $E = T_{\text{exp}}$ ,  $O_{\text{obs}} = T_{\zeta}(\text{reroutes})$
2. Flatten:  $E_f \in \mathbb{R}^{5 \times 522}$ ,  $O_f \in \mathbb{R}^{5 \times 522}$
3. Covariance:  $G = E_f \cdot O_f^T \in \mathbb{R}^{5 \times 5}$
4. Eigen:  $(\zeta, v_{\zeta}) = \text{eig}(G)$
5. Update:  $b_{\text{grand}} = \text{normalize}(v_{\zeta} \odot b_{\text{final}})$

## 5 Eigen Update Sequences and Interpretations

## 6 Example: Complete Cycle

### 6.1 Initial Conditions

- Coordinate: (0,1,2)
- Initial pattern: UNKNOWN
- Void at: "up" position (0,1,1)
- Reroute found: (0,1,0)

Pattern	Sequence	Interpretation
Standard patterns	$\alpha \rightarrow \beta \rightarrow \zeta$	Self $\rightarrow$ Position $\rightarrow$ Global
UNKNOWN pattern	$\alpha \rightarrow \gamma \rightarrow \beta \rightarrow \zeta$	Self $\rightarrow$ Neighbors $\rightarrow$ Position $\rightarrow$ Global
Eigenvalue	Range	Semantic Meaning
$\alpha$	0.0-1.0	Self-identity certainty
$\gamma$	0.0-1.0	Neighbor relation consistency
$\beta$	0.0-1.0	Position assignment quality
$\zeta$	0.0-1.0	Global pattern consistency

Table 3: Eigen update sequences and interpretations

## 6.2 Expected Eigenvalues

$$\alpha = 0.92, \quad \gamma = 0.78, \quad \beta = 0.85, \quad \zeta = 0.88$$

## 6.3 Final Decision

- Final pattern: DATA\\_INPUT
- Confidence: 0.85
- Active reroutes: 1 ("up"  $\rightarrow$  (0,1,0))

# 7 Matrix Dimension Summary

Operation	Input	Output
$T$ transform	$n \times 25$	$n \times 87$
Self covariance	$5 \times 87, 5 \times 87$	$5 \times 5$
$H(B)$ selection	$5 \times 5$	5 indices
Permutation	$5 \times 25, 5$ indices	$5 \times 25$
D matrix	$5 \times 87, 5 \times 87$	$5 \times 5$
$\beta$ update	$5 \times 5$	updated $b$ ( $5 \times 1$ )
Tensor fallback	$5 \times 522, 5 \times 522$	$5 \times 5$

Table 4: Matrix dimension transformations

# 8 Normalization Operations

## 8.1 Vector Normalization

For  $v \in \mathbb{R}^n$ :

$$\text{normalize}(v)_i = \frac{v_i}{\sum_{j=1}^n v_j} \quad \text{with } \epsilon = 10^{-10}$$

## 8.2 Matrix Row Normalization

For  $M \in \mathbb{R}^{m \times n}$ :

$$\text{row\_normalize}(M)_{ij} = \frac{M_{ij}}{\sum_{k=1}^n M_{ik}}$$

# 9 System Properties

## 9.1 Time Complexity

- $T$  transform:  $O(n \cdot 25 \cdot 9)$  where  $n \leq 5$
- Eigen decomposition:  $O(5^3)$  (constant)
- Complete cycle:  $O(1)$  bounded operations

## 9.2 Expected Behavior

- **High confidence:**  $> 0.7 \rightarrow$  enter recycling mode
- **Medium confidence:**  $0.4 - 0.7 \rightarrow$  continue processing
- **Low confidence:**  $< 0.4 \rightarrow$  likely UNKNOWN or pattern switch
- **Void handling:** Automatic rerouting preserves relational patterns
- **UNKNOWN specialization:**  $\gamma$  update refines neighbor expectations