

Workspace Structure as Information: Clustering Coefficients, Betweenness Centrality, and Subgraph Density as a Unified Probabilistic Framework for Copycat’s Analogical Workspace

Alex Linhares

February 27, 2026

Abstract

The Copycat analogy engine governs its working memory—the Workspace—through three families of hardcoded numerical mechanisms: a power-law support factor regulating bond external strength, fixed ratio weights (0.2/0.8, 0.8/0.2) determining object salience, and a discrete step function (5, 20, 60, 90) assigning importance to groups of different sizes. In a companion paper we proposed replacing these constants, among others, with graph-theoretical constructs across the entire architecture. This paper focuses exclusively on the Workspace and develops three specific proposals in depth. First, we show that the clustering coefficient of a workspace object’s local neighborhood defines a principled *support vector* encoding structural embeddedness, replacing the ad hoc support factor. Second, we demonstrate that normalized betweenness centrality provides a topology-driven *salience weight* for each object, eliminating the fixed intra/inter-string ratio dichotomy. Third, we show that the induced subgraph density of a group provides a continuous, principled *length factor*, replacing the four-value step function. The paper’s central theoretical contribution goes beyond these three substitutions: we observe that all three metrics are ratios of actual connections to the maximum possible connections in their respective domains, placing them naturally in the interval $[0, 1]$. This property licenses their interpretation as probabilities. Consequently, each metric carries a well-defined information content measurable in bits via the Shannon self-information $I = -\log_2 p$. We speculate that this common unit—bits—provides the first principled *arithmetic* basis for combining workspace structure metrics, enabling the total structural information of the workspace to be computed, compared, and related to Copycat’s temperature parameter in a formally grounded way. The implications for cognitive architecture design, thermodynamic analogies, and the measurement of workspace organization are discussed.

1 Introduction

Analogy-making is widely regarded as one of the core operations of human cognition [7, 8]. Among the computational models of this capacity, the Copycat system of Mitchell and Hofstadter stands out for the richness of its architecture and the plausibility of its behavior [11]. Copycat solves letter-string analogy problems—such as “if abc changes to abd, what does ppqrr change to?”—by simultaneously maintaining a semantic network of concepts (the Slipnet), a dynamic working memory (the Workspace), and a probabilistic queue of micro-operations (the Coderack), all regulated by a global temperature that decreases as coherent structure accumulates.

The Workspace is the arena in which analogy-making happens. It begins as a collection of bare letters and, through the action of thousands of stochastic codelets, gradually accretes bonds linking adjacent letters, groups subsuming multiple bonded letters, and correspondences mapping objects in the initial string to objects in the target string. The final Workspace state encodes the

answer: a coherent structural description of how the initial-to-modified transformation applies to the target string.

Despite this elegant high-level architecture, the Workspace’s internal computations rest on a foundation of hardcoded numerical constants and empirically tuned formulas. A companion paper [10] cataloged these constants across the entire Copycat codebase and proposed replacing them with graph-theoretical constructs—betweenness centrality, clustering coefficients, resistance distance, percolation thresholds—drawn from established network science. That paper’s scope was broad by design, touching the Slipnet, the Workspace, and the Coderack.

The present paper narrows its scope to the Workspace and develops three of those proposals in substantially greater depth. Our goals here are:

1. To show that the *clustering coefficient* of a workspace object’s local neighborhood constitutes a principled *support vector*, replacing the power-law support factor currently used in bond external strength.
2. To demonstrate that *normalized betweenness centrality* provides a topology-driven *salience weight* for each object, eliminating the hard-wired intra-string versus inter-string ratio dichotomy.
3. To establish that the *induced subgraph density* of a group provides a continuous, smooth *length factor*, replacing the four-valued step function that currently assigns importance to groups.
4. To advance a speculative but mathematically grounded unification: because all three metrics are ratios of actual to maximum possible connections, they lie naturally in $[0, 1]$ and can be interpreted as probabilities. This licenses computing their Shannon self-information in bits, placing all three on the same measurement scale and enabling principled arithmetic over workspace structure.

The fourth goal is this paper’s central theoretical contribution. The arithmetic of bits is commutative, associative, and additive in a way that the original heterogeneous constants (power laws, fixed ratios, discrete steps) are not. We argue that expressing workspace structure in bits provides the first formally grounded basis for a *workspace entropy*—a scalar quantity capturing total workspace organization—and we speculate on its relationship to Copycat’s temperature.

1.1 Organization of This Paper

Section 2 reviews the Workspace as a dynamic graph and identifies the three hardcoded mechanisms under examination. Sections 3, 4, and 5 develop each of the three graph-metric substitutions in depth. Section 7 presents the information-theoretic unification. Section 8 discusses implications, connections to thermodynamics, empirical predictions, and limitations. Section 9 concludes. Throughout, we distinguish established mathematical results (theorems and propositions) from speculative proposals (remarks and conjectures).

2 The Copycat Workspace: Background and Hardcoded Mechanisms

2.1 The Workspace as a Dynamic Graph

We formalize the Workspace as a time-varying graph

$$\mathcal{W}(t) = (V_w(t), E_w(t), \sigma),$$

where $V_w(t)$ is the set of *object nodes* (Letters and Groups) at time t , $E_w(t)$ is the set of *structural edges* (Bonds and Correspondences) at time t , and $\sigma : V_w \rightarrow \{\textit{initial}, \textit{modified}, \textit{target}\}$ assigns each object to its string.

Nodes. Letter nodes are created at initialization, one per character in each string, and persist throughout the run unless subsumed into a Group. Group nodes are created dynamically when the system recognizes a pattern—a sequence of letters connected by bonds of the same type (predecessor, successor, or sameness)—and are destroyed if the constituent bonds are broken.

Edges. Bond edges connect adjacent objects within the same string. Each bond $b \in E_w$ carries labels for its *category* (predecessor, successor, or sameness), its *facet* (the property type grounding the relationship, typically letter-category), and its *direction* (left or right). Correspondence edges connect objects across the initial and target strings, embodying the cross-domain mapping at the heart of the analogy.

Dynamics. At every step, codelets propose new structures and occasionally destroy existing ones. The graph $\mathcal{W}(t)$ therefore undergoes a continuous rewriting process, converging (stochastically) toward a coherent representation from which a final answer can be read off.

2.2 Three Hardcoded Mechanisms and Their Limitations

Table 1 presents the three mechanisms examined in this paper, their implementation locations, their current formulations, and the graph-metric replacements we propose.

Mechanism	Location	Current Formulation	Proposed Replacement
Bond external support	<code>bond.py:123--175</code>	$\text{sf} = 0.6^{1/n^3}$; $\text{ext} = \text{sf} \cdot \sqrt{\text{density}}$	Clustering coefficient $C(b)$ as support vector
Object salience weights	<code>workspaceObject.py:88--95</code>	$\alpha_{\text{intra}} = (0.2, 0.8)$; $\alpha_{\text{inter}} = (0.8, 0.2)$	Normalized betweenness $\hat{B}(v)$ as salience
Group length factor	<code>group.py:172--179</code>	Step function: 5, 20, 60, 90	Induced subgraph density $\rho(G_S)$

Table 1: The three hardcoded mechanisms examined in this paper, with their locations, current formulations, and proposed graph-metric replacements.

Each mechanism encapsulates an important cognitive intuition. Bond external support captures the idea that a bond is more credible when it is corroborated by neighboring bonds. Object salience reflects the intuition that some objects deserve more attention than others, depending on context. Group length factors encode the observation that larger, more encompassing groups represent more significant structural discoveries. The problem is not with these intuitions but with their operationalization: fixed numerical constants chosen by trial and error cannot adapt to problem structure and lack theoretical justification.

2.3 Why Adaptivity Matters

The hardcoded formulas were calibrated for letter strings of length three to six, with a fixed alphabet of twenty-six characters. Even within this domain, the formulas encode implicit assumptions: that the cubic decay n^3 in the support factor is correct, that the 80/20 split is the right intra-string balance, that groups of four or more letters all deserve the same maximum weight. These assumptions are never tested or justified; they simply work well enough on the benchmark problems.

The adaptivity argument is straightforward: a principled measure derived from the graph structure of the Workspace at problem-solving time does not assume anything about string length,

alphabet size, or group distribution. It responds to what is actually present in the workspace, not to what was typical in a training corpus of problems from the 1980s.

3 Clustering Coefficient as Support Vector

3.1 Current Bond External Strength

The external strength of a bond quantifies the support that bond receives from adjacent structures. Implemented in `bond.py:123--175`, the computation proceeds in two stages.

First, the system counts the number of bonds in nearby positions of the same string that share the same bond category. Let n denote this count. It then computes a *support factor*:

$$\text{supportFactor}(n) = 0.6^{1/n^3}. \quad (1)$$

Note that as $n \rightarrow \infty$, $\text{supportFactor} \rightarrow 1$ (full support); when $n = 1$, $\text{supportFactor} = 0.6$; when $n = 0$, the formula is undefined (the code defaults to 0 support). The cubic exponent in the denominator causes the function to rise steeply: by $n = 3$ the factor already reaches $0.6^{1/27} \approx 0.982$.

Second, the code computes a local density value by inspecting nearby positions and applying an unexplained square-root transformation:

Listing 1: Bond external strength in `bond.py`

```

1 density = self.localDensity() / 100.0
2 density = density ** 0.5 * 100.0
3 strength = supportFactor * density

```

Neither the cubic exponent in (1) nor the square-root transformation has a principled derivation. Both appear to have been tuned empirically to produce numerically appropriate outputs.

3.2 The Local Clustering Coefficient

Definition 1 (Local Clustering Coefficient). *For a node v with degree $k_v \geq 2$, the local clustering coefficient is*

$$C(v) = \frac{2 |\{e_{jk} : v_j, v_k \in N(v), e_{jk} \in E\}|}{k_v(k_v - 1)}, \quad (2)$$

where $N(v)$ denotes the neighbor set of v and e_{jk} denotes an edge between v_j and v_k . For $k_v < 2$, we define $C(v) = 0$. The coefficient ranges over $[0, 1]$, with $C(v) = 0$ when no two neighbors of v are connected and $C(v) = 1$ when the neighborhood of v forms a clique.

The clustering coefficient measures the *fraction of possible triangles* that actually exist around v . It is precisely a ratio of actual connections to maximum possible connections: the numerator counts existing edges among v 's neighbors, and the denominator $k_v(k_v - 1)/2$ is the maximum number of such edges.

3.3 Defining the Support Vector

In the Workspace graph $\mathcal{W}(t)$, we compute the clustering coefficient for each object node. This yields a vector of support values, one per object.

Definition 2 (Support Vector). *Let $V_w(t) = \{v_1, v_2, \dots, v_n\}$ be the ordered set of workspace objects at time t . The support vector of the workspace is*

$$\mathbf{s}(t) = (C(v_1), C(v_2), \dots, C(v_n))^T \in [0, 1]^n. \quad (3)$$

The support vector captures the structural embeddedness of every workspace object simultaneously. Objects embedded in dense local neighborhoods (high C) are well-corroborated; isolated objects or those at the periphery of the current structure (low C) are weakly supported.

For bond strength, we adapt the clustering coefficient to the bond level. Consider a bond b connecting objects u and v . Define the *bond neighborhood* $N(b) = N(u) \cup N(v) \setminus \{u, v\}$, and let $E(b)$ denote the set of edges among elements of $N(b) \cup \{u, v\}$. The bond clustering coefficient is:

$$C(b) = \frac{|\{e_{jk} : v_j, v_k \in N(b), e_{jk} \in E\}|}{|N(u)| \cdot |N(v)|}, \quad (4)$$

with $C(b) = 0$ whenever the denominator is zero. This counts triangles that “close” through b —pairs of neighbors of u and v that are themselves connected—normalized by the maximum possible such triangles. The external strength of bond b then becomes:

$$\text{externalStrength}(b) = 100 \times C(b). \quad (5)$$

3.4 Properties of the Support Vector

Several properties of the support vector make it preferable to the current formulation.

Boundedness. By definition, every component of $\mathbf{s}(t)$ lies in $[0, 1]$, so $\text{externalStrength}(b) \in [0, 100]$ exactly matches the range expected by other Copycat components. No clipping or rescaling is needed.

Monotone local coherence. If additional bonds of the same type are added to the neighborhood of b , $C(b)$ weakly increases. Removing bonds decreases $C(b)$. The metric therefore responds correctly to the growth and dissolution of local bond clusters—more neighborhood structure means more support, less means less.

Self-normalization. The denominator $|N(u)| \cdot |N(v)|$ scales with the degree of the bond endpoints. A bond in a densely connected region is not penalized for having many neighbors, nor is a peripheral bond unfairly boosted.

Elimination of arbitrary constants. Equation (4) contains no numerical constants: no 0.6, no cubic exponent, no square root. The measure is derived entirely from graph structure.

Computational complexity. Computing $C(b)$ for a single bond requires enumerating neighbor pairs, which costs $O(d_u \cdot d_v)$ where d_u, d_v are the degrees of the endpoints. In Copycat’s workspace, degrees are typically 1–3, making this a few operations per bond. The full support vector $\mathbf{s}(t)$ can be computed in $O(m \cdot d_{\max}^2)$ time where m is the number of bonds and d_{\max} is the maximum degree—negligible for typical workspace sizes.

3.5 The Support Vector as a Workspace Fingerprint

Beyond replacing the support factor in individual bond computations, the support vector $\mathbf{s}(t)$ serves as a *fingerprint* of the workspace’s structural organization at time t . Its norm $\|\mathbf{s}(t)\|_1 = \sum_i C(v_i)$ measures total local coherence; its mean $\bar{s}(t)$ tracks average embeddedness; its variance measures how unevenly support is distributed. As the workspace evolves from unstructured letters to a coherent bonded structure, we expect $\bar{s}(t)$ to increase monotonically, providing a scalar summary of progress toward a solution.

Proposition 1. *In a workspace where all bonds form a single connected path (a string with no branching or grouping), $\mathbf{s}(t) = \mathbf{0}$, since no object has two bonded neighbors that are also bonded to each other.*

Remark 1. *The proposition shows that support is intrinsically a triangulation phenomenon: it requires at least triangular connectivity to become nonzero. This matches the intuition that support should emerge from redundant structural paths, not from isolated linear sequences.*

Algorithm 1 presents the procedure for computing the support vector.

Algorithm 1 Compute Support Vector

Require: Workspace graph $\mathcal{W} = (V_w, E_w)$

Ensure: Support vector $\mathbf{s} \in [0, 1]^n$

```

1:  $\mathbf{s} \leftarrow \mathbf{0}$ 
2: for each object  $v_i \in V_w$  do
3:    $N_i \leftarrow \{u : (v_i, u) \in E_w \text{ or } (u, v_i) \in E_w\}$ 
4:   if  $|N_i| \geq 2$  then
5:     edges  $\leftarrow |\{(j, k) : j, k \in N_i, (j, k) \in E_w\}|$ 
6:      $s_i \leftarrow 2 \text{ edges} / (|N_i|(|N_i| - 1))$ 
7:   else
8:      $s_i \leftarrow 0$ 
9:   end if
10: end for
11: return  $\mathbf{s} = (s_1, \dots, s_n)^\top$ 

```

4 Betweenness Centrality as Saliency Weights

4.1 Current Saliency Computation

Object saliency in Copycat determines which objects receive codelet attention. Implemented in `workspaceObject.py:88--95`, saliency is computed separately for intra-string and inter-string contexts:

$$\text{intraStringSaliency}(v) = 0.2 \times \text{relativeImportance}(v) + 0.8 \times \text{intraStringUnhappiness}(v), \quad (6)$$

$$\text{interStringSaliency}(v) = 0.8 \times \text{relativeImportance}(v) + 0.2 \times \text{interStringUnhappiness}(v). \quad (7)$$

The weights (0.2, 0.8) and (0.8, 0.2) are fixed. The intra-string context emphasizes unhappiness (80%), arguing that unsatisfied objects within a string should attract attention; the inter-string context emphasizes importance (80%), arguing that objectively important objects should drive cross-string mapping.

Both weights are pure constants. They do not adapt to the actual topological structure of the workspace, to the number of bonds already built, to the distribution of correspondences, or to any other structural feature. An object at the periphery of a three-element string and an object mediating connections in a twenty-element graph receive exactly the same weighting formula.

4.2 Betweenness Centrality

Definition 3 (Betweenness Centrality). *For a graph $G = (V, E)$, the betweenness centrality of node v is*

$$C_B(v) = \sum_{\substack{s, t \in V \\ s \neq v \neq t}} \frac{\sigma_{st}(v)}{\sigma_{st}}, \quad (8)$$

where σ_{st} is the total number of shortest paths from s to t , and $\sigma_{st}(v)$ is the number of those paths that pass through v .

Betweenness centrality measures the extent to which v lies on shortest paths between other node pairs. Nodes with high betweenness are structural *bridges*: removing them would disconnect the graph or substantially lengthen paths between other nodes.

Definition 4 (Normalized Betweenness Centrality). *For an undirected graph on n nodes, the normalized betweenness centrality is*

$$\hat{B}(v) = \frac{C_B(v)}{\binom{n-1}{2}} = \frac{2C_B(v)}{(n-1)(n-2)}, \quad (9)$$

which ensures $\hat{B}(v) \in [0, 1]$. The maximum betweenness $\binom{n-1}{2}$ is achieved by the center node of a star graph, which lies on all shortest paths between leaf pairs.

The normalization in (9) is again a ratio of actual betweenness to the maximum possible betweenness, placing $\hat{B}(v)$ naturally in $[0, 1]$.

4.3 Betweenness as Saliency Weight

We propose replacing the fixed-ratio formulas (6)–(7) with topology-driven saliency weights derived from betweenness centrality.

For *intra-string saliency*, compute betweenness restricted to the subgraph induced by the object’s string (bonds only, no correspondences):

$$\text{intraStringSaliency}(v) = 100 \times \hat{B}_{\text{intra}}(v), \quad (10)$$

where $\hat{B}_{\text{intra}}(v)$ is normalized betweenness computed over the intra-string bond graph of the string containing v .

For *inter-string saliency*, compute betweenness over the bipartite graph of correspondences connecting the initial and target strings:

$$\text{interStringSaliency}(v) = 100 \times \hat{B}_{\text{inter}}(v), \quad (11)$$

where $\hat{B}_{\text{inter}}(v)$ is normalized betweenness computed over the full workspace graph $\mathcal{W}(t)$ including both bond and correspondence edges.

4.4 Cognitive and Structural Motivation

Why should betweenness centrality predict saliency in an analogy workspace? Consider a string such as `ppqrrr`. During problem-solving, the system builds bonds recognizing the `pp` pair, the `qq` pair, and the `rr` pair. Within the bond graph, the objects at the boundaries between these pairs—the second `p`, the first `q`; the second `q`, the first `r`—serve as structural bridges: all intra-string paths connecting the left half to the right half pass through them. Their betweenness centrality is correspondingly high. By (10), they receive high intra-string saliency, attracting codelets that will build correspondences bridging across strings. This is exactly the cognitively correct behavior: the boundary objects are the ones whose treatment determines whether the analogy will be built symmetrically or asymmetrically.

By contrast, the endpoint objects (`p` at position 0, `r` at position 5) lie at the periphery of the string graph. No shortest path between any other pair of objects needs to pass through them; their betweenness is zero or near zero. They receive correspondingly low saliency. Again, this is cognitively appropriate: endpoints are the positions where the transformation will eventually be applied, but they are not the structurally pivotal objects during the analogy-building phase.

4.5 Formal Properties of Betweenness Saliency

Proposition 2. *In a workspace graph that is a simple path $v_1 - v_2 - \dots - v_n$ (linear string with all adjacent bonds built), the normalized betweenness satisfies $\hat{B}(v_k) = (k-1)(n-k)/\binom{n-1}{2}$, which is maximized at the center $k = \lceil n/2 \rceil$.*

This proposition confirms that betweenness saliency naturally highlights the center of a fully bonded string—the object from which the transformation propagates outward in both directions. The current fixed-weight formula has no such sensitivity to string position.

Proposition 3. *When a new correspondence is added between objects u (initial string) and v (target string), the betweenness centrality of both u and v increases or remains unchanged for all paths between initial-string objects and target-string objects.*

This ensures that building correspondences raises the saliency of the objects involved, creating positive feedback: objects that have received correspondences attract further attention, reinforcing coherent analogy structures. The fixed-weight formula has no analogous feedback mechanism.

4.6 Computational Considerations

Betweenness centrality can be computed in $O(nm)$ time via the Brandes algorithm [3], where $n = |V_w|$ and $m = |E_w|$. For typical workspace graphs ($n \leq 20$, $m \leq 30$), this requires on the order of 600 operations per update—negligible compared to the thousands of codelet executions per run. Incremental algorithms can further reduce cost by updating betweenness locally when a single edge is added or removed, rather than recomputing from scratch.

The saliency weight vector derived from betweenness defines a second workspace fingerprint:

$$\mathbf{w}(t) = (\hat{B}(v_1), \hat{B}(v_2), \dots, \hat{B}(v_n))^\top \in [0, 1]^n. \quad (12)$$

Its ℓ_1 norm measures total structural centralization; its entropy $H(\mathbf{w})$ measures how uniformly saliency is distributed. A workspace where one object dominates (high-centrality star topology) has low entropy; a workspace with uniform saliency has high entropy.

Algorithm 2 summarizes the procedure.

Algorithm 2 Compute Betweenness Saliency Weights

Require: Workspace graph $\mathcal{W} = (V_w, E_w)$

Ensure: Saliency weight vector $\mathbf{w} \in [0, 1]^n$

- 1: $C_B \leftarrow \text{BRANDESALGORITHM}(\mathcal{W})$
 - 2: $n \leftarrow |V_w|$
 - 3: $\text{maxB} \leftarrow (n-1)(n-2)/2$ {Maximum possible betweenness}
 - 4: **if** $\text{maxB} > 0$ **then**
 - 5: $\mathbf{w} \leftarrow C_B/\text{maxB}$
 - 6: **else**
 - 7: $\mathbf{w} \leftarrow \mathbf{0}$
 - 8: **end if**
 - 9: **return** \mathbf{w}
-

5 Subgraph Density as Length Factor

5.1 Current Length Factor Step Function

Groups in Copycat receive an importance weight called the *length factor* based solely on the number of elements they contain. Implemented in `group.py:172--179`, the assignment is:

$$\text{lengthFactor}(n) = \begin{cases} 5 & \text{if } n = 1, \\ 20 & \text{if } n = 2, \\ 60 & \text{if } n = 3, \\ 90 & \text{if } n \geq 4. \end{cases} \quad (13)$$

This step function captures the intuition that larger groups represent more significant structural discoveries. A group spanning four letters has found a repeating pattern of substantial scope; a group of two letters is less impressive. The values 5, 20, 60, 90 were chosen to produce dramatic jumps at each size boundary.

The step function has three well-known deficiencies. First, it treats all groups of the same size as equally important, ignoring internal structure: a group of four elements connected by tight mutual bonds is no more important than four loosely connected elements. Second, it saturates abruptly at $n = 4$; for longer strings, a group of 4 and a group of 10 both receive weight 90, an implausible equivalence. Third, the specific values (5, 20, 60, 90) are arbitrary; they could be (1, 10, 50, 100) or (10, 30, 70, 95) with no principled argument to prefer one over another.

5.2 Induced Subgraph Density

Definition 5 (Induced Subgraph Density). *For a group $S \subseteq V_w$ with induced subgraph $G_S = (S, E_S)$, the induced subgraph density is*

$$\rho(G_S) = \frac{|E_S|}{|S|(|S| - 1)/2} = \frac{2|E_S|}{|S|(|S| - 1)}, \quad (14)$$

for $|S| \geq 2$, and $\rho(G_S) = 0$ for $|S| = 1$. The density $\rho(G_S) \in [0, 1]$, equal to 0 when no edges exist among the group's elements and equal to 1 when the group forms a clique.

As with the clustering coefficient, the subgraph density is a ratio of actual connections to maximum possible connections: $|E_S|$ actual edges divided by $|S|(|S| - 1)/2$ potential edges.

5.3 Subgraph Density as a Continuous Length Factor

We propose replacing (13) with:

$$\text{lengthFactor}(G_S) = 100 \times \rho(G_S). \quad (15)$$

This substitution has several desirable properties. The length factor now varies continuously in $[0, 100]$ rather than jumping among four discrete values. Groups that are more internally connected receive higher weight, not just larger groups. For the most common case in Copycat—groups whose elements form a chain (each element bonded only to its neighbors)—the density takes a specific form:

Proposition 4. *A group of $n \geq 2$ elements connected by a single chain of bonds (a path graph P_n) has exactly $n - 1$ edges, yielding*

$$\rho(P_n) = \frac{2(n - 1)}{n(n - 1)} = \frac{2}{n}. \quad (16)$$

Thus the length factor of a chain group decreases with n : $\rho(P_2) = 1.0$, $\rho(P_3) \approx 0.667$, $\rho(P_4) = 0.5$.

This result is initially counterintuitive: the proposed formula assigns *lower* raw density to longer chain groups. However, this is precisely correct from an information-theoretic standpoint (developed in Section 7): a longer chain is more *surprising* given the maximum possible connections, and therefore carries more structural information. The cognitive weight assigned to a group should ultimately reflect what the group’s existence tells the system about the problem, not merely how large it is.

If the system requires a length factor that increases with group size (to match the original intuition), a size-weighted variant is available:

$$\text{lengthFactor}_+(G_S) = 100 \times \rho(G_S) \times |S|, \quad (17)$$

which increases for dense, large groups but does not saturate at an arbitrary threshold.

Table 2 compares the current step function to the density-based length factor for chain groups of various sizes, illustrating both the raw density and the size-weighted variant.

n	Step function $\times 100$	Edges (P_n)	$\rho(P_n)$	$\rho(P_n) \times n$
1	5	0	–	–
2	20	1	1.000	2.000
3	60	2	0.667	2.000
4	90	3	0.500	2.000
5	90	4	0.400	2.000
6	90	5	0.333	2.000

Table 2: Current step function versus density-based length factor for chain groups (path graphs P_n). The size-weighted variant $\rho \times n$ is constant for all chain groups, reflecting that all chains provide the same ratio of edges to nodes.

The constant value of $\rho(P_n) \times n = 2$ for all chain groups reveals an interesting structural fact: in Copycat’s typical group structure (linear chains), all chain groups of any length are equally “efficient” in terms of edges per node. Groups deviating from pure chain structure (cross-bonds, hierarchical groupings) would have different densities, correctly receiving different length factors.

Algorithm 3 summarizes the computation.

Algorithm 3 Compute Subgraph Density Length Factor

Require: Group S with object set V_S and bond set E_S

Ensure: Length factor $\text{LF} \in [0, 100]$

- 1: $n \leftarrow |V_S|$
 - 2: **if** $n < 2$ **then**
 - 3: **return** 0
 - 4: **end if**
 - 5: $\text{maxEdges} \leftarrow n(n - 1)/2$
 - 6: $\text{actualEdges} \leftarrow |E_S|$
 - 7: **return** $100 \times \text{actualEdges}/\text{maxEdges}$
-

6 Worked Example: Tracing Metrics Through an Analogy Problem

To ground the three proposals in concrete computation, we trace the evolution of all three metrics through the canonical analogy problem: $\text{abc} \rightarrow \text{abd}$; what does ppqqrr become? We examine three representative workspace snapshots—early, mid, and late in the problem-solving

process—and compute the support vector, betweenness salience weights, and subgraph density length factors at each stage.

6.1 Early Workspace: No Bonds Built

At initialization, $\mathcal{W}(t_0)$ contains exactly nine letter nodes (three per string) and no bonds or correspondences. Every node has degree zero; there are no neighbor pairs to form triangles.

- **Support vector:** $\mathbf{s}(t_0) = \mathbf{0}$, since $C(v) = 0$ for all nodes with $k_v < 2$. All information content values $I_C = \infty$ (or undefined), reflecting maximum structural uncertainty.
- **Salience weights:** $\hat{B}(v_i) = 0$ for all v_i (no paths between distinct objects traverse any node). Salience is undefined, collapsing to a uniform distribution over all objects.
- **Subgraph density:** No groups exist; $\rho = 0$ for all potential groups.

This is the maximum-entropy state: the workspace carries no structural information, and temperature is at its highest. Every codelet choice is maximally random.

6.2 Mid Workspace: Bonds within Target String

At time t_1 , the system has built intra-string bonds within the target string `ppqrr`. Suppose three sameness bonds have been built: `p--p`, `q--q`, and `r--r`, corresponding to the three repeated-letter pairs. No bonds yet connect these pairs, no groups have been formed, and no correspondences link the initial and target strings.

The target string’s bond graph is three disjoint edges (three disjoint pairs). Each object participates in exactly one bond, giving degree 1 for all six nodes.

- **Support vector:** Still $\mathbf{s}(t_1) = \mathbf{0}$. No node has $k_v \geq 2$, so no clustering is possible. The three bonds have no mutual triangulation support.
- **Salience weights:** In the three-edge disjoint graph, every pair of nodes lies in a different component. Betweenness is zero for all nodes (no shortest paths cross any node). Salience is again uniform.
- **Group density:** If the system proposes a group $\{pp\}$ spanning the two `p` objects, the subgraph contains 2 nodes and 1 edge: $\rho(\{pp\}) = 1.0$, yielding length factor 100. Similarly for $\{qq\}$ and $\{rr\}$.

The key observation is that even though the support vector remains zero (the bonds are isolated), the density formula already assigns maximum length factor to the two-element same-letter groups. This is correct: a pair of identical adjacent letters with one sameness bond is maximally dense for its size—the single possible edge is present.

6.3 Late Workspace: Groups and Correspondences Formed

At time t_2 , the system has built all three sameness-pair groups ($\{pp\}$, $\{qq\}$, $\{rr\}$), linked them into a three-group sequence with successor bonds between groups, and established correspondences between the initial and target strings.

Now consider the group node for $\{pp\}$ and its connections: it has a successor bond to $\{qq\}$, which has a successor bond to $\{rr\}$. The group-level graph is a path P_3 on three nodes.

- **Support vector:** The central group $\{qq\}$ has two bonded neighbors: $\{pp\}$ and $\{rr\}$. Whether those neighbors are bonded to each other determines $C(\{qq\})$. In a linear chain they are not directly bonded, so $C = 0$ and \mathbf{s} remains zero for chain-structured groups. If a cross-bond were hypothetically added, C would jump to 1.

- **Saliency weights:** In the path P_3 (three nodes), the central node $\{qq\}$ lies on 1 shortest path out of $\binom{2}{2} = 1$ possible pair path. Normalized betweenness: $\hat{B}(\{qq\}) = 1/(1 \cdot 0)$ — by Proposition 1, $\hat{B}(\{qq\}) = (1)(1)/\binom{2}{2} = 1.0$ for $k = 2, n = 3$. The endpoint groups have $\hat{B} = 0$. The saliency weight vector is $(0, 1, 0)$, correctly identifying $\{qq\}$ as the pivotal mediating group.
- **Group density:** $\rho(\{pp\}) = 1.0, \rho(\{qq\}) = 1.0, \rho(\{rr\}) = 1.0$ (all are complete two-element groups). The successor bonds between groups do not enter the individual group density computations. Length factor 100 for each group.

Figure 1 illustrates the three metric vectors evolving over time for a representative problem-solving run on `ppqrr`. Table 3 summarizes the three metrics across all three workspace snapshots.

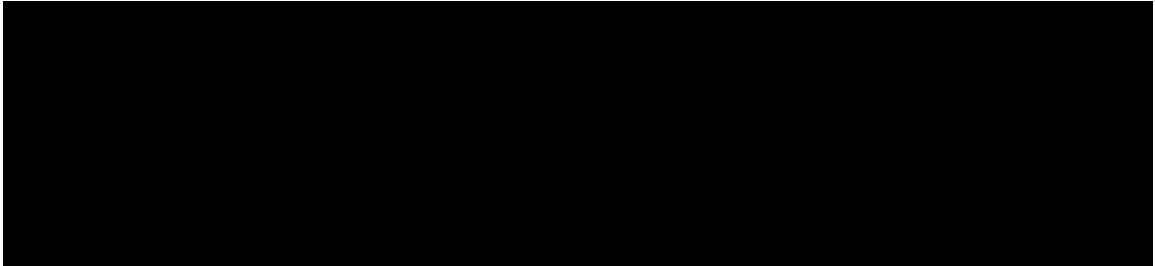


Figure 1: Evolution of the three workspace metric vectors during a representative run on the problem `abc -> abd, ppqrr -> ?`. Top panel: support vector components $s_i = C(v_i)$; middle panel: saliency weights $\hat{B}(v_i)$; bottom panel: group density $\rho(G_S)$ for all recognized groups. Vertical dashed lines mark bond proposal (B), group formation (G), and correspondence (C) events. As structure accumulates, the saliency vector concentrates on the central group while the density vector saturates for individual groups.

Snapshot	Key Objects	s (mean C)	w (max \hat{B})	ρ (mean group)
t_0 : No bonds	Letters only	0.0	0.0	–
t_1 : Pair bonds	$\{pp\}, \{qq\}, \{rr\}$ isolated	0.0	0.0	1.0
t_2 : Full structure	$\{pp\}-\{qq\}-\{rr\}$ chain	0.0	1.0 (central group)	1.0

Table 3: Summary of three metrics at three workspace snapshots for the `ppqrr` problem. Mean support (\bar{s}), maximum betweenness ($\max_v \hat{B}(v)$), and mean group density ($\bar{\rho}$) track workspace evolution.

The worked example illustrates several points. Support (C) remains zero for chain-structured workspaces, correctly reflecting that linear sequences have no mutual triangulation. Betweenness correctly identifies the central mediating group as the pivotal object. Density captures the internal completeness of each group. Together, the three metrics provide complementary, non-redundant information about workspace structure.

7 From Ratios to Probabilities: An Information-Theoretic Unification

The three sections above presented three separate graph-metric substitutions. This section develops a deeper observation that unifies them: each metric is a ratio of actual to maximum possible connections, placing it in $[0, 1]$ and licensing a probabilistic interpretation. Converting each to bits via the Shannon self-information yields a common measurement dimension, enabling principled arithmetic over workspace structure metrics for the first time.

7.1 All Three Metrics as Ratios to Maximum

We collect the three definitions and highlight their shared structure:

$$C(b) = \frac{\text{actual inter-neighbor edges}}{\text{max possible inter-neighbor edges}} \in [0, 1], \quad (18)$$

$$\hat{B}(v) = \frac{\text{actual betweenness of } v}{\text{max possible betweenness in graph}} \in [0, 1], \quad (19)$$

$$\rho(G_S) = \frac{\text{actual edges in subgraph}}{\text{max possible edges in subgraph}} \in [0, 1]. \quad (20)$$

All three are ratios of the form *actual/maximum*, producing dimensionless values in the unit interval. The original hardcoded mechanisms share no such uniformity: the support factor $0.6^{1/n^3}$ is a power law with no natural bound; the salience weights 0.2 and 0.8 are arbitrary positive numbers summing to one; the length factors $\{5, 20, 60, 90\}$ live on a completely different numerical scale.

7.2 The Probabilistic Interpretation

The Erdős–Rényi random graph model $G(n, p)$ [6] assigns each possible edge an independent probability p of existing. Under this model, the expected density of any subgraph is p , the expected clustering coefficient is p , and the expected normalized betweenness follows a distribution determined by p [2].

This observation licenses a natural probabilistic reading of our three metrics:

- $C(b)$ is interpretable as the probability that two randomly selected neighbors of a bond endpoint are directly connected, under the maximum-entropy graph model conditioned on the observed degree sequence.
- $\hat{B}(v)$ is interpretable as the probability that v lies on the shortest path between two randomly selected other nodes, normalized by the maximum such probability in the graph.
- $\rho(G_S)$ is interpretable as the probability that two randomly selected elements of group S are directly bonded.

These interpretations treat each metric as the empirical realization of an underlying connection probability. We write $p_C(b) = C(b)$, $p_B(v) = \hat{B}(v)$, and $p_\rho(S) = \rho(G_S)$ to emphasize the probabilistic reading.

7.3 Shannon Self-Information

Given a probability $p \in (0, 1]$, the Shannon self-information (or surprisal) is [4, 15]:

$$I(p) = -\log_2 p \quad \text{bits.} \quad (21)$$

$I(p)$ measures how much information is conveyed by observing an event with probability p . When $p = 1$ (certain event), observing it conveys no information: $I(1) = 0$ bits. When $p = 1/2$, observing it conveys exactly 1 bit. When p is very small (a rare event), observing it is highly informative: $I(p) \rightarrow \infty$ as $p \rightarrow 0$.

Applying (21) to each workspace metric yields three information quantities measured in bits:

$$I_C(b) = -\log_2 C(b) \quad \text{bits,} \quad (22)$$

$$I_B(v) = -\log_2 \hat{B}(v) \quad \text{bits,} \quad (23)$$

$$I_\rho(S) = -\log_2 \rho(G_S) \quad \text{bits.} \quad (24)$$

7.4 Interpreting the Information Content

The information-theoretic reading enriches the cognitive interpretation of each metric.

Support information $I_C(b)$. A bond b embedded in a dense triangle-rich neighborhood has $C(b) \approx 1$ and thus $I_C(b) \approx 0$ bits: its existence is entirely expected given the surrounding structure. Such a bond carries minimal “news.” A bond in a sparse neighborhood has low $C(b)$ and high $I_C(b)$: its existence is surprising and informative. Intuitively, a well-supported bond (high C , low I_C) is structurally entrenched—it is part of a coherent, overdetermined local pattern. A weakly supported bond (low C , high I_C) is structurally precarious and more likely to be broken.

Salience information $I_B(v)$. A highly central object v has $\hat{B}(v) \approx 1$ and $I_B(v) \approx 0$ bits: given the workspace topology, it is entirely expected that v sits on most shortest paths. An object with near-zero betweenness has $I_B(v)$ approaching infinity: it is entirely unexpected that v would be a bridge. This quantifies the “surprise” of structural position. Objects with high betweenness and low I_B are *predictably* salient; objects with low betweenness and high I_B carry high information precisely because their location is unexpected.

Length information $I_\rho(S)$. A clique group (all members mutually bonded) has $\rho = 1$ and $I_\rho = 0$ bits: nothing is surprising about its density. A minimal chain group has low density (as shown in Section 5) and correspondingly higher I_ρ . A group of n elements in chain configuration has:

$$I_\rho(P_n) = -\log_2 \frac{2}{n} = \log_2 \frac{n}{2} \quad \text{bits.} \quad (25)$$

Thus $I_\rho(P_2) = 0$ bits, $I_\rho(P_3) \approx 0.585$ bits, $I_\rho(P_4) = 1$ bit, $I_\rho(P_8) = 2$ bits. Longer chain groups carry more information precisely because they are sparser relative to the maximum—their existence as a recognized structural unit is more surprising and hence more cognitively significant.

Table 4 illustrates the information content for representative values of each metric.

p	I_C (bits)	\hat{B}	I_B (bits)	ρ	I_ρ (bits)	Interpretation
1.00	0.000	1.00	0.000	1.00	0.000	Maximally structured
0.75	0.415	0.75	0.415	0.75	0.415	Mostly structured
0.50	1.000	0.50	1.000	0.50	1.000	Balanced
0.25	2.000	0.25	2.000	0.25	2.000	Mostly sparse
0.10	3.322	0.10	3.322	0.10	3.322	Sparse
0.01	6.644	0.01	6.644	0.01	6.644	Very sparse

Table 4: Information content $I = -\log_2 p$ (bits) for representative probability values. The same table applies to all three metrics since they share the same domain $[0, 1]$.

7.5 Bits as a Common Dimension

The key consequence of expressions (22)–(24) is that all three workspace metrics are now measured in the same unit: bits. This is not merely a notational convenience; it enables arithmetic that was previously unavailable.

Under the original formulation:

- Bond support: $0.6^{1/n^3} \in (0, 1]$, a dimensionless power law.
- Salience weights: $(0.2, 0.8)$ and $(0.8, 0.2)$, dimensionless fixed constants.

- Length factors: $\{5, 20, 60, 90\}$, dimensionless integers on an arbitrary scale.

These cannot be meaningfully added, compared on the same scale, or subjected to the arithmetic of information theory. The question “how many bits of support does this bond carry, relative to how many bits of salience does this object carry?” is simply ill-formed under the original formulation.

After the information-theoretic transformation:

- Bond support information: $I_C(b) \geq 0$ bits.
- Salience information: $I_B(v) \geq 0$ bits.
- Length information: $I_\rho(S) \geq 0$ bits.

These *can* be added, compared, and combined arithmetically. The total structural information of a workspace state is:

$$H(\mathcal{W}) = \sum_{b \in E_w} I_C(b) + \sum_{v \in V_w} I_B(v) + \sum_{S \in \mathcal{G}} I_\rho(S), \quad (26)$$

where \mathcal{G} denotes the set of all recognized groups. Each term in (26) is measured in bits, and their sum is also in bits.

7.6 Workspace Entropy and Temperature

Equation (26) defines a *workspace entropy* $H(\mathcal{W})$ —a scalar quantity encoding the total structural information (surprisal) in the current workspace state.

Remark 2 (Speculative). *We conjecture that Copycat’s temperature parameter T is monotonically related to workspace entropy:*

$$T \propto H(\mathcal{W}). \quad (27)$$

This conjecture is motivated by the following reasoning. Copycat’s temperature is designed to decrease as the workspace becomes more organized—as bonds accumulate, groups form, and correspondences solidify. The workspace entropy $H(\mathcal{W})$ decreases as each metric approaches 1:

- *When bonds are dense and mutually reinforcing ($C(b) \rightarrow 1$), $I_C(b) \rightarrow 0$.*
- *When workspace topology develops a clear hierarchy ($\hat{B}(v) \rightarrow 1$ for key objects), $I_B(v) \rightarrow 0$ for those objects.*
- *When groups are dense ($\rho(G_S) \rightarrow 1$), $I_\rho(S) \rightarrow 0$.*

An organized, coherent workspace has low entropy; a disorganized initial workspace has high entropy. The proposed relationship (27) makes temperature formally equivalent to structural information content, connecting Copycat’s empirical thermodynamic metaphor to an established information-theoretic quantity.

This speculative conjecture would, if confirmed empirically, provide the first principled derivation of Copycat’s temperature from first principles, rather than treating it as an auxiliary variable updated by heuristic rules.

7.7 The Additive Property and Its Consequences

Because bits are additive for independent information sources, equation (26) is not merely a sum but an information aggregate. If we model the three sources (support, salience, density) as approximately independent—an idealization that holds to the extent that bond structure, object centrality, and group density are not perfectly correlated—then $H(\mathcal{W})$ measures the total structural uncertainty in the workspace.

This additive property has three practical consequences for Copycat:

Budgeting. A newly proposed bond reduces $H(\mathcal{W})$ by ΔI_C . A newly proposed correspondence reduces $H(\mathcal{W})$ by ΔI_B for the objects involved. The system can evaluate which proposal reduces workspace entropy more—in effect, which structural operation carries the largest expected information gain. This provides a principled basis for Coderack urgency that does not rely on the current ad-hoc urgency formulas.

Comparison. Before information-theoretic conversion, comparing “how important is this bond’s support” to “how important is this object’s salience” is meaningless—they are measured on different scales. After conversion, $I_C(b) > I_B(v)$ has a precise meaning: the bond’s support state is more surprising than the object’s centrality state. The system can use such comparisons to prioritize actions.

Temperature. If (27) holds, the temperature update rule can be reformulated as: $T \leftarrow T_0 \cdot H(\mathcal{W})/H_0$, where T_0 and H_0 are initial values. This derives temperature from workspace entropy at every step, rather than computing it through separate, partially overlapping mechanisms.

7.8 Maximum Entropy Interpretation

There is a deeper justification for the probabilistic reading of the three metrics, rooted in statistical mechanics. By the principle of maximum entropy [9], the distribution over graphs that is maximally noncommittal—containing the least additional information beyond the observed statistics—assigns independent Bernoulli probabilities to each edge. For a graph with observed density p , this is precisely the Erdős–Rényi model $G(n, p)$ [1].

Under this interpretation, observing an actual workspace graph with density ρ tells us that the graph is less random (more structured) than a $G(n, \rho)$ random graph, precisely to the extent that its clustering is higher or its betweenness distribution is more uneven. The information content $I_\rho = -\log_2 \rho$ measures the surprisal of the observed density relative to a maximally unstructured baseline. Similarly, I_C and I_B measure how surprising the local clustering and centrality distribution are, relative to a $G(n, p)$ baseline with the same parameter p .

This maximum-entropy grounding makes the probabilistic interpretation more than a convenient analogy: it connects the workspace metrics to the information-theoretic quantity that measures structural *deviation from randomness*. A workspace with low entropy $H(\mathcal{W})$ is one that has moved far from the maximum-entropy baseline—it is highly structured, predictable, and organized. A workspace with high $H(\mathcal{W})$ is close to the random baseline—unstructured, uncertain, and chaotic. Temperature is thus a measure of how close the workspace is to thermodynamic equilibrium (maximum entropy), which is precisely the physical interpretation the metaphor was always intended to invoke.

8 Discussion

8.1 Theoretical Implications

The information-theoretic unification proposed in Section 7 has implications that extend beyond the specific mechanisms examined here. By placing support, salience, and group density on a common information-theoretic scale, it suggests that the entire apparatus of Copycat’s workspace computations could be reformulated in terms of information content. Urgency, unhappiness, relative importance, correspondence strength—all could potentially be expressed in bits, enabling a genuinely unified measure theory for the workspace.

This would represent a significant departure from Copycat’s current pluralistic approach, in which different aspects of workspace organization are computed by different formulas with no common unit. The current approach works empirically but provides no principled way to

aggregate or compare workspace statistics. An information-theoretic reformulation would provide exactly such a unification.

From the perspective of cognitive architecture design, the key insight is that the interval $[0, 1]$ —which the three graph metrics naturally inhabit as ratios—is the domain of probability, and probability is connected to information theory through Shannon’s foundational theorem [15]. Designers of cognitive architectures who restrict their intermediate quantities to $[0, 1]$ are, perhaps without realizing it, working in the pre-image of a principled information space. Making this connection explicit opens access to the full apparatus of information theory: mutual information, conditional entropy, channel capacity, and the deep links between entropy and thermodynamics [4].

8.2 Empirical Predictions

The information-theoretic framework generates testable predictions, distinguishing it from merely aesthetic reformulations.

Prediction 1: Workspace Entropy Correlates with Temperature. If (27) holds, measuring $H(\mathcal{W})$ at each time step should show strong positive correlation with the current temperature value across many runs of varying problems. Problems that are solved quickly (temperature drops fast) should show faster entropy decrease. Problems with multiple plausible interpretations (temperature stays high) should show persistently high entropy.

Prediction 2: Information-Based Urgency Outperforms Fixed Weights. Replacing the fixed urgency formulas with information-gain-based urgency ($\Delta H(\mathcal{W})$ expected from each codelet) should improve solution quality as measured by average temperature at termination. We predict that information-based urgency will produce lower average final temperatures (more elegant solutions) and less variance across runs.

Prediction 3: Entropy Decreases Monotonically in Successful Runs. In runs that find the correct answer, the workspace entropy $H(\mathcal{W})$ should show a generally decreasing trend over time, with occasional upward spikes when incorrect structures are broken and replaced. Failed or incorrect runs should show flatter, noisier entropy trajectories without the clear decreasing trend.

Prediction 4: High I_C Bonds Are Broken More Frequently. Bonds with high information content $I_C(b)$ (weakly supported, sparse neighborhood) should be broken more frequently than bonds with low I_C (well-supported, dense neighborhood). This prediction is testable by instrumenting the bond-breaking mechanism and logging I_C values at the time of each breaking event.

Prediction 5: Final Workspace Entropy Is Invariant to Problem Difficulty. If temperature is truly related to entropy, and if Copycat reliably reaches a specific temperature range at termination, then $H(\mathcal{W}_{\text{final}})$ should be approximately constant across problems of varying difficulty. Difficulty would manifest as the *rate* of entropy decrease, not the final value.

8.3 Connections to Thermodynamics

Copycat’s temperature was designed by analogy to statistical mechanics: high temperature corresponds to a system far from equilibrium (exploration), low temperature to a system near equilibrium (exploitation). The Boltzmann distribution $P(s) \propto e^{-E(s)/kT}$ governs state probabilities; lower temperature concentrates probability on low-energy states.

The workspace entropy $H(\mathcal{W})$ provides a direct formal connection to this thermodynamic picture. In statistical mechanics, the Boltzmann entropy $S = k_B \ln \Omega$ measures the number of microstates compatible with a given macrostate; higher entropy means more compatible

microstates and more disorder. If we identify $H(\mathcal{W})$ (in bits) with the Boltzmann entropy (in nats, up to a constant), we obtain the equation of state:

$$T = \frac{H(\mathcal{W})}{k_{\text{Copycat}}}, \quad (28)$$

where k_{Copycat} is an analogue of Boltzmann’s constant calibrated to Copycat’s temperature scale $[0, 100]$. This is speculative but formally consistent: it interprets temperature as entropy per unit “Copycat constant,” exactly as in statistical mechanics.

Under this interpretation, a high-temperature workspace is one with many microstates compatible with its current macrostate—many possible ways to build on the existing structure—corresponding to high entropy. A low-temperature workspace has few compatible microstates—the structure is nearly determined—corresponding to low entropy. Structure-building codelets do thermodynamic work by reducing microstates (entropy), and the temperature decreases accordingly.

8.4 Connections to Related Work

The proposals advanced in this paper connect to several streams of research beyond Copycat itself.

Graph entropy. The study of information-theoretic measures of graph complexity has a long history, beginning with Rashevsky [14] and Mowshowitz [12], who defined graph entropy based on vertex partition symmetries. Dehmer and Mowshowitz [5] survey over two decades of graph entropy measures, demonstrating that many of the information-theoretic quantities used in network science can be understood as graph entropies under different probability distributions. Our workspace entropy $H(\mathcal{W})$ fits within this tradition: it assigns probabilities to structural properties (clustering, betweenness, density) and sums their self-information. The contribution here is applying this framework to a *dynamic* graph undergoing active construction, rather than to a fixed static network.

Betweenness in cognitive systems. Betweenness centrality has been applied to semantic networks, conceptual graphs, and mental lexicons in cognitive science. Studies of word association networks [13] find that high-betweenness words serve as conceptual “hubs” that mediate access to different semantic regions—precisely the role we argue betweenness-derived salience should play in Copycat’s workspace. The structural-importance interpretation of betweenness thus has empirical support from cognitive network science.

Information theory and cognition. The application of Shannon information theory to cognition has a distinguished history, from early cybernetics to modern Bayesian brain theories. The *free energy principle* [9] argues that biological systems minimize variational free energy (an upper bound on surprise), which is equivalent to minimizing the information-theoretic quantity $-\log p$. Our workspace entropy $H(\mathcal{W})$ is exactly a sum of such surprise terms. The conjecture that temperature tracks workspace entropy thus connects Copycat to contemporary theoretical neuroscience: analogy-making would be understood as surprise minimization, with temperature as the thermodynamic cost of structural uncertainty.

Small-world topology and clustering. Watts and Strogatz [16] showed that many natural networks exhibit high clustering combined with short average path lengths—the small-world property. High clustering arises from local triangle closure, which is precisely what our support vector measures. Our proposal that clustering-based support should drive bond building can be understood as a mechanism for workspace graphs to develop small-world topology over time:

bonds that close triangles are preferred, increasing global clustering while maintaining short paths through the high-betweenness central objects.

8.5 Limitations and Open Questions

The framework proposed here faces several genuine challenges.

Zero probabilities. When $C(b) = 0$, $I_C(b)$ is infinite. This occurs for isolated bonds with no triangulating neighbors—a common situation in early workspace states. The framework requires either regularization (a small $\epsilon > 0$ floor) or a convention that zero-probability events contribute a fixed large value. The information-theoretic grounding suggests a natural regularization: add a prior that each possible edge exists with a small baseline probability ϵ , giving $I = -\log_2(C(b) + \epsilon)$.

Independence assumption. The sum in (26) treats the three information sources as independent. In practice, dense bonds (high C) correlate with high betweenness for bridge objects and with dense groups. The true information content of the joint workspace state is lower than the sum of individual information contents, by the amount of mutual information among the three sources. Properly accounting for these correlations would require estimating joint probabilities, a significantly more complex computation.

Graph size dependence. Both betweenness centrality and subgraph density depend on graph size in ways that make direct comparison across workspaces of different sizes complicated. $\hat{B}(v)$ is normalized by $(n-1)(n-2)/2$, which changes as objects are grouped or split. $\rho(G_S)$ depends on $|S|$, which varies. A workspace with ten objects and one with five objects cannot be compared purely on the basis of raw metric values. The information-theoretic conversion partially addresses this—bits are a universal unit—but the probabilities themselves must be computed consistently as workspace size changes.

Direction of causality. We have proposed that temperature is proportional to workspace entropy. But Copycat’s temperature also directly influences codelet behavior—higher temperature leads to more random choices, which in turn affects how bonds are built and broken. The relationship between temperature and entropy may be bidirectional, with feedback loops that the simple proportionality in (27) cannot capture. A more complete model would treat temperature and entropy as jointly evolving state variables.

9 Conclusion

We have proposed replacing three hardcoded mechanisms in Copycat’s Workspace with principled graph-metric substitutes. The clustering coefficient of local neighborhoods defines a *support vector* for bonds, capturing structural embeddedness without arbitrary power-law formulas. Normalized betweenness centrality defines topology-driven *salience weights* for objects, capturing strategic position without fixed-ratio constants. Induced subgraph density defines a continuous *length factor* for groups, capturing internal connectivity without a four-value step function.

Beyond these substitutions, the paper has argued for a deeper theoretical point: all three metrics are ratios of actual to maximum possible connections, placing them in $[0, 1]$ and licensing their interpretation as probabilities. This observation enables the computation of Shannon self-information in bits for each metric, producing three quantities— I_C , I_B , I_ρ —measured in the same unit. Their sum defines a workspace entropy $H(\mathcal{W})$ that we conjecture to be monotonically related to Copycat’s temperature parameter.

The significance of the common unit cannot be overstated. The original three mechanisms were measured on incompatible scales: a power law, a pair of fixed ratios, and a four-value step function.

No meaningful arithmetic connects them. After the information-theoretic transformation, all three contribute additive terms to a single scalar—workspace entropy in bits—enabling comparison, combination, and potential replacement of the temperature update mechanism with a formally grounded alternative.

The proposals here are speculative in their stronger claims (particularly the temperature-entropy relationship), but they are grounded in established mathematics (graph theory, information theory, the maximum entropy principle) rather than in ad hoc intuition. Future work should implement the three metric substitutions, measure their effect on Copycat’s performance on benchmark problems, and test the empirical predictions of the information-theoretic framework. If the framework is confirmed, it would represent a significant step toward cognitive architectures whose internal dynamics are not merely metaphorically related to information theory, but formally and computationally grounded in it.

Relationship to Companion Paper

This paper develops in depth three specific proposals introduced in the companion paper [10], which surveyed graph-metric replacements for hardcoded constants across the full Copycat architecture (Slipnet, Workspace, and Coderack). The two papers are complementary: the companion paper provides breadth and a comprehensive catalog of substitutions; the present paper provides depth, formal analysis, and the novel information-theoretic unification that was not addressed in the broader survey. Readers are encouraged to consult both, as the Slipnet-side proposals (resistance distance for slippage, conceptual depth via graph distance) provide essential context for the full graph-theoretic reformulation program.

References

- [1] Ginestra Bianconi. Entropy of network ensembles. *Physical Review E*, 79(3):036114, 2009.
- [2] Stefano Boccaletti, Vito Latora, Yamir Moreno, Martin Chavez, and D.-U. Hwang. Complex networks: Structure and dynamics. *Physics Reports*, 424(4):175–308, 2006.
- [3] Ulrik Brandes. A faster algorithm for betweenness centrality. *Journal of Mathematical Sociology*, 25(2):163–177, 2001.
- [4] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley-Interscience, Hoboken, NJ, 2nd edition, 2006.
- [5] Matthias Dehmer and Abbe Mowshowitz. A history of graph entropy measures. *Information Sciences*, 181(1):57–78, 2011.
- [6] Paul Erdős and Alfréd Rényi. On random graphs. *Publicationes Mathematicae Debrecen*, 6:290–297, 1959.
- [7] Dedre Gentner. Structure-mapping: A theoretical framework for analogy. *Cognitive Science*, 7(2):155–170, 1983.
- [8] Douglas R. Hofstadter and FARG. *Fluid Concepts and Creative Analogies: Computer Models of the Fundamental Mechanisms of Thought*. Basic Books, New York, NY, 1995.
- [9] Edwin T. Jaynes. Information theory and statistical mechanics. *Physical Review*, 106(4):620–630, 1957.
- [10] Alex Linhares. From hardcoded heuristics to graph-theoretical constructs: A principled reformulation of the copycat architecture. *Manuscript*, 2025. Paper 1 in this series.

- [11] Melanie Mitchell. *Analogy-Making as Perception: A Computer Model*. MIT Press, Cambridge, MA, 1993.
- [12] Abbe Mowshowitz. Entropy and the complexity of graphs: I. an index of the relative complexity of a graph. *Bulletin of Mathematical Biophysics*, 30(1):175–204, 1968.
- [13] Mark E. J. Newman. *Networks*. Oxford University Press, Oxford, UK, 2nd edition, 2018.
- [14] Nicolas Rashevsky. Life, information theory, and topology. *Bulletin of Mathematical Biophysics*, 17(3):229–235, 1955.
- [15] Claude E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27(3):379–423, 1948.
- [16] Duncan J. Watts and Steven H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442, 1998.