Information Compression and Knowledge Graph Efficiency:
A Graph-Theoretic Framework for Cognitive Optimization

Gabriele Digilio

May 9, 2025

Abstract

The exponential growth of information in the digital era has precipitated a fundamental crisis in human capital development: the cognitive capacity of the learner remains biologically static while the complexity of knowledge domains scales incessantly. This paper proposes a unified theoretical framework that models knowledge systems as graph-based structures—where nodes represent concepts and edges represent semantic relationships—to solve for the optimization of **Cognitive Load** (encoding cost) against **Retrievability** (access cost). We rigorously compare disparate graph topologies, including dense cliques, sparse hierarchies, and scale-free networks, evaluating their efficacy through the lens of Information Theory and Cognitive Load Theory (CLT). By defining novel metrics such as **Von Neumann Graph Entropy**, **Normalized Information Distance**, and **Topological Redundancy**, we quantify the efficiency of knowledge representation. Furthermore, we employ agent-based simulations grounded in **Information Foraging Theory** and **Lévy Flight dynamics** to model the learner's navigation through these semantic spaces. This research establishes that "Navigable Small-World" networks, characterized by high local clustering and specific long-range connectivity distributions, offer the mathematically optimal structure for human learning. This theoretical groundwork serves as a precursor to applied adaptive learning engines, identifying the structural constraints that would later inform systems such as Risorgi.com.

1 Introduction: The Cognitive-Information Paradox

1.1 The Biological Constraint on Information Processing

The central challenge of the information age is not data availability, but rather the stark asymmetry between the volume of external information and the limited bandwidth of human cognition. The human cognitive architecture is defined by a severe bottleneck: **Working Memory (WM).** Classical research by Miller famously quantified this capacity as "seven plus or minus two" items, though contemporary studies by Cowan and others suggest a more restrictive limit of approximately four discrete "chunks" of information that can be actively processed simultaneously [1].

This limitation creates a paradox for complex domain mastery. To understand a system like "quantum mechanics" or "macroeconomics," a learner must integrate hundreds of thousands of discrete facts. However, they can only manipulate a handful at any given second. The only mechanism by which the brain overcomes this bottleneck is **Information Compression**—the process of recoding raw data into increasingly abstract, high-level schemas stored in **Long-Term Memory (LTM)** [2]. Once a schema is formed (e.g., the concept of "supply and demand"), it occupies only a single "slot" in working memory, yet it can be unpacked to reveal vast amounts of underlying data.

1.2 The Graph-Theoretic Turn in Cognitive Science

We posit that the efficiency of this compression process—and the subsequent ease of retrieving that information—is strictly determined by the **topological structure** in which the information is encoded. If knowledge is presented or stored as a disconnected list of facts (a graph with $E \approx 0$), the cognitive load is maximal because no compression is possible. If it is presented as a dense "hairball" of interconnections (a complete graph K_n), the load is equally high due to **element interactivity**—the overwhelming number of relationships that must be processed simultaneously [3, 4].

Therefore, learning can be mathematically modeled as a **Graph Rewriting Problem**: transforming a raw input graph G_{in} into a schematic mental graph G_{mental} such that the **Description Length** of G_{mental} is minimized (compression) while the **Navigability** (retrieval speed) is maximized. This paper explores the mathematical properties of G that facilitate this optimization.

1.3 Scope and Historical Context

This research provides the theoretical substrate for what would later become known as "adaptive learning ecosystems." While modern platforms utilize machine learning to dynamically serve content, this paper investigates the *a priori* structural properties that make knowledge learnable. We analyze the "physics" of information space—how the geometry of concepts affects the thermodynamics of thought.

We explicitly frame this research as a precursor to the developments seen in adaptive engines like Risorgi.com. The models discussed herein—specifically the use of graph entropy and small-world navigation to predict learning outcomes—provide the rigorous academic validation for the algorithmic approaches that such platforms would later operationalize. The focus here is on the theoretical bounds of efficiency: identifying the "ideal" knowledge graph structure that respects biological constraints.

2 Theoretical Framework: Bridging Cognition, Information, and Topology

To rigorously analyze knowledge graph efficiency, we must first bridge the gap between qualitative cognitive theories and quantitative graph metrics. We effectively translate "mental effort" into "bits" and "connectivity."

2.1 Cognitive Load Theory (CLT) as Graph Constraints

Cognitive Load Theory (CLT) provides the foundational psychology for our model. It distinguishes between three types of load, which we map directly to graph-theoretic properties [5, 6]:

2.1.1 Intrinsic Cognitive Load (ICL) and Element Interactivity

ICL is the inherent difficulty of the material. Sweller defines this through **Element Interactivity**: material is difficult if many elements must be processed *simultaneously* to be understood [4, 7].

- **Graph Mapping**: In a knowledge graph, element interactivity is a function of **local node degree** (k) and the **clustering coefficient** (C) of unlearned nodes. If a concept node v has edges to five other nodes $u_1 ... u_5$ that the learner has not yet mastered, the ICL is high. The learner must hold v and all u_i in working memory to resolve the relationship.

- **Metric implication**: An efficient graph minimizes the simultaneous exposure of un-chunked high-degree nodes.

2.1.2 Extraneous Cognitive Load (ECL) and Topological Noise

ECL is generated by the manner of presentation—information that does not contribute to schema acquisition but consumes mental resources [8, 9].

- **Graph Mapping**: This corresponds to **redundant edges** that convey no new semantic information (e.g., transitive links that can be inferred) or **poor layouts** that obscure the graph's hierarchical "backbone." In visualization terms, this is the "hairball" effect, where edge density exceeds the human visual processing capability, forcing the brain to expend energy filtering noise rather than learning structure [3].
- **Metric implication**: An efficient graph maximizes **sparsity** relative to reachability, eliminating edges that do not lower the average path length significantly.

2.1.3 Germane Cognitive Load (GCL) and Schema Formation

GCL is the "good" load devoted to processing information into schemas [5].

- **Graph Mapping**: This corresponds to the detection of **communities** or **modules** within the graph. When a learner identifies a dense cluster of nodes (e.g., "types of renewable energy") and encapsulates them into a single super-node, they are engaging in Germane processing.
- **Metric implication**: An efficient graph must exhibit high **modularity** (Q), possessing clear boundaries between conceptual clusters to facilitate this encapsulation.

2.2 The Minimum Description Length (MDL) Principle

The MDL principle is an information-theoretic formalism of Occam's Razor: the best explanation for a dataset is the one that permits the greatest compression [10, 11]. In our context, "learning" is the discovery of regularities that allow the knowledge graph to be compressed.

- **Kolmogorov Complexity**: The absolute minimum amount of information needed to describe an object [12]. While uncomputable, we approximate it using MDL.
- **Graph Compression**: A random graph requires listing every edge explicitly (high description length). A structured graph (like a star or grid) can be described by a simple rule (low description length).
- **Application**: We argue that the human brain attempts to minimize the description length of stored knowledge. Therefore, the most "efficient" knowledge graph is one that provides the highest fidelity of information with the lowest **Kolmogorov Complexity**. This aligns with recent findings in neural networks where compressibility correlates with generalization capability [13].

2.3 Knowledge Representation in Semantic Networks

We define a **Knowledge Graph (KG)** formally as $G = (V, E, \mathcal{T}, \mathcal{R})$, where:

- V is the set of vertices (concepts).
- E is the set of edges (relationships).

- $\mathcal{T} \subseteq V \times C$ assigns types to nodes (Ontology).
- $\mathcal{R} \subseteq E \times P$ assigns predicates to edges (Semantics) [14].

This semantic layering is critical. Unlike a pure interaction network (like Facebook), a KG encodes meaning. "Redundancy" in a KG is not just structural loops but **semantic equivalence**. If Node A ("Canine") and Node B ("Dog") have identical neighbor sets, they are semantically redundant. Identifying and merging such nodes is a primary mechanism of increasing efficiency [15].

3 Structural Typologies: The Topology of Knowledge

To optimize the KG, we must select the appropriate topology. We analyze four dominant structures found in natural and artificial systems.

3.1 Hierarchical Structures (Trees)

Hierarchies are the traditional mode of organizing knowledge (e.g., biological taxonomy, library Dewey Decimal systems).

- **Structure**: Directed Acyclic Graphs (DAGs) where every node has exactly one parent (except the root).
- **Cognitive Profile**: Hierarchies are optimized for **storage efficiency** and **categorization**. They reduce ICL by breaking complex topics into progressively smaller sub-topics (chunking) [16].
- **Navigational Deficit**: The "Tree" structure fails at **lateral retrieval**. To connect two "leaf" nodes in different branches (e.g., connecting "quantum tunneling" in Physics to "enzyme catalysis" in Biology), the learner must traverse all the way up to the root and back down. This results in a high **Retrieval Cost** for interdisciplinary or "far transfer" tasks [16].
- **Database Analogy**: Hierarchical databases excel at parent-child queries but struggle with many-to-many relationships, unlike relational or graph databases [16].

3.2 Scale-Free Networks (The "Hub" Model)

Scale-free networks, described by Barabási and Albert, follow a power-law degree distribution $P(k) \sim k^{-\gamma}$ (typically $2 < \gamma < 3$) [17].

- **Structure**: Most nodes have few connections; a few "hubs" have thousands.
- **Cognitive Profile**: This models the "Expert Mind." An expert has a few core concepts (hubs) that connect diverse domains.
- **Efficiency**:
  - **Retrieval**: Extremely fast. The **Average Path Length** scales as $L \sim \ln \ln N$, significantly faster than random networks ($L \sim \ln N$) [17, 18]. Hubs act as "wormholes" allowing rapid jumps between disparate ideas.
  - **Robustness**: The network is robust to the random loss of peripheral nodes (forgetting minor details) but catastrophic if a hub (core concept) is lost [17].
- **Knowledge Transfer**: Research indicates that scale-free networks provide the optimal pattern for knowledge diffusion, as hubs rapidly broadcast information to the entire system [18].

3.3 Small-World Networks (Watts-Strogatz)

Small-World networks occupy the "sweet spot" between regular lattices and random graphs.

- **Structure**: High **Clustering Coefficient** (C) (like a lattice) and short **Average Path Length** (L) (like a random graph) [19].
- **Mechanism**: They are formed by taking a regular lattice (highly clustered) and rewiring a small probability p of edges to random distant nodes (shortcuts).
- **Cognitive Profile**: This is the most biologically plausible model for semantic memory [19, 20]. The high clustering represents "topics" (dense patches of related concepts), while the shortcuts represent associations that link different topics (e.g., "Bank" connected to "River" and "Money").
- **Navigability**: Not all small-world networks are navigable. Kleinberg proved that for a decentralized agent (a learner) to find short paths, the long-range links must follow a specific distribution (d^{-2} in 2D space) [19, 21].

3.4 Dense vs. Sparse Graphs

- **Dense Graphs ($|E| \approx |V|^2$)**:
  - **Pros**: Direct access to any concept (low path length).
  - **Cons**: Massive redundancy and cognitive overload. The "Fan Effect" in psychology shows that as the number of facts associated with a concept increases, the retrieval time for any specific fact increases due to interference [22].
  - **Metric**: High **Graph Entropy** (high complexity/disorder).
- **Sparse Graphs ($|E| \approx |V|$)**:
  - **Pros**: Low storage cost, clear structure.
  - **Cons**: Fragile connectivity. If one link is forgotten, large sections of the graph may become inaccessible (graph partitioning).
  - **Optimization**: The goal is **Sparsification**—removing edges that provide low information gain (redundant paths) to reduce density while maintaining global reachability [23].

4 Quantifying Efficiency: The Metric Suite

To operationalize these comparisons, we define a suite of metrics. These metrics serve as the objective functions for the graph optimization algorithms used in adaptive learning.

4.1 Path Length and Global Efficiency

The most fundamental metric for retrieval is the geodesic distance between concepts.

- **Characteristic Path Length (L)**: The average number of steps to get from node i to node j.

$$L = \frac{1}{N(N-1)} \sum_{i \neq j} d_{ij}$$

- **Global Efficiency (E_{glob})**: Because L diverges for disconnected graphs, Efficiency is preferred:

$$E_{glob} = \frac{1}{N(N-1)} \sum_{i \neq j} \frac{1}{d_{ij}}$$

  In cognitive terms, E_{glob} measures the speed of parallel information diffusion across the memory network [24]. Sleep deprivation studies show a marked decrease in the E_{glob} of brain functional networks, directly correlating with reduced cognitive performance [24].

4.2 Graph Entropy (Von Neumann Entropy)

To measure the "complexity" or "information content" of the graph structure itself, we utilize **Von Neumann Entropy (S_{VN})**. Originally from quantum mechanics, this metric is defined over the spectrum of the graph's density matrix [25, 26, 27].

Let L be the graph Laplacian and ρ be the density matrix constructed by scaling L to have unit trace:

$$\rho = \frac{L}{\text{tr}(L)}$$

The Von Neumann entropy is:

$$S_{VN} = -\text{tr}(\rho \ln \rho) = -\sum_i \lambda_i \ln \lambda_i$$

where λ_i are the eigenvalues of ρ.

- **Interpretation**:
  - **Low Entropy**: The graph is highly regular (e.g., a simple ring or star). It has low structural information.
  - **High Entropy**: The graph is random or highly complex.
  - **Correlation with Load**: High graph entropy correlates with high complexity. However, very low entropy (total regularity) might imply a lack of rich associations. The "Entropic Brain Hypothesis" suggests that optimal cognition occurs at a "critical" level of entropy—high enough to allow flexible states, but low enough to maintain order [28, 29].
- **Redundancy Metric**: We can define **Topological Redundancy (R_{topo})** as the gap between the maximum possible entropy ($S_{max} = \ln N$) and the actual entropy:

$$R_{topo} = 1 - \frac{S_{VN}}{\ln N}$$

  High redundancy implies the graph has more connections than strictly necessary to encode the topology, which aids robustness (error correction) but increases encoding cost [30, 31].

4.3 Normalized Information Distance (NID)

While path length measures topological hops, we need a metric for **semantic distance**. How "far" is the concept "Dog" from "Wolf" vs. "Car"? We employ the **Normalized Information Distance**, approximated by the **Normalized Compression Distance (NCD)** [32, 33].

$$NID(x, y) = \frac{\max\{K(x|y), K(y|x)\}}{\max\{K(x), K(y)\}}$$

Approximated as:

$$NCD(x,y) = \frac{C(xy) - \text{min}\{C(x), C(y)\}}{\text{max}\{C(x), C(y)\}}$$

where $C(x)$ is the compressed size of concept x (e.g., using standard compression algorithms on textual descriptions).

Application: An efficient Knowledge Graph should create edges between nodes with low NID. If $NID(u,v)$ is low (they share much information), an edge (u,v) is "cheap" to learn. If NID is high, the edge represents a "far transfer" and incurs high intrinsic load.

4.4 Cyclomatic Complexity vs. Cognitive Complexity

Borrowed from software engineering, **Cyclomatic Complexity** measures the number of linearly independent paths through a graph [34].

$$M = E - N + 2P$$

Where E is edges, N is nodes, P is connected components.

- **Cognitive Interpretation**: A graph with high cyclomatic complexity offers many alternative ways to derive a fact. This is good for expert reasoning (robustness) but bad for novices (confusion/decision paralysis).
- **Refactoring**: "Cognitive Complexity" metrics attempt to correct Cyclomatic Complexity by weighting nested structures more heavily [35]. In KG design, this suggests that deeply nested, recursive relationships are harder to learn than linear chains, even if the node count is identical.

5 Simulating Learner Navigation: Agents in Semantic Space

Static metrics tell only half the story. Learning is a dynamic process of traversing the graph. We simulate this using agent-based models.

5.1 The Learner as an Information Forager

Information Foraging Theory (IFT) models users as organisms in an information environment, maximizing the rate of gain of valuable information per unit cost [36, 37].

- **The Patch Model**: Knowledge is clustered in "patches" (topics).
- **Marginal Value Theorem (MVT)**: The learner stays in a patch as long as the marginal rate of information gain ($g_P(t) = dV/dt$) exceeds the average rate of gain for the environment (R^*).

  $$g_P(t_{leave}) = R^*$$

  Once the rate drops (diminishing returns, i.e., "I'm not learning anything new about this topic"), the learner jumps to a new patch [38].

5.2 Lévy Flights in Memory Retrieval

Empirical studies on memory retrieval (e.g., the semantic fluency task: "list all animals you know") reveal that human retrieval patterns follow **Lévy Flights** [39, 40].

- **Pattern**: Bursts of rapid, short-distance retrievals (within a cluster, e.g., "lion, tiger, cheetah") separated by rare, long-distance jumps (between clusters, e.g., "...zebra. [pause]... whale, dolphin").
- **Distribution**: The probability of a jump of length l follows a heavy-tailed power law:

$$P(l) \sim l^{-\mu}$$

where $\mu \approx 2$ is optimal for searching sparse, random environments [40].
- **Simulation Implication**: A KG that does not support Lévy flights—i.e., a graph that is a uniform mesh without clusters—will frustrate the learner's natural retrieval mechanism. The graph *must* be clumpy (high clustering) to support the "bursts" and have shortcuts (small world) to support the "jumps."

5.3 Random Walks with Restart (RWR) vs. Spreading Activation

To simulate how a learner retrieves related concepts, we compare two algorithms:

5.3.1 Biased Random Walk with Restart (RWR)

This models a learner exploring a topic but occasionally "resetting" to the core concept to stay on track [41, 42]. The steady-state probability vector $\vec{p}$ is:

$$\vec{p} = (1 - c)\tilde{W}\vec{p} + c\vec{e}_q$$

Where $\tilde{W}$ is the normalized weight matrix, c is the restart probability, and $\vec{e}_q$ is the starting vector (the query or focus concept).

- **Outcome**: RWR effectively identifies the "contextual neighborhood" of a concept. It prevents the learner from drifting into irrelevant semantic territory (the "drift" problem in random walks).

5.3.2 Spreading Activation

This models neural activation spreading from a source node [43, 44].

$$A(t + 1) = A(t) \cdot W \cdot D$$

Where D is a decay factor.

- **Comparison**: While RWR is excellent for *ranking* relevance, Spreading Activation is better for simulating *priming*—how activating "Hospital" makes "Nurse" easier to retrieve even if not directly requested. For a learner, Spreading Activation predicts which concepts are "pre-loaded" in working memory.

5.4 Navigability and Kleinberg's Constant

Jon Kleinberg proved a fundamental limit on navigability. In a small-world network built on a d-dimensional lattice, decentralized greedy routing (where the agent only knows its neighbors and the target direction) is efficient ($O(\log^2 N)$) **if and only if** the probability of a long-range link decays as:

$$P(u \to v) \sim d(u, v)^{-\alpha}$$

where $\alpha = d$ (the dimensionality of the lattice) [19, 21].

- **Critical Insight**: If long-range links are too random ($\alpha = 0$), the graph has short paths, but the learner *cannot find them* because there is no correlation between geographic/semantic distance and connectivity. If α is too high, the graph lacks shortcuts.

- **Design Rule**: To make a KG "learnable," we cannot just add random links. We must add links that bridge semantic distances in a predictable way, matching the learner's intrinsic dimensionality of the topic space.

6 Information Compression and MDL

6.1 The MDL Objective Function

The Minimum Description Length (MDL) principle allows us to formalize the trade-off between model complexity and data fit. In graph terms, we want to find a summary graph (Schema) S that minimizes:

$$L(G, S) = L(S) + L(G|S)$$

- $L(S)$: The number of bits to encode the schema (the simplified graph, the "rules").

- $L(G|S)$: The number of bits to encode the "corrections" or "exceptions" required to reconstruct the full graph G from S [11, 45].

6.2 Sparsification Algorithms

To optimize a dense knowledge graph for a learner, we apply **MDL-based Sparsification** [23, 46].

1. **Identify Communities**: Use modularity optimization to find dense clusters.

2. **Abstraction**: Collapse each cluster into a super-node (reducing N).

3. **Prune Redundancy**: Remove edges that are explained by the structure. For example, if the graph is a hierarchy, and we know $A \to B$ and $B \to C$, an explicit edge $A \to C$ is redundant if the learner understands the rule of transitivity. Removing $A \to C$ reduces $L(G)$ without losing information.

4. **Preserve Bridges**: MDL dictates we *must* keep edges that connect different communities, as these cannot be predicted by local rules (high "surprise" value).

6.3 Semantic Compression

We can also compress the graph semantically. Using **Formal Concept Analysis (FCA)**, we can group nodes that share identical attribute sets into a concept lattice, effectively reducing the redundancy of repeated properties [15]. This moves the graph from an "Instance Graph" (raw data) to an "Ontology Graph" (semantic rules), which is far more efficient for storage in LTM [14].

7 Comparative Analysis Table

The following table summarizes the performance of different graph topologies across the defined cognitive and computational metrics.

Analysis:

- **Dense graphs** fail due to Cognitive Load (Element Interactivity).

Metric	Dense (Clique)	Sparse (Lattice)	Hierarchical (Tree)	Scale-Free (Hub)	Navigable Small-World
Avg. Path Length (L)	1 (Opt)	$\sim \sqrt{N}$ (Poor)	$\sim \log N$	$\sim \ln \ln N$ (Exc)	$\sim \log N$ (Good)
Clustering Coeff. (C)	1	High	0	Low	**High**
Element Interactivity	V. High	Low	Low	Mixed	**Balanced**
Navigability	Instant	Slow	Rigid	Fast	**Optimal**
Graph Entropy	Low	Low	Low	Medium	**Critical State**
MDL (Compression)	High	Medium	Low	Low	**Optimal**
Robustness	High	Low	Low	High	**High**

Table 1: Comparison of Graph Topologies on Cognitive Metrics

- **Hierarchies** fail due to Retrieval Cost (no lateral movement).
- **Scale-Free networks** are excellent for flow but the "Hubs" can become bottlenecks of Intrinsic Load (a concept connected to 1000 others is hard to learn).
- **Navigable Small-World networks** emerge as the winner. They maintain the local clustering needed for "Patch" foraging (lowering load via coherence) while providing the specific long-range links needed for Lévy flight retrieval (lowering access cost).

8 Simulation Results: The Learner Agent

We simulated a "Learner Agent" traversing these topologies.

- **Setup**: Agent starts at random node S. Target is node T. Agent has limited "energy" (attention). Agent uses a greedy routing strategy based on semantic similarity (edge weights).
- **Results**:
  - **In Random Graphs**: The agent engages in "thrashing." Without structural cues (clustering), semantic similarity is a poor guide. The agent visits many irrelevant nodes. Cost is high.
  - **In Hierarchies**: The agent frequently travels up to the root and back down. While successful, the path length is often $2 \times$ depth, which is inefficient for related "cousin" nodes.
  - **In Small-World Networks**: The agent utilizes local clusters to build "momentum" (spreading activation) and uses long-range links to jump closer to the target. This topology consistently yielded the highest success rate per unit of energy expended.

9 Implications and Future Directions

9.1 The "Pre-Risorgi" Landscape

This theoretical analysis predates the implementation of adaptive engines such as **Risorgi.com**. While Risorgi and similar platforms operationalize these metrics—dynamically adjusting content graphs based on user performance—this paper establishes the *mathematical inevitability* of those design choices. The "Navigable Small-World" is not just a convenient data structure; it is the boundary condition for efficient human learning.

9.2 Parallel with Retrieval-Augmented Generation (RAG)

In modern AI, we see a parallel tension in **GraphRAG** systems. Pure vector retrieval (Dense) is fast but lacks structure (high entropy, hallucination risk). Graph-based retrieval (Sparse) is structured but computationally expensive [47]. The solution in AI mirrors the solution in cognition: a hybrid, hierarchical graph that uses vector embeddings for local scent and explicit edges for global structure [48].

9.3 Conclusion

We conclude that the efficiency of a knowledge system is maximizing the ratio of **Semantic Reachability** to **Structural Entropy**. The optimal topology is a **Hierarchical Small-World Network** that has been compressed via **MDL** principles to remove non-informative redundancy while preserving the long-range links required for **Lévy flight** navigation.

This structure minimizes Intrinsic Load by controlling Element Interactivity through clustering, minimizes Extraneous Load by reducing topological noise, and maximizes Germane Load by making the community structure (schema) explicit. Future instructional design must move beyond static lists and trees to engineer these ”living” graph topologies that resonate with the fundamental physics of the human mind.

10 References

1, 4, 5, 6 **Cognitive Load Theory:** Working memory limits, Element Interactivity, Types of Load.

19, 21, 49 **Network Theory:** Small-World Networks, Decentralized Search, Kleinberg's Navigability.

10, 11, 12, 13 **Information Theory:** Minimum Description Length (MDL), Kolmogorov Complexity, Compression.

25, 26, 27, 28 **Graph Entropy:** Von Neumann Entropy, Spectral Graph Theory, Brain Entropy.

36, 50, 51 **Information Foraging:** Optimal Foraging Theory, Patch Models.

39, 40 **Lévy Flights:** Human memory retrieval patterns, Heavy-tailed distributions.

17, 18 **Scale-Free Networks:** Hubs, Power laws, Knowledge transfer efficiency.

15, 16 **Hierarchies & Ontologies:** Tree structures, Semantic layering, FCA.

32, 33 **Distance Metrics:** Normalized Information Distance (NID/NCD).

47, 48 **Modern Applications:** GraphRAG, Vector vs. Graph retrieval.

24, 52 **Neuroscience:** Brain functional networks, Global efficiency.