Appendix A Appendix of Networks Multiscale Entropy Analysis

A.1 Details of Compression-Based Entropy Estimation

Compression-based entropy estimation consists of two stages: encoding and compression. Each graph is transformed into a binary sequence representing its adjacency structure in the encoding phase. We follow the method proposed by Choi et al. [10], which encodes the presence or absence of edges into a compact binary string. The encoding is designed to be isomorphism-invariant, ensuring consistent entropy estimates across differently labeled but structurally identical graphs.

In the compression phase, we apply arithmetic coding, a lossless data compression technique that represents the binary sequence as a sub-interval of the unit interval $[0, 1)$. The resulting compressed representation's size reflects the graph's redundancy and structural regularity. More structured graphs yield shorter encoded lengths, while more random graphs result in longer bit-strings. The final length $L(G)$ of the compressed binary representation serves as an estimate of the graph's structural entropy. This method captures local and global patterns, robustly quantifying complexity.

The process was carried out as follows:

1. **First Phase – SZIP Compression:** We implemented the SZIP algorithm [10], which encodes a labeled graph $G = (V, E)$ into two binary sequences, B_1 and B_2. The procedure is iterative:
   - At each step, a vertex v is removed from the current partition of V.
   - For each subset of remaining vertices, we encode the number of neighbors of v using $\lceil \log(|U| + 1) \rceil$ bits. These multi-bit encodings are appended to B_1.
   - When the subset is a singleton ($|U| = 1$), we record a single bit, appended to B_2.
   - The partition is refined according to adjacency: subsets split into neighbors and non-neighbors of v. This process continues until all vertices are removed.

   The result is a pair (B_1, B_2) that preserves the complete topological structure of the graph while remaining independent of vertex labeling.
2. **Second Phase – Encoding of Binary Sequences:** The sequences B_1 and B_2 are then compressed using *Arithmetic Coding* [35]:
   - Arithmetic coding represents an entire sequence as a single number in $[0, 1)$, providing near-optimal compression.
   - The compressed length $L(G)$ closely approximates the *structural entropy* of the graph.
   - Complexity interpretation:

The resulting compression length $L(G)$ is a measure of the graph's structural complexity, providing a quantitative metric for assessing the regularity or randomness of its structure. It provides an interpretable measure of complexity, where more complex or random structures yield longer encodings and regular or patterned structures allow for shorter, more compressed representations.

A.2 Details of Link Prediction Method

The third component of our methodology aims to quantify the structural predictability of a network using link prediction algorithms. To achieve this, we implement a leave-one-out strategy that evaluates the model's ability to recover existing edges from the graph.

The procedure is as follows:

1. For each existing edge e_i in the graph:
   (a) Temporarily remove the edge e_i from the graph.
   (b) Compute a similarity score for all non-connected node pairs (including e_i) using two commonly used link prediction metrics:
   - **Jaccard Coefficient:** Measures the similarity between two nodes u and v based on their neighbor sets:

     $$\text{Jaccard}(u, v) = \frac{|\Gamma(u) \cap \Gamma(v)|}{|\Gamma(u) \cup \Gamma(v)|}$$

     where $\Gamma(u)$ denotes the set of neighbors of node u.
   - **Adamic-Adar Index:** Assigns higher weights to shared neighbors with fewer connections:

     $$\text{Adamic-Adar}(u, v) = \sum_{w \in \Gamma(u) \cap \Gamma(v)} \frac{1}{\log |\Gamma(w)|}$$

   (c) Rank all candidate non-edges by their score in descending order.
   (d) Record the rank r_i assigned to the removed edge e_i in this list.
2. After processing all edges, construct a ranking sequence $D = \{r_1, r_2, \ldots, r_E\}$, where E is the total number of edges in the original graph.
3. To compute the prediction entropy H, divide the full ranking range $[1, \frac{N(N-1)}{2} - \frac{\langle k \rangle N}{2} + 1]$ into equally sized bins, where N is the number of nodes and $\langle k \rangle$ is the graph's average degree.
4. Finally, calculate the entropy as:

   $$H = -\sum_{j=1}^{N/2} p_j \log p_j$$

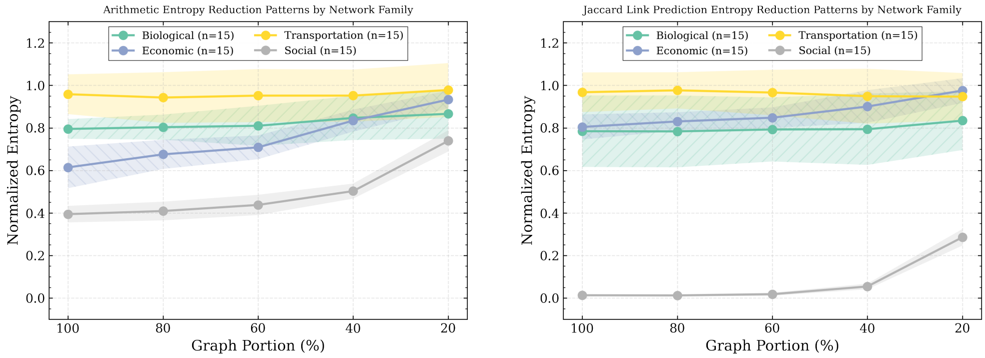
   where p_j denotes the probability that a rank r_i falls within bin j.

This entropy value H serves as a measure of the network's structural predictability: low entropy indicates that removed edges are consistently ranked near the top (i.e., highly predictable). In contrast, high entropy reflects a more random and unpredictable structure.

A.3 Regression Analysis

To complement the analysis presented in the main text using the Adamic-Adar link prediction method, Figure A1 displays entropy trajectories based on the Jaccard link prediction index. The plots show average entropy values across successive levels of node reduction for four network domains: biological, social, transportation, and economic.

As in the Adamic-Adar case, we observe a strong alignment between structural (compression-based) entropy and link prediction entropy derived from Jaccard scores. Notably, social networks again display consistently lower entropy across both measures, highlighting their high structural regularity and predictability. Meanwhile, biological and economic networks show greater heterogeneity, indicating more complex and less redundant topologies. These patterns reinforce the robustness of the observed multiscale entropy-link prediction relationship across distinct similarity heuristics.



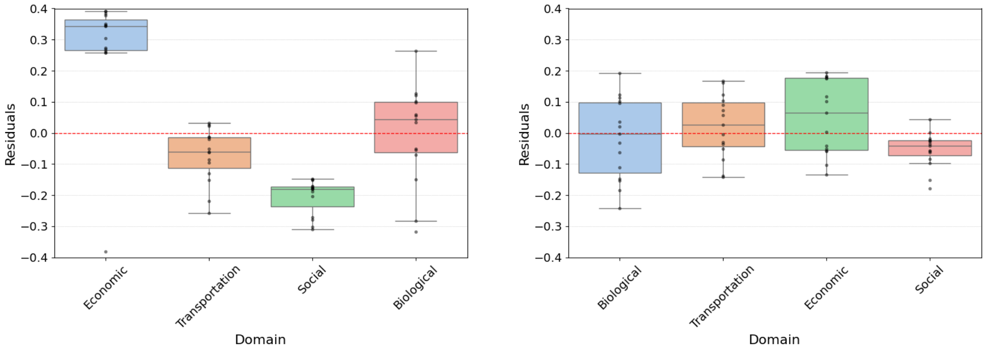
(a) Multiscale Entropy

(b) Link Prediction Entropy (Jaccard)

Fig. A1: Average entropy trajectories under multiscale reduction for four network families using Multiscale Entropy and Jaccard Link Prediction Entropy.

To further examine model performance, we analyzed the residuals of the regression predictions for normalized Adamic-Adar entropy by network domain. Figure A2 shows boxplots and individual residual points for each domain under Model 1 (single-scale) and Model 5 (multiscale).

In Model 1, residuals are notably dispersed across domains, with economic and social networks showing a high spread of residuals. In contrast, Model 5 demonstrates significantly reduced residual spread across all domains. This suggests that multiscale entropy inputs provide not only more accurate but also more consistent predictive performance across heterogeneous network structures. These residual patterns corroborate the results of the regression models: incorporating multiscale entropy reduces both bias and variance in link prediction entropy estimation.



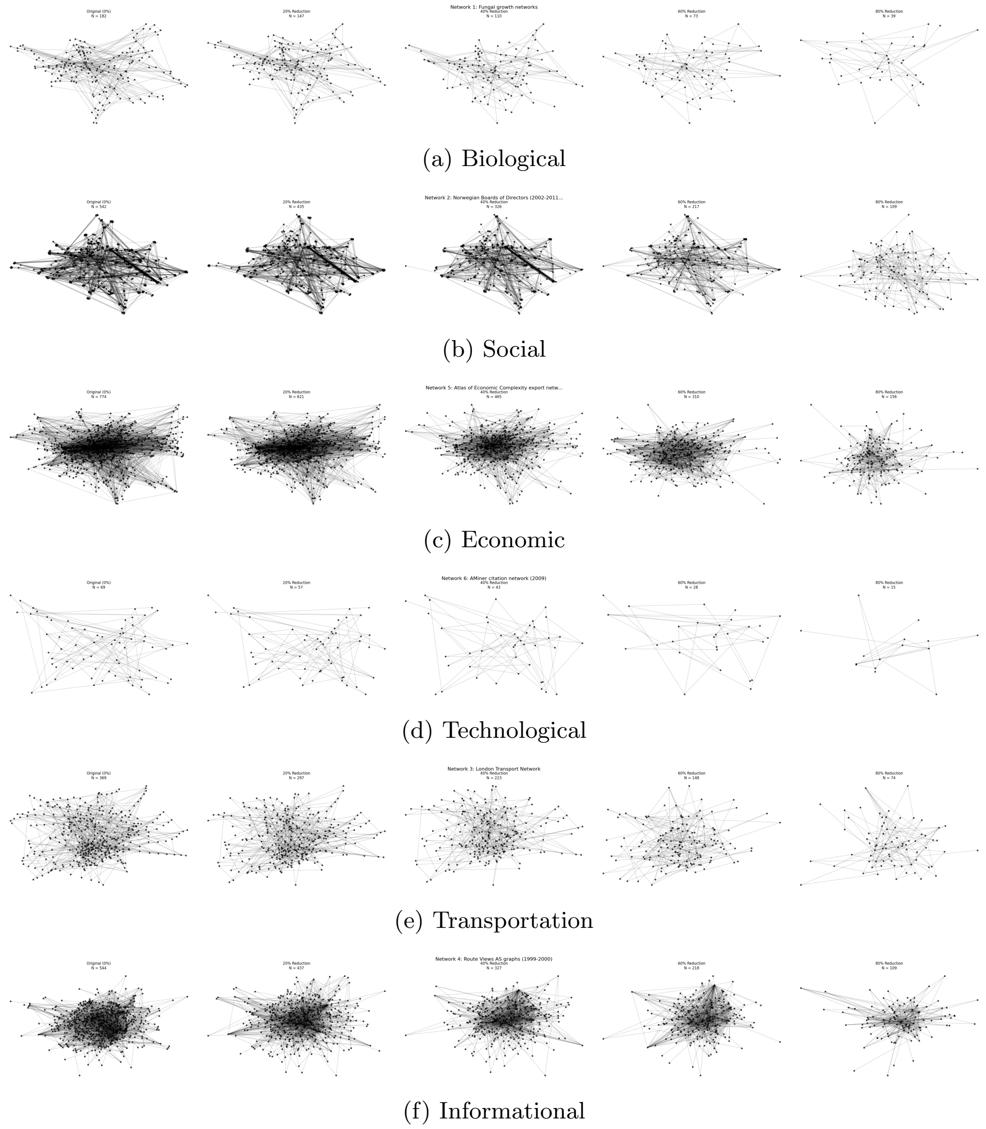
(a) Residuals by domain (Model 1)

(b) Residuals by domain (Model 5)

Fig. A2: Distribution of residuals for predicted Adamic-Adar entropy across four network domains. Multiscale regression (Model 5) substantially reduces residual variance compared to the single-scale model (Model 1).

A.4 Illustrations of Real Networks Across Scales

This appendix presents representative *real* networks from each domain (Biological, Social, Economic, Technological, Transportation, Informational), visualized at five reduction levels: 100%, 80%, 60%, 40%, and 20% of nodes. These examples complement the multiscale entropy analysis by highlighting how characteristic structures evolve under spectral coarsening.



(a) Biological

(b) Social

(c) Economic

(d) Technological

(e) Transportation

(f) Informational

Fig. A3: Representative networks from six domains, shown across multiple reduction scales (100%, 80%, 60%, 40%, 20%).