



PAPER

OPEN ACCESS

RECEIVED
31 July 2025REVISED
12 November 2025ACCEPTED FOR PUBLICATION
8 January 2026PUBLISHED
23 January 2026

Original content from this work may be used under the terms of the [Creative Commons Attribution 4.0 licence](#).

Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.



Mapping memory-biased dynamics with compact models reveals overlapping communities in large networks

Maja Lindström^{1,2,3,9,*} , Rohit Sahasrabudhe^{4,5,9} , Anton Holmgren² , Christopher Blöcker^{2,6,7} , Daniel Edler^{2,8} and Martin Rosvall^{2,3} ¹ Department of Computing Science, MIT-huset, Umeå University, SE-901 87 Umeå, Sweden² Integrated Science Lab, Department of Physics, Umeå University, SE-901 87 Umeå, Sweden³ Siftlab AB, Döbelngatan 12, SE-113 58 Stockholm, Sweden⁴ Mathematical Institute, University of Oxford, Oxford, United Kingdom⁵ Institute for New Economic Thinking, University of Oxford, Oxford, United Kingdom⁶ Data Analytics Group, Department of Informatics, University of Zurich, CH-8006 Zurich, Switzerland⁷ Chair of Machine Learning for Complex Networks, Center for Artificial Intelligence and Data Science (CAIDAS), University of Würzburg, DE-97070 Würzburg, Germany⁸ Gothenburg Global Biodiversity Centre, University of Gothenburg, Gothenburg, Sweden⁹ These authors contributed equally to this work.

* Author to whom any correspondence should be addressed.

E-mail: maja.lindstrom@umu.se and rohit.sahasrabudhe@maths.ox.ac.uk**Keywords:** overlapping communities, higher-order networks, flow-based community detection, random walks, map equation, Infomap, information theory

Abstract

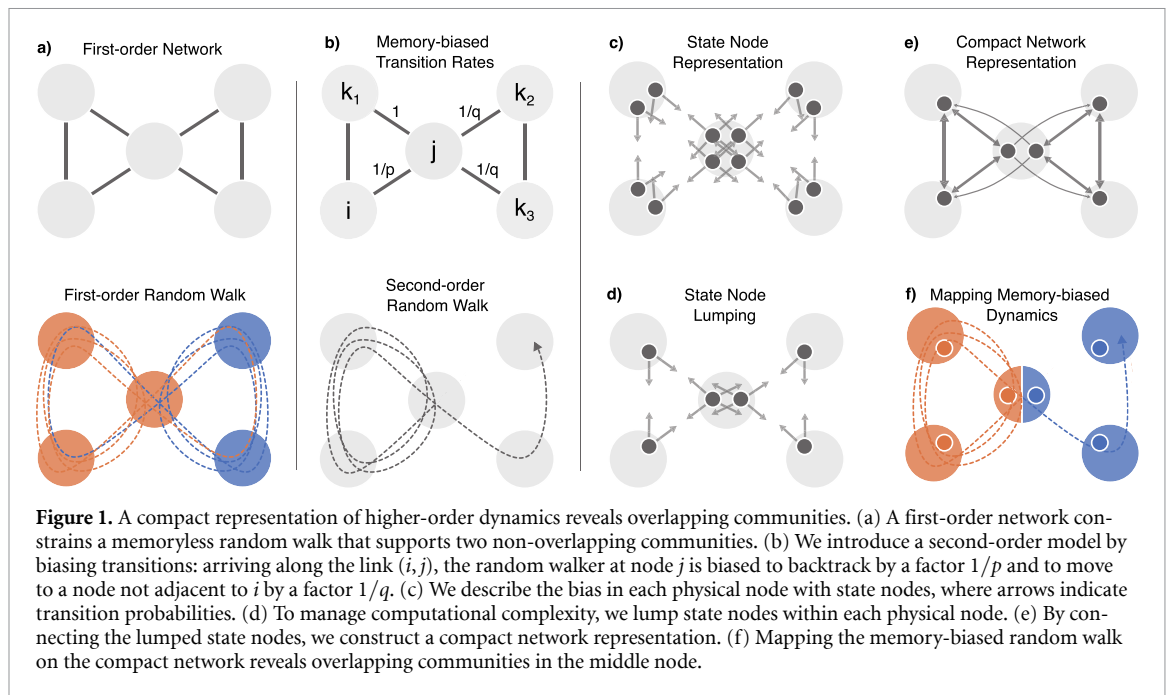
Many real-world systems, from social networks to protein-protein interactions and species distributions, exhibit overlapping flow-based communities that reflect their functional organisation. However, reliably identifying such overlapping flow-based communities requires higher-order relational data, which are often unavailable. To address this challenge, we capitalise on the flow model underpinning the representation-learning algorithm *node2vec* and model higher-order flows through memory-biased random walks on first-order networks. Instead of simulating these walks, we model their higher-order dynamic constraints with compact models and control model complexity with an information-theoretic approach. Using the map equation framework, we identify overlapping modules in the resulting higher-order networks. Our compact-model approach proves robust across synthetic benchmark networks, reveals interpretable overlapping communities in empirical networks, and scales to large networks.

1. Introduction

Network-theoretic tools offer a powerful framework for modelling, analysing, and understanding the organisation and function of complex systems [1, 2]. These systems often consist of tightly knit modules, or communities, that represent distinct functional parts [3]. In many networks, these communities overlap: nodes may belong to more than one community, with overlaps ranging from rare exceptions to extensive intersections [4].

Popular methods for detecting overlapping communities rely on topological heuristics, such as clique percolation [5], local optimisation of order statistics [6], and link communities [7], recently extended to hypergraphs [8]. But many real-world networks, particularly those with weighted or directed links, represent more than local structure; they impose constraints on network flows that connect non-adjacent nodes [9–13].

Mapping such flows naturally reveals overlapping communities. For example, citation paths traverse through overlapping research fields in scholarly publishing [14], global maritime traffic move through overlapping clusters of international ports [15], and information circulate within overlapping groups of friends and colleagues in social networks [16]. These flow patterns define the system's organisation: groups of nodes that trap flows form overlapping flow-based communities. When higher-order relational



data are available, such overlapping communities can be identified directly without relying on topological heuristics [14–18].

Without higher-order network data, flow-based methods impose hard boundaries or minimal overlap [19], misrepresenting the system’s organisation. For example, applying a memoryless Markov model to a journal citation network assigns a multidisciplinary journal like Nature to a single research field, neglecting its broader role [20]. Since higher-order data are often unavailable, this limitation raises a critical question: How can we identify overlapping flow-based communities from link data alone?

To reveal overlapping communities, we approximate higher-order dynamics on first-order networks using memory-biased random walks. In the spirit of memory models with tuned rates for return, triangular, and exploratory steps [14], we focus on a second-order model inspired by the representation learning algorithm *node2vec* [21]. A random walker at node j remembers the previously visited node i and selects the next node k based on the shortest-path distance between i and k (figure 1). By weighting transitions by distance, we can adjust the walker’s exploration radius to favour local or global flow patterns.

Bypassing explicit random walk simulations, we model their constraints using memory networks [17], where *physical nodes* mirror the original nodes in the first-order network and *state nodes* track transitions that depend on the previous step. To manage the combinatorial explosion of second-order paths, we build a compact representation [16, 20] that retains essential flow dynamics while cutting model complexity. We then apply the map equation framework [9, 22, 23] on the compact higher-order networks to detect overlapping communities (figure 1).

The proposed method proves robust across synthetic benchmarks and effectively identifies interpretable overlapping flow-based communities in real-world networks. The compact higher-order models scale to large networks while retaining the prominent flow constraints of memory-biased random walks.

2. Modelling memory-biased random walks

The simplest model of flow on a network captures movement constrained solely by the out-links. In an unbiased first-order random walk, a walker at node j transitions to node k with rate

$$\pi_{jk} = \frac{w_{jk}}{\sum_{k'} w_{jk'}}, \quad (1)$$

where w_{jk} is the weight of the link $j \rightarrow k$. The transitions depend only on the current position in this memoryless model.

To model memory effects, second-order random walks incorporate the previous step. The transition from j to k depends not only on j but also on the node i from which the walker arrived. The *node2vec* walk [21] captures such memory by biasing the first-order transition rate depending on whether i and k

are neighbours,

$$B(i, k) = \begin{cases} 1/p & \text{if } i = k, \\ 1 & \text{if } i \text{ and } k \text{ are neighbours,} \\ 1/q & \text{otherwise.} \end{cases} \quad (2)$$

The parameters p and q control the rate of diffusion: p influences the likelihood of going back to the previous node, while q affects the tendency to move to a node not neighbouring i .

2.1. Full second-order representation

To fully capture the *node2vec* dynamics, we construct the weighted and directed second-order network M_2 . For every physical node j with in-degree d_j^{in} , we create state nodes j_i for each in-link $i \rightarrow j$, representing arrivals at j from i . The link weight between state nodes j_i and k_j is the physical link weight w_{jk} divided between the state nodes j_i to preserve the physical network and multiplied by the biased transition rates

$$w_{j_i k_j} = \frac{w_{jk}}{d_j^{\text{in}}} B(i, k). \quad (3)$$

For physical nodes without in-links, we create a single state node and use the first-order transition rates. With this state node network representation, the map equation framework [17] can reveal modular structures in the higher-order dynamics. To compress the network dynamics, Infomap can split a physical node's state nodes across overlapping modules.

The second-order model of a physical node j with in- and out-degrees d_j^{in} and d_j^{out} , has d_j^{in} state nodes, each with d_j^{out} out-links. For a network with n nodes and m links, M_2 has m nodes, or $2m$ if undirected, and $\sum_{j=1}^n d_j^{\text{in}} d_j^{\text{out}}$ links, making analysis computationally demanding for large networks.

2.2. Compact second-order representation

To efficiently analyse large networks, we lump state nodes into a compact second-order model that preserves key flow dynamics. For a partition \mathcal{P}^j of physical node j 's state nodes, we create lumped state nodes j_α for each subset $\alpha \in \mathcal{P}^j$. Each lumped node j_α represents arrivals at j from any node $i \in \alpha$. To preserve flow dynamics when aggregating links, we weight them according to arrival frequencies. The link weight between lumped nodes j_α and k_β is

$$w_{j_\alpha k_\beta} = \sum_{i \in \alpha} \frac{p_{j_i}}{p_{j_\alpha}} d_j^{\text{in}} w_{j_i k_j}, \quad (4)$$

where p_{j_i} is the PageRank flow of state node j_i , and $p_{j_\alpha} = \sum_{i \in \alpha} p_{j_i}$ is the aggregated flow for the lumped node j_α .

To measure the information loss by replacing M_2 with the compact model M_c , we use the Jensen–Shannon divergence with the Shannon entropy H measured in bits. For physical node j , the Jensen–Shannon divergence is

$$\text{JSD}(M_c^j, M_2^j) = \overbrace{\sum_{\alpha \in \mathcal{P}^j} p_{j_\alpha} H(\pi_{j_\alpha})}^{M_c^j} - \overbrace{\sum_i p_{j_i} H(\pi_{j_i})}^{M_2^j}, \quad (5)$$

where $\pi_{j_i} = w_{j_i \cdot} / \sum_{k'} w_{j_i k'}$ is the transition rates from state node j_i and

$$\pi_{j_\alpha} = \sum_{i \in \alpha} \frac{p_{j_i}}{p_{j_\alpha}} \pi_{j_i} \quad (6)$$

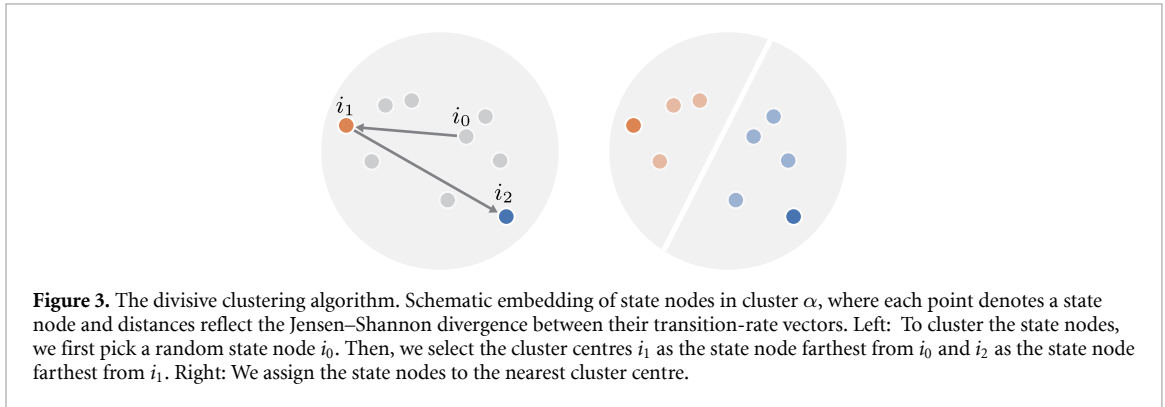
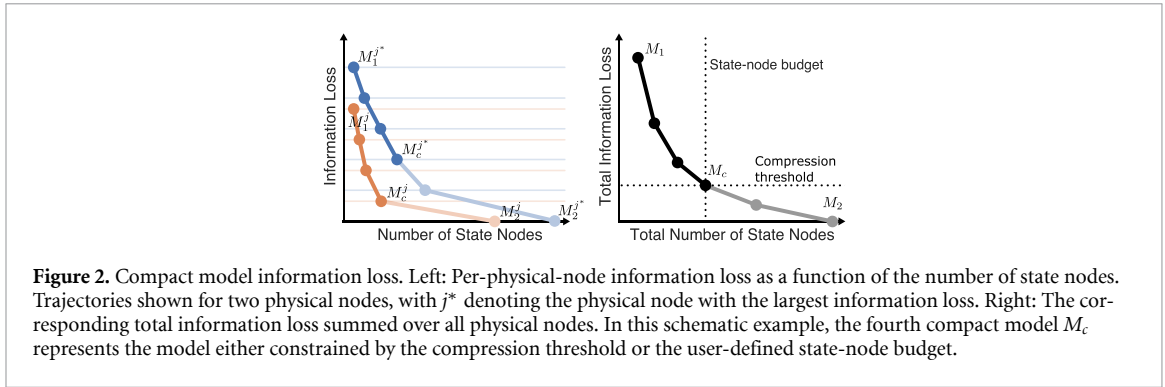
the flow-weighted lumped transition rates.

For the entire network,

$$\text{JSD}(M_c, M_2) = \sum_j \text{JSD}(M_c^j, M_2^j), \quad (7)$$

which is the difference in entropy rates between the compact and M_2 model. When the random walk is unbiased ($p = q = 1$), all transition-rate vectors π_{j_i} at each physical node coincide so that $\text{JSD}(M_c, M_2) = 0$.

We interpolate between a first-order model M_1 and M_2 by iteratively dividing the compact node with the highest $\text{JSD}(M_c, M_2)$ in two (figure 2) using a divisive clustering algorithm.



2.2.1. Divisive clustering algorithm

The divisive clustering algorithm splits lumped state nodes in each physical node to minimise information loss. Given parameters p and q , we assemble the compact network model by

- (i) creating lumped state nodes independently for each physical node, and
- (ii) adding links between lumped state nodes $j_\alpha \rightarrow k_\beta$ for $j \in \beta$ with weight $w_{j_\alpha k_\beta}$ (equation (4)).

We propose a fast divisive clustering algorithm that partitions state nodes in each physical node to minimise information loss. From equation (5), the Jensen–Shannon divergence of cluster α of physical node j is

$$p_{j_\alpha} H(\pi_{j_\alpha}) - \sum_{i \in \alpha} p_{j_i} H(\pi_{j_i}). \quad (8)$$

Starting from the first-order model, a single cluster containing all state nodes of a physical node, we iteratively split the cluster with the highest divergence into two clusters with similar state nodes. We continue until the total compression $\Delta_{\text{JSD}} = 1 - \text{JSD}(M_c, M_2) / \text{JSD}(M_1, M_2)$ exceeds a compression threshold Δ_{MIN} , default 90 percent, or we reach an optional state node budget, a limit on the number of state nodes. To split cluster α , we:

- (i) pick a random seed state node $i_0 \in \alpha$;
- (ii) identify two cluster centres by Jensen–Shannon distance, $i_1 \in \alpha$ farthest away from i_0 and $i_2 \in \alpha$ farthest away from i_1 ;
- (iii) assign each state node in α to its nearest centre (figure 3).

Because this procedure is fast and stochastic, we repeat these steps several times and retain the division with the lowest divergence.

Each iteration scales linearly with cluster size. A physical node with degree d has d state nodes. Identifying cluster centres and assigning state nodes requires $\mathcal{O}(d)$ operations. Splitting each node at most d times across all n nodes gives $\mathcal{O}(nd^2) = \mathcal{O}(md)$ complexity. The algorithm thus scales linearly with the number of links for bounded average degree.

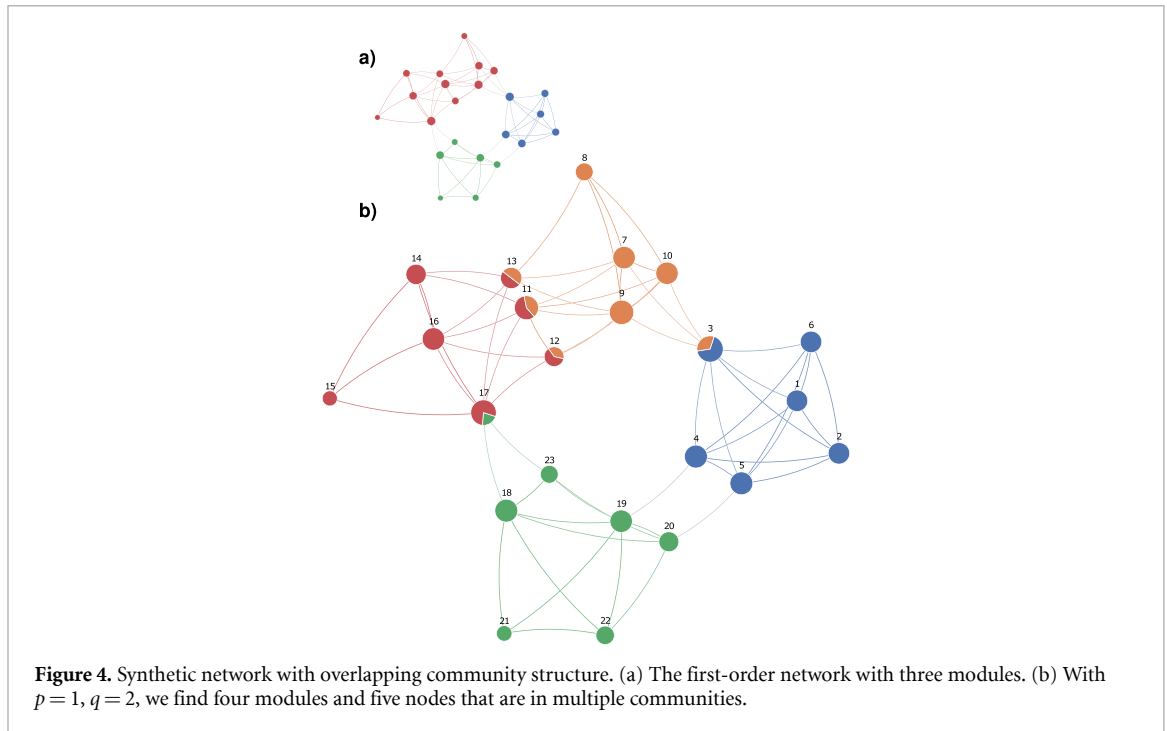


Figure 4. Synthetic network with overlapping community structure. (a) The first-order network with three modules. (b) With $p = 1$, $q = 2$, we find four modules and five nodes that are in multiple communities.

3. Mapping memory-biased random walks

To detect communities, we apply Infomap [23], which searches for a network partition \mathbf{M} that minimises the expected codeword length $L(\mathbf{M})$ for a modular description of a random walk on the network [9]. For each network and set of parameters, we run 20 independent Infomap optimisation trials and select the one with the lowest codeword length. For clarity in illustrations, we constrain the search to two-level partitions, though the proposed approach naturally extends to hierarchical partitions.

Our evaluation strategy reflects a conceptual difference in how we represent overlapping communities. Most overlapping community-detection algorithms assign nodes to communities with fuzzy membership weights, producing a single partition of the node set with probabilistic assignments. Our state-node approach produces a different object: it identifies which flow paths through a node participate in which communities. A node visited from different predecessors can play distinct roles in different communities. This representation captures constraints on higher-order flows that fuzzy membership cannot express. We therefore focus our evaluation on internal validation. First, how well do compact models approximate complete higher-order dynamics? Second, do detected communities reflect interpretable structures in real-world networks? Comparisons with established methods serve to illustrate sensible behaviour, not to rank performance on standardised metrics.

3.1. Synthetic networks

We test our method on a schematic network to illustrate detected overlapping communities and on benchmark networks to evaluate how well compact models approximate full second-order models.

3.1.1. Schematic network

To showcase the full second-order model, we apply it to the hand-crafted schematic network designed by Palla *et al* [5] to illustrate overlapping communities (figure 4). We set $p = 1$ and $q = 2$ and identify the previously reported overlapping communities and, unlike the original study, also detect a small overlap at node 17.

3.1.2. LFR benchmark networks

To assess the accuracy, we construct instances of the LFR benchmark networks with planted overlapping communities [24] using $N = 100$ nodes, average node degree $k = 6$, clustering coefficient $C = 0.3$, min module size $c_{\min} = 10$, max module size $c_{\max} = 20$, max degree $k_{\max} = 15$, and mixing parameter $\mu = 0.2$.

We vary the number of overlapping nodes $o_n \in \{5, 15\}$, and the number of memberships of the overlapping nodes $o_m \in \{2, 4\}$. To assess the effect to the bias parameters, we use $p \in \{0.5, 1\}$ and $q \in \{2, 4\}$. To compute average performance, we generate 20 network instances for each parameter combination.

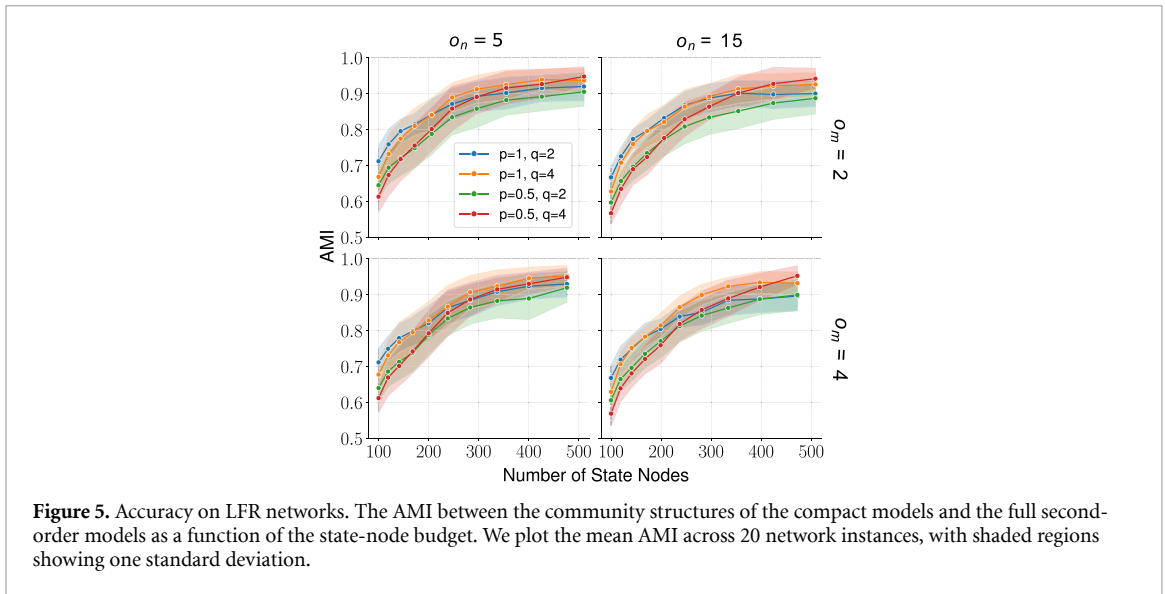


Figure 5. Accuracy on LFR networks. The AMI between the community structures of the compact models and the full second-order models as a function of the state-node budget. We plot the mean AMI across 20 network instances, with shaded regions showing one standard deviation.

For each network instance, we create 10 compact models M_c with different state node budgets. We run Infomap 20 times on each compact model to find modules that retain the memory-biased flow, which may differ from those planted in the LFR framework.

To measure the accuracy, we compare the community structure of M_c to that of the full second-order model M_2 . Across different values of p , q , o_n , and o_m , the adjusted mutual information (AMI) remains high (figure 5). Because we compare with one particular M_2 solution, we do not expect the AMI to reach its maximum value of one. The solution landscape is flat: different partitions can have similar codelengths yet look quite different [25]. When we evaluate compact models with few state nodes, we compare their AMI against that of larger models. As the number of state nodes increases, the AMI rises sharply at first, showing that even relatively small compact models capture important effects of the bias. When M_c contains about half as many state nodes as M_2 , the AMI stabilises near 0.9, demonstrating that we can closely match the community structure of the full second-order network with a much smaller model.

3.1.3. ABCD+ o^2 benchmark networks

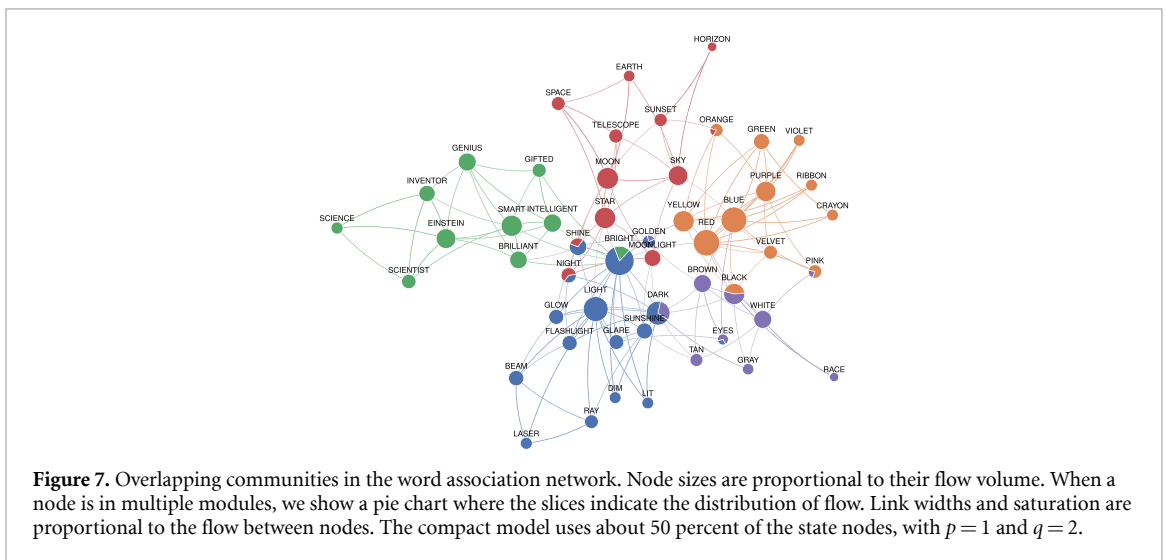
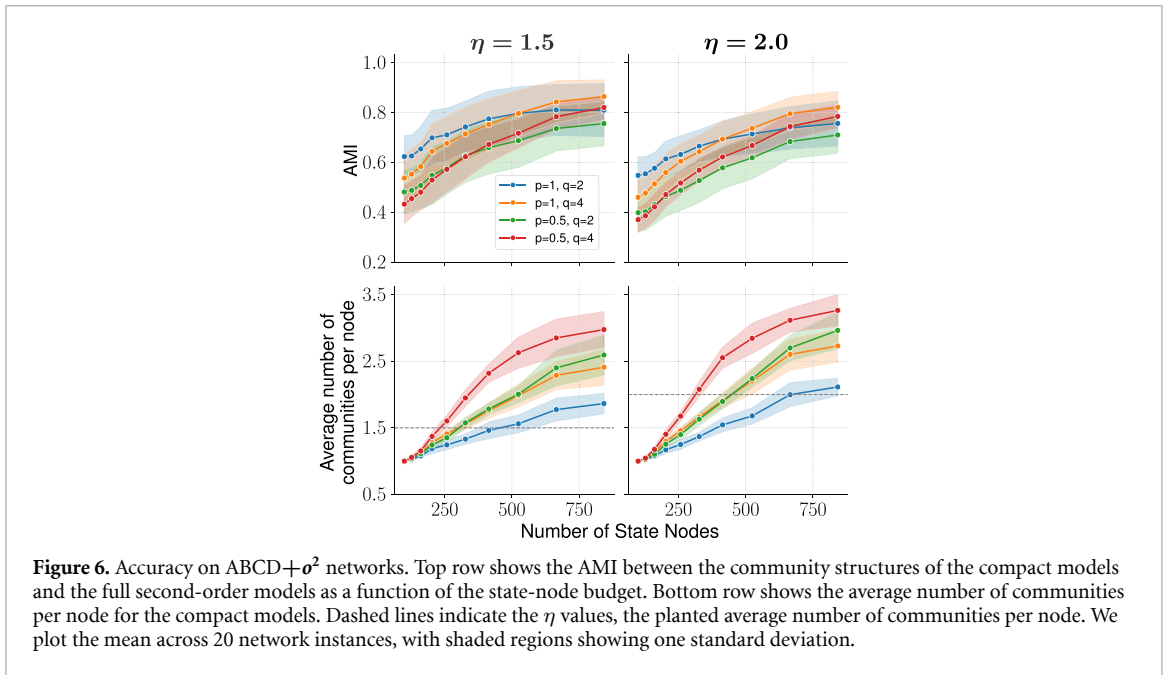
To extend our analysis on synthetic benchmark networks, we include the Artificial Benchmark for Community Detection with Outliers and Overlapping Communities (ABCD+ o^2) [26, 27]. This benchmark produces large and realistic networks characterised by heavy-tailed degree and community-size distributions, with scalability to millions of nodes. We generate ABCD+ o^2 instances with $N = 100$ nodes, $N_{\text{out}} = 1$ outlier, power-law exponent for the degree distribution $t_1 = 2$, minimum degree $d_{\text{min}} = 6$, maximum degree $d_{\text{max}} = 15$, power-law exponent for the community-size distribution $t_2 = 1$, minimum community size $c_{\text{min}} = 10$, maximum community size $c_{\text{max}} = 20$, fraction of edges in the background graph $\xi = 0.2$, latent space dimensionality $d = 2$, and correlation between node degree and number of communities $\rho = 0.0$.

We evaluate these networks as the LFR benchmarks, but instead of varying o_n and o_m , we vary the average number of communities per node, $\eta \in \{1.5, 2.0\}$. We measure both the AMI between compact and full second-order models and the average number of communities per node in the detected overlapping communities (figure 6).

As with the LFR networks, the AMI first increases quickly with the number of state nodes. Capturing full second-order models with more overlap requires more state nodes: models with low p and high q produce higher overlap and achieve lower AMI with the same state-node budget. The compact models successfully identify more overlap when planted networks contain more overlap. Models with $p = 1$ and $q = 2$, which bias walks toward neighbours of the source node, best capture the planted structures.

3.2. Real-world networks

We apply our method to real-world networks to assess whether it reveals overlapping communities that reflect explainable structure. Specifically, we analyse a word association network centred on the word *bright*, a network of teams competing in U.S. college football, and a character association network from



a fantasy novel (figure A1). We then compare our compact memory-biased model with established overlapping community detection methods and with a full second-order memory model.

3.2.1. Word association network

The word association network represents free associations between words collected by researchers at the University of South Florida [28]. We use the network centred around the word *bright*, as studied in [5], where undirected edges connect words which are associated with each other above a given frequency threshold.

We generate a compact network model using approximately 50 percent of the state nodes with $p = 1$ and $q = 2$. Infomap identifies five communities, each corresponding to a distinct concept: *intelligence* in green, *light* in blue, *appearance* in purple, *colours* in orange, and *astronomy* in red. The modules overlap at several words with multiple meanings. For instance, both *bright* and *dark* belong to the *light* module, while *bright* also appears in the *intelligence* module and *dark* in the *appearance* module. Similarly, while *moon* and *star* are found only in the *astronomy* module, *night* also appears in the *light* module (figure 7).

Table 1. Conference assignment of independent college football teams. In the first-order memoryless model, each team is forced into exactly one community, Big East, Mid-American, or its own independent community. In contrast, in the compact memory-biased model, teams split their membership across multiple conferences, with bar-length indicating flow percentages.

	Central Florida	Connecticut	Louisiana Lafayette	Louisiana Monroe	Louisiana Tech	Middle Tennessee State	Navy	Notre Dame
First-order model								
Big East	—	—	—	—	—	—	100	100
IA Independents	100	—	100	100	100	100	—	—
Mid-American	—	100	—	—	—	—	—	—
Compact model								
Atlantic Coast	10	—	—	—	—	—	16	—
Big 10	—	—	—	—	—	—	—	16
Big 12	—	—	36	—	—	—	—	14
Big East	—	12	—	—	—	—	56	45
Conference USA	—	11	9	—	—	—	19	—
IA Independents	26	12	55	65	80	79	—	—
Mid-American	54	65	—	—	—	—	—	—
Mountain West	—	—	—	—	—	—	9	8
Pac-10	—	—	—	—	—	—	—	17
Southeastern	10	—	—	35	—	21	—	—
Western Athletic	—	—	—	—	20	—	—	—

3.2.2. American college football network

The American college football network represents games played between Division IA colleges during the fall 2000 regular season. Nodes represent teams, and edges denote games played [29, 30]. The network exhibits a strong community structure: teams are grouped into conferences and play more games within than outside their conference. In 2000, teams played an average of seven intra-conference games and four inter-conference games. Inter-conference games were unevenly distributed, geographically closer teams were more likely to face each other.

The first-order solution recovers the known conferences and consolidates nearly all independent teams into a single module, except for *Navy* and *Notre Dame*, which join the *Big East* conference and *Connecticut* which joins the *Mid-American* conference (table 1). Partitioning the compact model, constructed with $p = 1$, $q = 4$, and just under 70 percent of the state nodes, reveals 12 communities. The communities mostly overlap at the independent teams *Central Florida*, *Connecticut*, *Navy*, *Notre Dame*, *Middle Tennessee State*, *Louisiana Tech*, and *Louisiana Monroe* (table 1). Additional overlaps arise when teams frequently play multiple inter-conference games against geographically nearby conferences (figure 8).

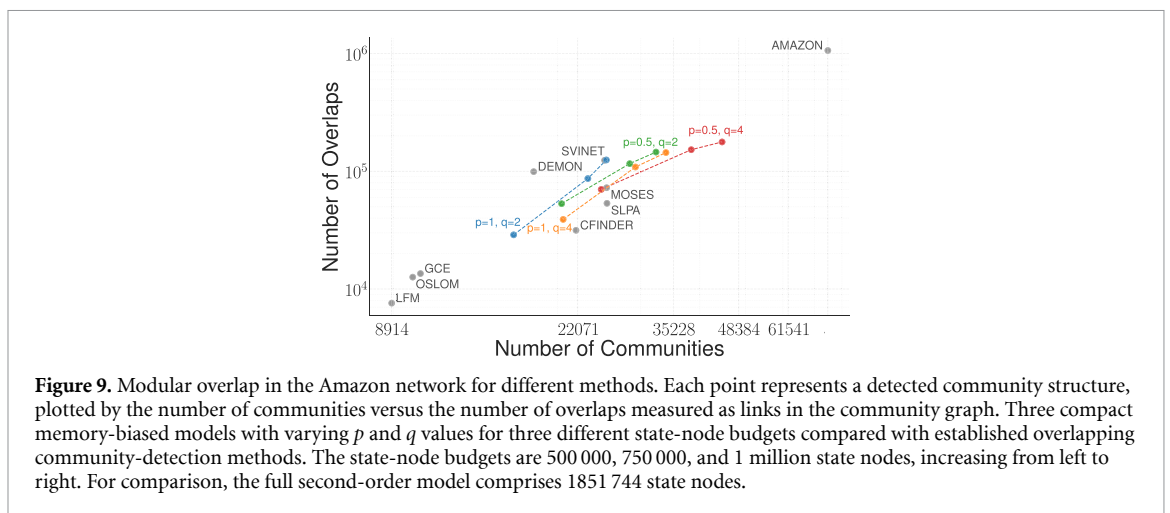
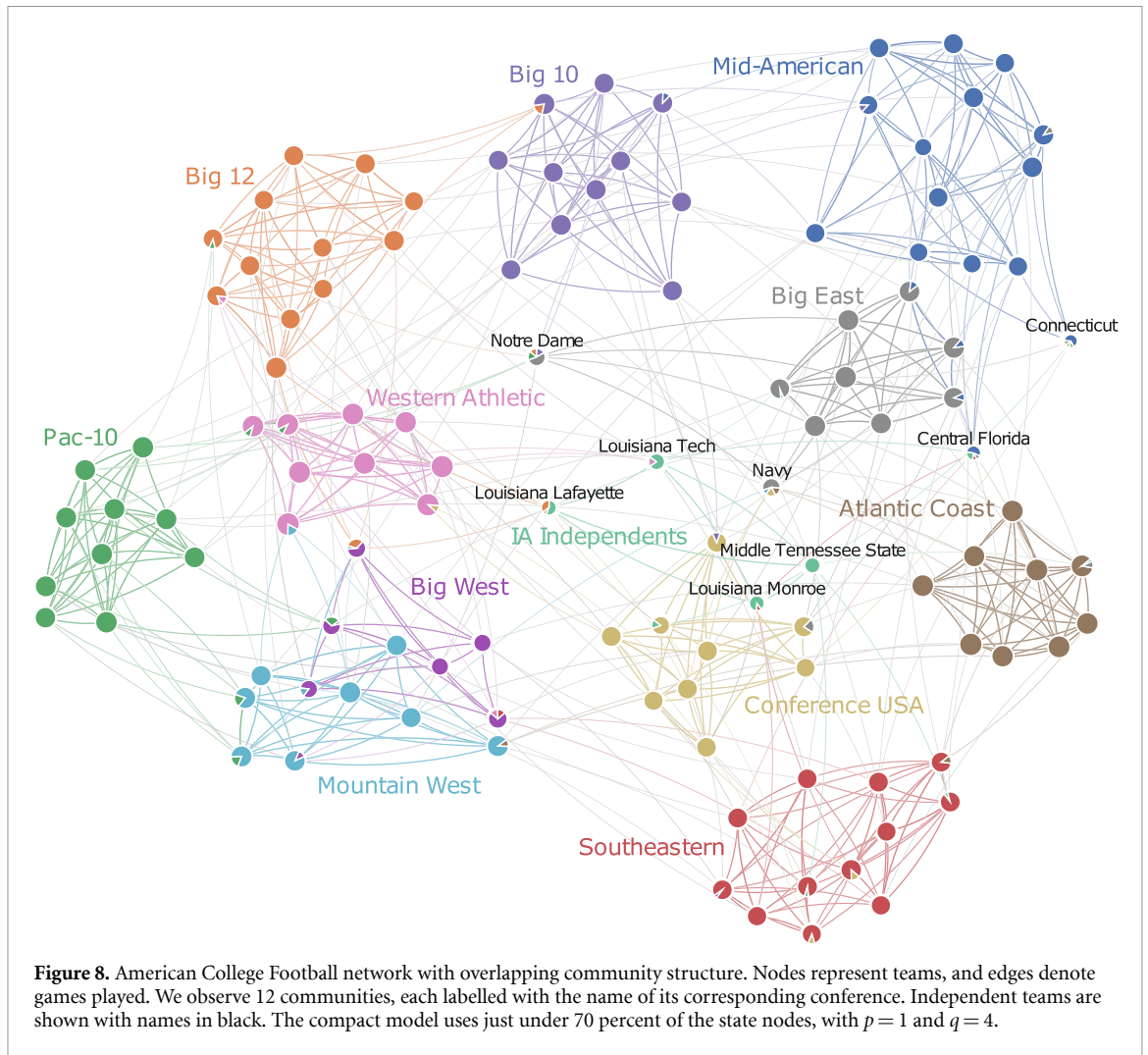
3.2.3. A Storm of Swords network

We explore overlapping story arcs in the fantasy novel *A Storm of Swords* using its character co-occurrence network [31]. The network consists of 303 nodes representing characters and 1008 edges, where weights count how often their names appear within 15 words of each other. We analyse the community structure of a compact model with 537 state nodes constructed with $p = 1$ and $q = 4$. Infomap identifies 31 modules, of which 10 include at least 10 characters and together account for 90 percent of the total flow. These modules largely correspond to story arcs in distinct locations such as *The Wall*, *King's Landing*, and *Essos* (appendix A.1).

Of the 9 central characters with more than 2 percent of the flow each, 6 appear in multiple modules, indicating that they play roles in multiple storylines (table A2). For instance, Jon Snow appears in *North of the Wall*, *The Wall*, *Jon's Watch*, and *Winterfell* modules, and Arya Stark is involved in *Arya's adventures*, *The Brotherhood without Banners*, *The Riverlands*, and *Winterfell*.

3.2.4. Amazon network

To compare our compact memory-biased model with established overlapping community detection methods, we analyse the Amazon product co-purchasing network [32]. In this network, nodes represent products and edges connect items frequently bought together. Amazon assigns each product to one or more categories, but this classification serves commercial purposes rather than capturing network structure. The categories produce far more communities and overlaps than any detection method



recovers-including LFM [33], OSLOM [6], and our approach (figure 9). Following the work of Jebabli et al [34], we construct a *community graph* where nodes represent communities and edges represent overlaps. To align with their results, we remove singleton nodes from the final community graph.

For the compact memory-biased model, the parameters control overlap in intuitive ways. Lower p values and higher q values increase the tendency to backtrack, causing random walks to revisit physical nodes assigned to multiple communities before settling. This behaviour produces more overlap. The compact models with larger state-node budgets identify more communities with greater overlap than existing methods, though still far fewer than Amazon’s category system suggests.

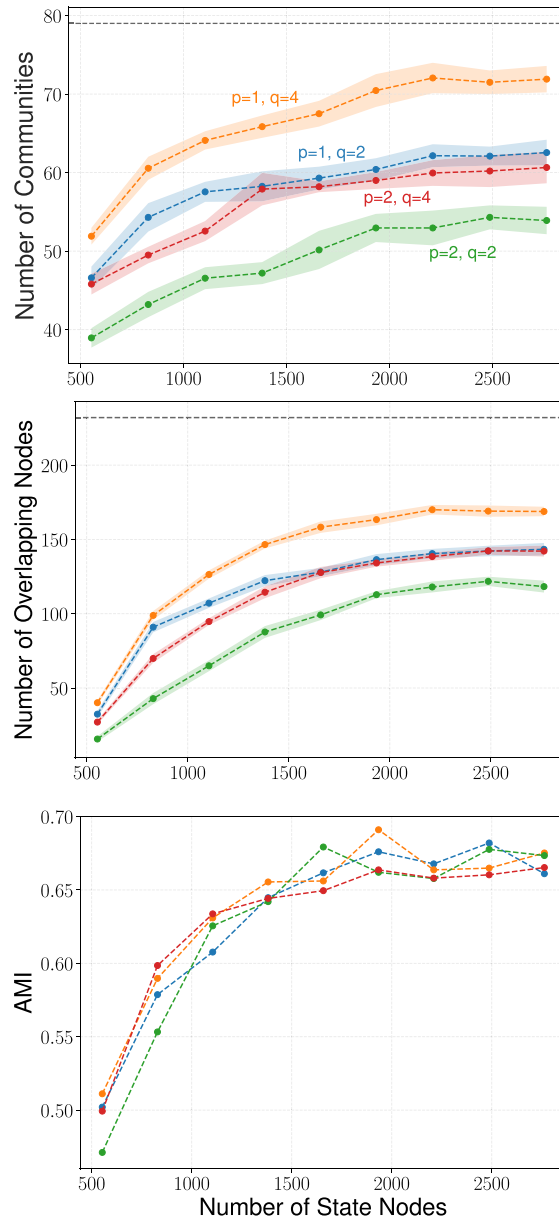


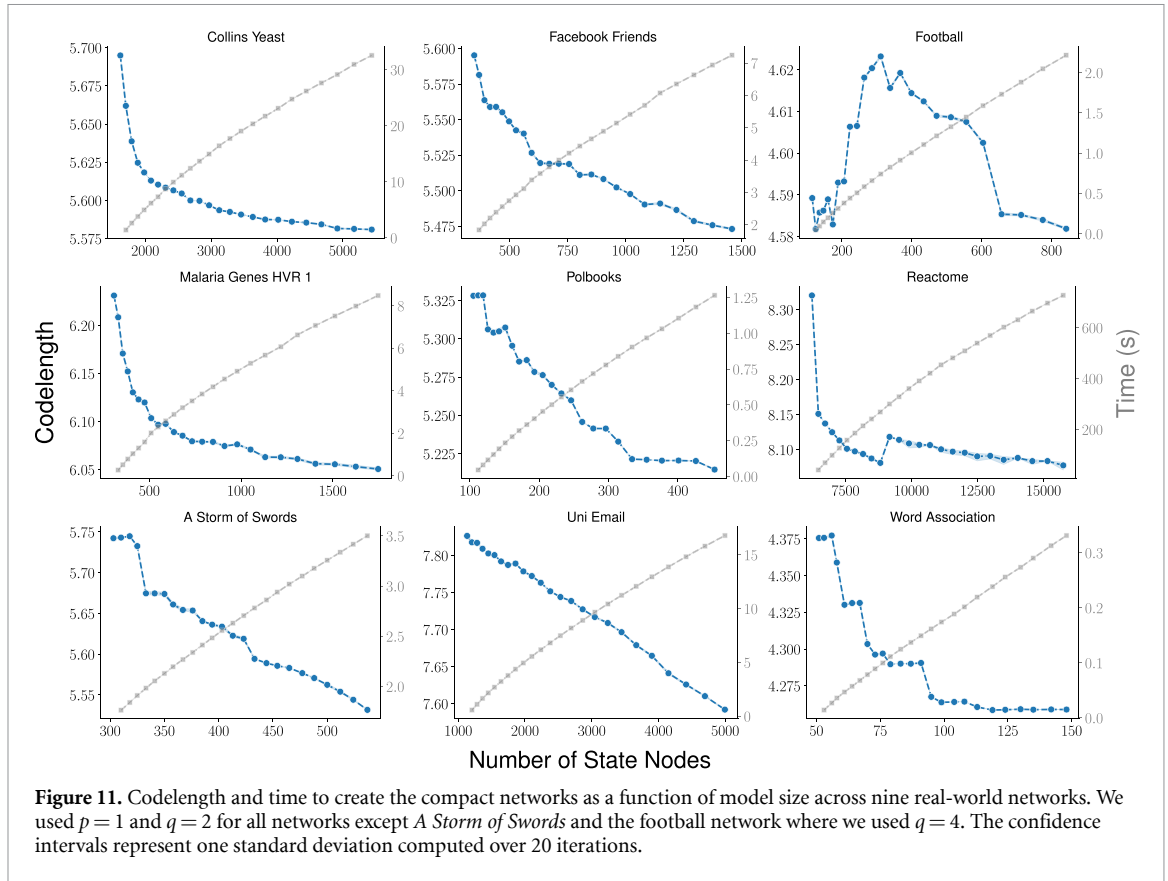
Figure 10. Compact memory-biased models compared with a model using higher-order network data for taxi drives. Four compact memory-biased models with varying p and q values for nine state-node budgets compared with a model using complete trigram data (dashed lines). The confidence intervals represent one standard deviation computed over 20 iterations.

3.2.5. Taxi drives

We compare the compact memory-biased model to a model constructed from the complete second-order representation of GPS-tracked taxi drives in San Francisco [14]. Examining the actual taxi trajectories reveals asymmetric memory: taxis return to recently visited locations less frequently after two steps but more frequently after three steps compared with memoryless expectations. This pattern suggests parameters with $p > 1$ and $q > 1$.

As we increase the state-node budget, the compact model identifies more communities that overlap in more nodes (figure 10). The number of communities and number of overlapping nodes approach those found with complete trigram data. The AMI shows a similar trend: high- q models, which best match the triangle-closing bias in the trigram data, achieve the highest AMI before levelling off.

The AMI plateaus below 1 for two reasons. First, we compare with one particular partition of the trigram data in a flat solution landscape. Second, uniform p and q parameters cannot capture the spatially varying memory in real taxi flows. While the compact model cannot fully replicate the spatial heterogeneity present in complete trigram data, it can capture dominant memory patterns and approximate the community structure.



3.3. Model size

Our aim is to recover the modular structure of the full second-order model using fewer state nodes. Adding more state nodes expands the space of possible partitions and typically yields higher compression but also longer search times. We use the codelength as a proxy for model quality to guide the selection of an appropriate model size.

To evaluate this trade-off in practice, we analyse nine different networks (figure 11) and observe a consistent pattern. In most cases, the codelength drops sharply with the initial addition of state nodes, then gradually levels off as more are included. This drop suggests that substantial improvements can be achieved by moving beyond the simplest model M_1 , but as the number of state nodes approaches that of M_2 , the additional gains become marginal. Construct time increases sub-linearly with state-node budget, consistent with the divisive clustering algorithm's $\mathcal{O}(md)$ complexity. Infomap clustering then scales linearly with the number of links in the compact network [17]. These scaling properties define the practical trade-off: larger models achieve lower codelength but require longer construction and clustering, while compact models sacrifice some compression for faster analysis.

4. Conclusion

We introduce a method to detect overlapping communities in first-order networks by modelling memory-biased random walks. Drawing on principles from *node2vec* and higher-order network models, we avoid explicit random walk simulations by expressing memory constraints with state nodes. This representation enables ergodic network flow calculations and direct use of the community-detection algorithm Infomap for assigning state nodes to modules. For a compact representation that balances sparsity and expressiveness, we lump state nodes based on an information loss criterion. Users can control this trade-off in two ways: by setting a state-node budget that limits the total number of nodes, or by specifying the acceptable level of information loss. By default, the method uses a compression threshold of 90 percent.

The bias parameters p and q in *node2vec* provide flexibility, allowing users to tailor the memory structure to the network at hand. Applied to overlapping LFR and ABCD- o^2 benchmark networks, our method is robust across a wide range of parameter values. The AMI rises sharply as we add state nodes, demonstrating that compact models with about 50 percent of the full representation capture essential

dynamics. Models successfully identify overlap when present, with the state-node requirement scaling naturally with the amount of overlap. In real-world networks with overlapping community structures, we identify coherent and interpretable overlapping communities similar to those identified with complete higher-order network data. While the uniform p and q parameters cannot capture node-specific memory patterns present in real higher-order data, our empirical results demonstrate that compact models with global parameters approximate community structures obtained from complete higher-order data.

Data availability statement

We collected the word association network from [35], the *A Storm of Swords* network from [31], the Amazon network from [32], the taxi drives from [14], and downloaded the other networks from [36]. The LFR and ABCD+ o^2 benchmark source codes are available from [37, 38].

No new data were analysed in this study.

Acknowledgments

We thank Jelena Smiljanić, Renaud Lambiotte and Tommy Löfstedt for helpful discussions and comments. M L was supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) [39], funded by the Knut and Alice Wallenberg Foundation. R S was funded by the Mathematical Institute at the University of Oxford. C B acknowledges funding from the Swiss National Science Foundation, Grant 176938, and the German Federal Ministry of Education and Research, Grant 100582863 (TissueNet). M R was supported by the Swedish Research Council under Grant 2023-03705.

Appendix

A.1. A storm of swords

We list the 10 modules in the compact Storm of Swords network with at least 10 characters and name them based on the story arc they correspond to.

1. **King's Landing (63 characters, 28.6% flow)** Addam Marbrand, Aegon Targaryen (son of Rhaegar), Aemon Targaryen (Dragonknight), Aerys II Targaryen, Alayaya, Alerie Hightower, Amory Lorch, Anders Yronwood, Baelor I Targaryen, Balon Swann, Boros Blount, Brella, Bronn, Butterbumps, Cersei Lannister, Chataya, Doran Martell, Eddard Stark, Elia Martell, Ellaria Sand, Galyeon of Cuy, Garlan Tyrell, Gregor Clegane, High Septon (Tyrions), Jaime Lannister, Joanna Lannister, Joffrey Baratheon, Kella, Kevan Lannister, Lancel Lannister, Leonette Fossoway, Lollys Stokeworth, Loras Tyrell, Lysa Arryn, Mace Tyrell, Maegor I Targaryen, Mandon Moore, Margaery Tyrell, Mathis Rowan, Meryn Trant, Mordane, Myrcella Baratheon, Oberyn Martell, Olenna Redwyne, Osmund Kettleblack, Paxter Redwyne, Podrick Payne, Pycelle, Renly Baratheon, Robert Baratheon, Sansa Stark, Shae, Stannis Baratheon, Symon Silver Tongue, Tanda Stokeworth, Tommen Baratheon, Tyrion Lannister, Tysha, Tywin Lannister, Urswyck, Varys, Viserys Targaryen, Willas Tyrell
2. **The Wall (35 characters, 13.0% flow)** Aemon Targaryen (Maester Aemon), Alliser Thorne, Bannen, Bedwyck, Blane, Bowen Marsh, Byam Flint, Chett, Clubfoot Karl, Clydas, Cotter Pyke, Craster, Denys Mallister, Dirk, Donal Noye, Dywen, Eddison Tollett, Gilly, Grenn, Hobb, Janos Slynt, Jeor Mormont, Jon Snow, Lark, Mance Rayder, Othell Yarwyck, Pypar, Qhorin Halfhand, Ragwyle, Samwell Tarly, Small Paul, Softfoot, Stannis Baratheon, Watt, Wynton Stout
3. **The Riverlands (42 characters, 12.3% flow)** Aegon Frey (son of Stevron), Arya Stark, Balon Greyjoy, Bennett, Brynden Tully, Catelyn Stark, Dacey Mormont, Desmond Grell, Edmure Tully, Elmar Frey, Gawen Westerling, Hoster Tully, Jaime Lannister, Jeyne Westerling, Jon Umber (Greatjon), Jon Umber (Smalljon), Lothar Frey, Maege Mormont, Marq Piper, Olyvar Frey, Perwyn Frey, Petyr Frey, Ramsay Snow, Raynald Westerling, Rickard Karstark, Robb Stark, Robin Ryger, Rodrik Cassel, Rollam Westerling, Rolph Spicer, Roose Bolton, Roslin Frey, Ryman Frey, Stevron Frey, Sybell Spicer, Tansy, Tristifer IV Mudd, Tywin Lannister, Utherydes Wayn, Vyman, Walder Frey, Walder Rivers
4. **Arya's adventures (12 characters, 7.4% flow)** Arya Stark, Cersei Lannister, Dunsen, Gregor Clegane, Ilyn Payne, Joffrey Baratheon, Meryn Trant, Mycah, Polliver, Rafford, Sandor Clegane, Tickler
5. **North of the Wall (23 characters, 6.2% flow)** Arson, Big Boil, Dalla, Errok, Gendel, Gorne, Grigg, Harma, Jarl, Jon Snow, Joramun, Mance Rayder, Munda, Orell, Rattleshirt, Ryk, Styr, Tormund, Val, Varamyr, Weeper, Ygritte, Zei

Table A1. Summary of the real-world networks used in the analysis.







Networks	Nodes	Edges	State nodes (M_2)	State nodes (M_c)
collins_yeast [40]	1622	9070	18 140	5436
facebook_friends [41]	362	1988	3976	1457
football_tsevens [29, 30]	115	613	1226	842
malaria_genes/HVR_1 [42]	307	2812	5624	1748
polbooks [43]	105	441	882	453
reactome [44]	6327	147 547	292 320	15 745
storm_of_swords [31]	303	1008	2016	537
uni_email [45]	1133	10 902	10 902	4995
word_association [5]	51	149	298	148

Table A2. Central characters in A Storm of Swords. The characters with more than 2 percent flow each and the modules they belong to.

Character	Total flow	Module (fraction of flow)
Tyrion Lannister	0.035	King's Landing (1.0)
Joffrey Baratheon	0.030	King's Landing (0.82) Arya's adventures (0.18)
Jon Snow	0.030	North of the Wall (0.42) The Wall (0.26) Jon's Watch (0.19) Winterfell (0.13)
Sansa Stark	0.027	King's Landing (1.0)
Robb Stark	0.027	The Riverlands (0.90) Winterfell (0.10)
Jamie Lannister	0.025	King's Landing (0.74) The Riverlands (0.15) Module # 11 (0.11)
Cersei Lannister	0.023	King's Landing (0.73) Arya's adventures (0.27)
Arya Stark	0.022	Arya's adventure (0.42) The Brotherhood without Banners (0.31) The Riverlands (0.13) Winterfell (0.13)
Catelyn Stark	0.020	The Riverlands (1.0)

6. **Essos (25 characters, 5.9% flow)** Arstan, Barristan Selmy, Belwas, Ben Plumm, Cleon, Daario Naharis, Daenerys Targaryen, Drogo, Ghael, Grazdan mo Eraz, Grey Worm, Groleo, Illyrio Mopatis, Irri, Jhiqui, Jorah Mormont, Kraznys mo Nakloz, Mero, Missandei, Oznak zo Pahl, Rhaegar Targaryen, Rhaegel Targaryen, Rhaella Targaryen, Robert Baratheon, Viserys Targaryen
7. **The Brotherhood without Banners (20 characters, 5.7% flow)** Anguy, Arya Stark, Beric Dondarrion, Gendry, Greenbeard, Harwin, Hot Pie, Husband, Jack Be Lucky, Jack Bulwer, Lem, Mad Huntsman, Merrit, Pate (Old), Ravella Swann, Sandor Clegane, Sharna, Tansy, Thoros of Myr, Tom of Sevenstreams
8. **Dragonstone (19 characters, 5.0% flow)** Alester Florent, Andrew Estermont, Axell Florent, Azor Ahai, Cressen, Davos Seaworth, Devan Seaworth, Edric Storm, Gerald Gower, Guncer Sunglass, Khorane Sathmantes, Lamprey, Melisandre, Porridge, Pylos, Salladhor Saan, Selyse Florent, Shireen Baratheon, Stannis Baratheon
9. **Jon's Watch (10 characters, 3.3% flow)** Dick Follard, Grenn, Hareth (Moles Town), Jon Snow, Kegs, Mully, Owen, Pypar, Satin, Spare Boot
10. **Winterfell (11 characters, 2.7% flow)** Arya Stark, Benjen Stark, Bran Stark, Brandon Stark, Eddard Stark, Jon Snow, Luwin, Nan, Rickon Stark, Robb Stark, Theon Greyjoy

ORCID iDs

Maja Lindström  0009-0009-9224-4646
 Rohit Sahasrabudde  0000-0002-2779-8310
 Anton Holmgren  0000-0001-5859-4073
 Christopher Blöcker  0000-0001-7881-2496
 Daniel Edler  0000-0001-5420-0591
 Martin Rosvall  0000-0002-7181-9940

References

- [1] Pósfai M and Barabási A-L'o 2016 *Network Science* vol 3 (Cambridge University Press)
- [2] Newman M 2018 *Networks* (Oxford University Press)
- [3] Fortunato S and Hric D 2016 Community detection in networks: a user guide *Phys. Rep.* **659** 1–44
- [4] Xie J, Kelley S and Szymanski B K 2013 Overlapping community detection in networks: the state-of-the-art and comparative study *ACM Comput. Surv.* **45** 1–35
- [5] Palla G, Derényi I, Farkas I and Vicsek T 2005 Uncovering the overlapping community structure of complex networks in nature and society *Nature* **435** 814–8
- [6] Lancichinetti A, Radicchi F, Ramasco J J, Fortunato S and Ben-Jacob E 2011 Finding statistically significant communities in networks *PLoS One* **6** e18961
- [7] Ahn Y-Y, Bagrow J P and Lehmann S 2010 Link communities reveal multiscale complexity in networks *Nature* **466** 761–4
- [8] Lotito Q F, Musciotto F, Montresor A and Battiston F 2024 Hyperlink communities in higher-order networks *J. Comp. Netw.* **12** cnae013
- [9] Rosvall M and Bergstrom C T 2008 Maps of random walks on complex networks reveal community structure *Proc. Natl Acad. Sci.* **105** 1118–23
- [10] Scholtes I, Wider N, Pfitzner R, Garas A, Tessone C J and Schweitzer F 2014 Causality-driven slow-down and speed-up of diffusion in non-Markovian temporal networks *Nat. Commun.* **5** 5024
- [11] Masuda N, Porter M A and Lambiotte R 2017 Random walks and diffusion on networks *Phys. Rep.* **716** 1–58
- [12] Ballal A, Kion-Crosby W B and Morozov A V 2022 Network community detection and clustering with random walks *Phys. Rev. Res.* **4** 043117
- [13] Bovet A, Delvenne J-C and Lambiotte R 2022 Flow stability for dynamic community detection *Sci. Adv.* **8** eabj3063
- [14] Rosvall M, Esquivel A V, Lancichinetti A, West J D and Lambiotte R 2014 Memory in network flows and its effects on spreading dynamics and community detection *Nat. Commun.* **5** 1–13
- [15] Jian X, Wickramaratne T L and Chawla N V 2016 Representing higher-order dependencies in networks *Sci. Adv.* **2** e1600028
- [16] Sahasrabudde R, Lambiotte R and Rosvall M 2025 Concise network models of memory dynamics reveal explainable patterns in path data *Sci. Adv.* **11** eadw4544
- [17] Edler D, Bohlin L and Rosvall M 2017 Mapping higher-order network flows in memory and multilayer networks with infomap *Algorithms* **10** 112
- [18] Lambiotte R, Rosvall M and Scholtes I 2019 From networks to optimal higher-order models of complex systems *Nat. Phys.* **15** 313–20
- [19] Esquivel A V and Rosvall M 2011 Compression of flow can reveal overlapping-module organization in networks *Phys. Rev. X* **1** 021025
- [20] Persson C, Bohlin L, Edler D and Rosvall M 2016 Maps of sparse Markov chains efficiently reveal community structure in network flows with memory (arXiv:1606.08328)
- [21] Grover A and Leskovec J 2016 node2vec: Scalable feature learning for networks *Proc. 22nd ACM SIGKDD* pp 855–64
- [22] Rosvall M, Axelsson D and Bergstrom C T 2009 The map equation *Eur. Phys. J. Spec. Top.* **178** 13–23
- [23] Edler D, Holmgren A and Rosvall M 2023 The MapEquation software package (available at: <https://mapequation.org>)
- [24] Lancichinetti A and Fortunato S 2009 Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities *Phys. Rev. E* **80** 016118
- [25] Calatayud J, Bernardo-Madrid R, Neuman M, Rojas A and Rosvall M 2019 Exploring the solution landscape enables more reliable network community detection *Phys. Rev. E* **100** 052308
- [26] Kamiński B, Prałat P and Théberge F 2021 Artificial benchmark for community detection (abcd)-fast random graph model with community structure *Netw. Sci.* **9** 153–78
- [27] Barrett J, DeWolfe R, Kamiński B, Prałat P, Smith A and Théberge F 2025 The artificial benchmark for community detection with outliers and overlapping communities (abcd+ o2) *Int. Workshop on Modelling and Mining Networks* (Springer) pp 125–40
- [28] Nelson D L, McEvoy C L and Schreiber T A 2004 The University of South Florida word association, rhyme, and word fragment norms *Behavior Research Methods, Instruments, & Computers* vol 36 pp 402–7 (available at: <https://link.springer.com/article/10.3758/BF03195588>)
- [29] Girvan M and Newman M E J 2002 Community structure in social and biological networks *Proc. Natl Acad. Sci.* **99** 7821–6
- [30] Evans T S 2010 Clique graphs and overlapping communities *J. Stat. Mech.* **P12037**
- [31] Beveridge A A song of ice and fire character interaction networks (available at: <https://github.com/mathbeveridge/asoiaf/tree/master>) (Accessed 16 April 2025)
- [32] Yang J and Leskovec J 2012 Defining and evaluating network communities based on ground-truth *Proc. of the ACM SIGKDD Workshop on Mining Data Semantics* pp 1–8
- [33] Lancichinetti A, Fortunato S and Kertész J 2009 Detecting the overlapping and hierarchical community structure in complex networks *New J. Phys.* **11** 033015
- [34] Jebabli M, Cherifi H, Cherifi C and Hamouda A 2018 Community detection algorithm evaluation with ground-truth data *Physica A* **492** 651–706
- [35] Palla G, Derényi I, Farkas I and Vicsek T 2005 CFinder is free software for finding and visualizing overlapping dense groups of nodes in networks, based on the clique percolation method (CPM) *Nature* **435** 814–8 (available at: www.cfinder.org)
- [36] Peixoto T P 2020 *The Netzschleuder network catalogue and repository* (available at: <https://networks.skewed.de/>)

- [37] Lancichinetti A, Fortunato S and Kertész J 2009 Detecting the overlapping and hierarchical community structure of complex networks *New J. Phys.* **11** 033015 (available at: <https://sites.google.com/site/andrealancichinetti/benchmarks>)
- [38] Kamiński B, Prałat P and Théberge F 2019 ABCDGraphGenerator.jl (MIT License) (available at: <https://github.com/bkamins/ABCDGraphGenerator.jl>)
- [39] Wallenberg AI, Autonomous Systems and Software Program (WASP) (available at: <http://wasp-sweden.org>)
- [40] Collins S R, Kemmeren P, Zhao X-C, Greenblatt J F, Spencer F, Holstege F C P, Weissman J S and Krogan N J 2007 Toward a comprehensive atlas of the physical interactome of *saccharomyces cerevisiae* *Mol. Cell. Proteom.* **6** 439–50
- [41] Maier B F and Brockmann D 2017 Cover time for random walks on arbitrary complex networks *Phys. Rev. E* **96** 042307
- [42] Larremore D B, Clauset A, Buckee C O and Antia R 2013 A network approach to analyzing highly recombinant malaria parasite genes *PLoS Comput. Biol.* **9** e1003268
- [43] Pasternak B and Ivask I 1970 Four unpublished letters *Books Abroad* **44** 196–200
- [44] Joshi-Tope G *et al* 2005 Reactome: a knowledgebase of biological pathways *Nucl. Acids Res.* **33** D428–32
- [45] Guimera R, Danon L, Diaz-Guilera A, Giralt F and Arenas A 2003 Self-similar community structure in a network of human interactions *Phys. Rev. E* **68** 065103